Наредбе за организацију циклуса

Када у алгоритму за решавање неког проблема имамо потребу да се нека наредба изврши више пута са променом почетних вредности или са истим вредностима онда имамо потребу за наредбама циклуса (вишеструког извршавања). У зависности од тога да ли унапред знамо број извршења циклуса или тај број зависи од неког услова разликујемо:

- бројачке циклусе и
- циклусе са условом

Бројачки циклус је ***For*** циклус. Код њега се тачно зна колико пута ће се извршити нека наредба или група наредби. Циклус извршења наредби се не може прекинути пре извршења унапред задатог броја понављања.

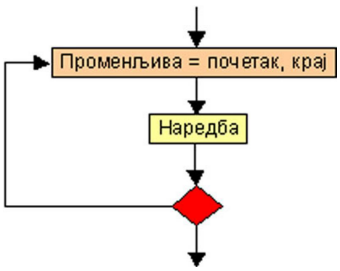
Циклуси са условом се извршавају произвољан број пута, а извршење циклуса наредби се прекида када се постигне неки услов. Циклуси са условом могу бити:

- циклуси са условом на крају (***Repeat ... until <uslov>***) и
- циклуси са условом на почетку (***While <uslov> do ...***)

Синтакса наредби за организацију циклуса.

- ***Циклус For ... do ...***

Циклус ***For ... do ...*** је најједноставнија наредба циклуса. Користи се када је број понављања извршења наредбе или групе наредби унапред познат, односно, тачно одређен.



Ова наредба има два општа облика:

```
FOR <promenljiva> := <n1> TO <n2> DO <naredba>;
```

и

```
FOR <promenljiva> := <n1> DOWNTO <n2> DO <naredba>;
```

<*promenljiva*> је бројач, односно, бројачка променљива циклуса, а број извршења циклуса је разлика између бројева <*n1*> и <*n2*>, тачније, број извршења циклуса добија се по формули *|n1-n2|+1*. Иза службене речи ***do*** никада не стоји тачка-зарез јер ту није крај наредбе циклуса. Ако се, ипак, стави онда се подразумева да је циклус празан, односно, у циклуси се неће извршити ни једна наредба. Бројачка променљива циклуса мора бити простог типа, односно, типа *boolean, integer* или *char* (или неког од њих изведеног), а <*n1*> и <*n2*> морају се по типу слагати са бројачком променљивом циклуса. Ако је вредност променљиве <*n1*> мања употребљава се службена реч ***to***, а ако је вредност променљиве <*n2*> мања службена реч ***downto***.

Ако иза службене речи ***do*** треба да стоји више од једне наредбе оне се повезују наредбом састављања, односно:

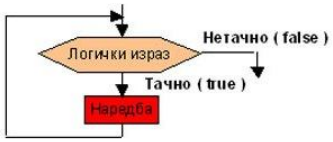
```
FOR <promenljiva> := <n1> TO <n2> DO
    Begin <naredbe>;
    End;
FOR <promenljiva> := <n1> DOWNTO <n2> DO
    Begin <naredbe>;
    End;
```

У случају када је иза ***do*** само једна наредба, наредба састављања није обавезна, али је дозвољена, односно, може се употребити ако то желимо, али је синтаксно непотребна.

- ***REPEAT - циклус са условом на крају***

Када неки посао треба обавити више пута, али се не зна тачан број понављања већ тај број зависи од неког услова може се употребити ***For*** наредба. Међутим, њена употреба би захтевала и употребу комбинације наредби условног и безусловног преласка, чиме би структура програма била јако оштећена. Да би се и овакви задаци могли једноставније решити

уведена је наредба ***Repeat ... until ...*** која се извршава све док се услов који одређује број понављања не испуни.



Општи облик ове наредбе је:

```
REPEAT <naredba1>;
       <naredba2>;
        ...
       <naredbaN>;
UNTIL  <uslov>;
```

Број наредби унутар циклуса није ограничен, тј. у случају да у циклусу треба да се изврши више од једне наредбе нема потребе за писањем наредбе састављања, али је она дозвољена (уколико неко баш воли да је пише може је додати и то ће бити без последица и без нарушавања структуре програма само ће се програм, непотребно, компликовати).

Сваки задатак решен наредбом ***For*** може се решити исто тако једноставно наредбом ***Repeat***. Обрнуто не важи. Тачније, сваки задатак решен ***Repeat*** наредбом може се решити и ***For*** наредбом, али то није увек и једноставно.

Услов који одређује број извршавања циклуса може бити прост или сложен, односно, састављен од неколико простих услова повезаних логичким операцијама. У случају сложеног услова важе сва правила логичког закључивања, а сваки од простих услова, из којих се он састоји, пише се између посебних заграда:

```
REPEAT <naredbe>;
UNTIL (uslov1)or(uslov2)and(uslov3);
```

Да би се изашло из циклуса мора бити испуњен први услов или истовремено друга два услова (могу бити испуњена и сва три услова).

```
REPEAT <naredbe>;
UNTIL ((uslov1)or(uslov2))and(uslov3);
```

Да би се изашло из циклуса мора бити испуњен бар један од прва два услова и трећи услов (могу бити испуњена и сва три услова).

У оквиру једне ***repeat*** наредбе може бити других ***repeat*** наредби или других наредби циклуса. У таквим случајевима мора се водити рачуна да не дође до нарушавања структуре програма. Наиме, није дозвољено да се неки циклус делимично налази унутар другог циклуса, а делимично изван њега. Циклус може бити у другом циклусу с тим да му и почетак и крај буду унутар тог другог циклуса или су циклуси један за другим, тј. почетак другог је иза краја првог циклуса.

У циклусу може бити гранања, при чему не сме доћи до нарушавања структуре, тј. не може се десити да једна грана ***if*** наредбе (на пример ***then*** грана) буде у циклусу, а друга изван циклуса, нити неки случајеви ***case*** наредбе могу бити у циклусу, а други изван њега.

- ***WHILE - циклус са условом на почетку***

Ако број извршавања циклуса одређује услов који треба да се испуни користи се наредба ***Repeat***. Међутим, врло често број извршавања наредби циклуса зависи од услова који је већ испуњен. У том случају циклус се извршава све док је тај услов испуњен. Тада је природније употребити ***While ... do ...*** наредбу.

Општи облик ове наредбе је:

```
WHILE <uslov> DO <naredba>;
```

односно:

```
WHILE <uslov> DO
      Begin <naredbe>;
       End;
```

Ако иза службене речи ***do*** треба да стоји више од једне наредбе, оне се обавезно везују наредбом састављања, јер ако се то не уради у циклусу ће се извршавати само прва у том низу наредби што може довести до формирања бесконачног циклуса (ако се у тој првој наредби вредност услова не мења) или до проласка кроз циклус потребан број пута, али без резултата, ако се услов мења у првој наредби, а остале радње извршавају у наредним. И у једном и у другом случају циклус није оправдао своје постојање, па зато треба водити рачуна о овоме. У случају када иза до треба да стоји само једна наредба, наредба састављања није обавезна, али је дозвољена, односно, може се употребити, али за њом нема потребе.

Сваки задатак који подразумева циклус може се једноставно решити овом наредбом, тј. ***While*** наредба може да замени и ***For*** и ***Repeat*** наредбу. Обрнуто не важи. Тачније, не може се сваки задатак са циклусом једноставно решити без коришћења наредбе ***While***.

Услов који одређује број извршавања циклуса може бити прост или сложен, тј. састављен од неколико простих услова. У случају сложеног услова важе правила логичког закључивања, а сваки од простих услова пише се између посебних заграда:

```
WHILE (uslov1)or(uslov2)and(uslov3) DO <naredba>;
```

да би се изашло из циклуса не сме бити испуњен први услов и бар један од друга два услова.

```
WHILE ((uslov1)or(uslov2))and(uslov3) DO <naredba>;
```

да би се изашло из циклуса не сме бити испуњен ниједан од прва два услова или трећи услов.

Примена наредби break и continue у циклусима

Наредба ***break*** користи се за превремени завршетак циклуса ***for, while*** и ***repeat***. Ова наредба се мора налазити у границама тела циклуса. У случају њеног извршења, прекида се извршавање циклуса који је непосредно обухвата и предаје контрола наредби која следи непосредно иза циклуса.

Наредба ***break*** омогућава излазак само из текућег циклуса, а не и из свих циклуса унутар којих се налази, ако је текући угњежден у неки други циклус.

Наредба ***continue*** прекида извршавање текуће итерације (прескакањем преосталих наредби тела циклуса) оног циклуса који је непосредно обухвата и обавља прелазак на следећу итерацију, ако су задовољени услови за њено извршавање (провером вредности излазних критеријума наредбе циклуса); у противном, циклус се прекида. ***Continue*** наредба присиљава прескакање преосталих наредби унутар циклуса и прелазак на почетак циклуса, на следећу итерацију циклуса.

Continue наредба скаче само на почетак текућег циклуса, а не и на почетак свих циклуса унутар којих се налази, ако је текући угњежен у неки други циклус.

- **Важно**:
  Наредбе ***break*** и ***continue*** нису у духу структурног програмирања јер ломе структуру програма и као такве их треба избегавати. Добар програмер увек може организовати циклус тако да коришћење ових наредби није потребно. Зато, у задацима које будемо решавали нећемо користити ове наредбе.

Неки алгоритми са циклусима

Следи неколико примера урађених уз коришћење циклуса ***while***, а затим и ***repeat*** циклуса. Обратите пажњу код преласка са решења са циклусом ***repeat*** на решење са циклусом ***while*** услов у задатку мора бити супротан, тј. < прелази у ≥, > прелази у ≤, ≤ прелази у >, ≥ прелази у <, ***or*** прелази у ***and***, а ***and*** прелази у ***or***.

- ***табелирање вредности функција***

Ако је потребно израчунати вредности неких функција за серију почетних вредности аргумената онда се за то могу користити циклуси. Који циклус ће се користити зависи од односа између вредности аргумената. Ако постоји неко правило на основу којег се аргументи мењају и то тако да се унапред зна број вредности аргумената можемо користити било коју врсту циклуса. Ако се не може унапред знати колико пута ће се рачунати вредност функције онда ћемо користити неки од циклуса са условом.

Нека треба исписати вредности тригонометријских функција *sin* и *cos* на интервалу од a^o до b^o степени са кораком од c^o. Задатак ћемо решити применом циклуса ***while***. Као почетну вредност аргумента функције узећемо број *a*, одредити вредност функције и исписати је, а затим повећавати аргумент док не стигнемо до броја *b*. Потребно је водити рачуна о томе да тригонометријске функције раде са угловима у радијанима.

```
ug:=a/180*pi;          // претварамо степене у радијане
While ug<=b/180*pi do   // проверавамо да ли је угао у интервалу који приказујемо
begin vred:=sin(ug);    // одређујемо вредност функције синус
      vred:=cos(ug);    // одређујемо вредност функције косинус
      ug:=ug+c/180*pi;  // одређујемо нову вредност аргумента
end;
```

Задатак је једноставно могуће решити и коришћењем циклуса ***repeat*** уз услов да корисник упише да је $a<b$, а ако није, пре примене решења, треба заменити вредности промењљивих. Решење помоћу циклуса ***for*** није тако једноставно.

```
ug:=a/180*pi;          // претварамо степене у радијане
Repeat vred:=sin(ug);  // одређујемо вредност функције синус
       vred:=cos(ug);  // одређујемо вредност функције косинус
       ug:=ug+c/180*pi; // одређујемо нову вредност аргумента
until ug>b/180*pi;            // проверавамо да ли је угао у интервалу који приказујемо
```

- ***израчунавање сума и производа***

Ако је потребно рачунати суме или производе неке групе бројева који имају неку функционалну зависност, углавном могу се користити све врсте циклуса, али се најчешће користи циклус ***for*** јер је такво решење најједноставније (осим у неким специфичним случајевима).

Нека треба сабрати све парне бројеве од a до b. Најједноставније решење је:

```
      suma:=0;             // почетна вредност за збир је 0 јер не утиче на коначан резултат
      For i:=(a+1)div 2 to b div 2 do
{ треба сабирати само парне бројеве ра је почетна вредност бројае половина броја a, користимо
операцију div јер се ради о целим бројевима, а (a+1) због тога што a може бити непаран број,
па је у том случају 2*(a div 2)<a, тј. не сабира се број a, крај сабирања је када број буде
половина броја b јер је 2*(b div 2)<=b}
       suma:=suma+2*i;// сабирамо све парне бројеве зато і множимо са 2
```

Овакав задатак је лакше решити помоћу циклуса са условом:

```
suma:=0;                // почетна вредност је иста
a:=(a+1)div 2*2;        // намештамо да а буде паран број
While a<=b do           // крај сабирања је када број а буде једнак броју b
begin suma:=suma+a;     // сабирамо паран број а са претходнима
      a:=a+2;           // узимамо следећи паран број
end;
```

Наравно, постављање почетног броја на паран број већи или једнак броју a могло се извести и помоћу једне овакве *If* наредбе:

```
                    If Odd(a) then a:=a+1;
```

али то је већ ствар *укуса* програмера:

```
suma:=0;                // почетна вредност је иста
a:=(a+1)div 2*2;        // намештамо да а буде паран број
If a<=b then            // проверавамо да ли је а мање или једнако b
  Repeat suma:=suma+a;  // сабирамо паран број а са претходнима
         a:=a+2;        // узимамо следећи паран број
      until a>b;        // крај сабирања је када број а буде једнак броју b
end;
```

На исти начин би се решавали и задаци са производом бројева. Ако би требало сабирати или множити бројеве који нису цели, онда би требало користити циклусе са условом јер је решење једноставније него са бројачким циклусима (а понекад и једино могуће).

- ***испитивање својстава целих бројева***

Проблеми типа: одредити да ли је број прост, исписати све просте делиоце броја, исписати савршене бројеве до n, исписати Армстронгове бројеве до n и слични, могу се решавати било којом врстом циклуса, а често се користи и комбиновање различитих врста, да би се решење поједноставило.

Приказаћемо први алгоритам. Број је прост ако осим 1 и самог себе нема других делиоца, зато ћемо избројати такве делиоце:

```
bd:=0;                          // број делиоца различитих од 1 и самог броја је на почетку 0
For del:=2 to broj-1 do         // почињемо тестирање од 2 до првог мањег броја
 If broj mod del=0 then bd:=bd+1;   // ако је број дељив увећавамо број делилаца за 1
If bd>0 then                        // исписати број није прост
        else                        // исписати број је прост
```

У алгоритму има пуно непотребних тестирања, па је неефикасан и не треба га користити. Било који број, осим самог себе, нема делиоца који су већи од његове половине. Број није прост ако има бар једног делиоца различитог од 1 и самог себе, па нема потребе тражити све такве делиоце. Бољи алгоритам се прекида чим се пронађе први такав делилац:

```
del:=2;                       // за првог делиоца узимамо 2
While (broj mod del<>0)       // проверавамо да ли је број дељив њиме
      and(del<=Sqrt(broj)) do // проверавамо да ли је делилац већи од корена броја
      del:=del+1;             // увећавамо делиоца за 1
If del>Sqrt(broj) then        // исписати број није прост
                  else        // исписати број је прост
del:=1;                       // за првог делиоца узимамо 2
```

```
Repeat del:=del+1;              // увећавамо делиоца за 1
until (broj mod del=0)          // проверавамо да ли је број дељив њиме
      or(del>Sqrt(broj));       // проверавамо да ли је делилац већи од корена броја
If del>Sqrt(broj) then          // исписати број није прост
                  else          // исписати број је прост
```

Први услов у циклусу је јасан. Објаснићемо други. Алгоритам је замишљен да се циклус прекида чим се наиђе на првог делиоца који је већи од 1. То значи, ако је број дељив са 2 нема потребе проверавати да ли је дељив са својом половином; ако је број дељив са 3 нема потребе проверавати да ли је дељив својом трећином и тако даље до неког броја *n*. Да бисмо пронашли који је то број формирамо низ парова делилаца који као производ дају тај број, односно: *2 * broj/2, 3 * broj/3, 4 * broj/4, ...* Први делилац стално расте, други се стално смањује и први је увек мањи од другог. Јасно је да први делилац не може бити већи од корена тог броја (јер ако је већи мора се помножити са бројем који је мањи од њега да би се добио *broj*). Значи, последњи делилац у овом низу је корен почетног броја, зато је то последњи број за који проверавамо да ли може бити делилац почетног и ако је *del* већи од корена броја значи да нисмо нашли ниједног делиоца, па је почетни број прост.

Питања на која треба обратити пажњу

- Шта је циклус
- Врсте циклуса
- Синтакса бројачког циклуса
- Број извршења бројачког циклуса
- Тип бројача циклуса
- Циклуси са условом
- Синтакса циклуса са условом
- Примена циклуса
- Сложени услов у циклусима
- Наредба састављања у циклусима

Задаци са циклусима

Пошто смо до сада већ научили да дизајнирамо лепе форме у примерима који следе нећемо описивати креирање форме, а нећемо ни писати процедуре за тастере *Крај рада* и *Обриши*, већ ћемо сву пажњу посветити алгоритамском решењу проблема и приказивати само процедуре које описују алгоритам.

- ***Саставити програм који одређује збир првих n природних бројева.***

Код задатака са циклусима важно је схватити механизам, односно, начин извршавања. На овом примеру ћемо покушати да *уђемо у штос*. Од нас се тражи да саберемо неколико узастопних бројева. Природно је овакво размишљање: сума = $a_1 + a_2 + a_3 + ... + a_n$, односно, напишемо све сабирке и затим два по два сабирамо до коначног резултата. На рачунару се тако не може. Наиме, чак и када бисмо унапред знали колико је бројева потребно сабрати не можемо овај природни запис превести у програм који ће ефикасно радити. Овако ћемо размишљати: сума = $(((...(a_1) + a_2) + a_3) + ... + a_n)$, а затим ћемо се ослобађати заграда и сваки пут ћемо вредност заграде означавати неким симболом:

$s_1:=a_1$
$s_2:=a_1+a_2:=s_1+a_2$
$s_3:=a_1+a_2+a_3:=s_2+a_3$
...
$s_n:=a_1+a_2+a_3+...a_n:=s_{n-1}+a_3$

овај низ једнакости можемо поједностављено посматрати (приказати) и овако:

$s_1:=a_1$
$s_2:=s_1+a_2$
$s_3:=s_2+a_3$
...
$s_n:=s_{n-1}+a_n$

сада замислимо да смо, грешком, испустили редне бројеве сума и добићемо:

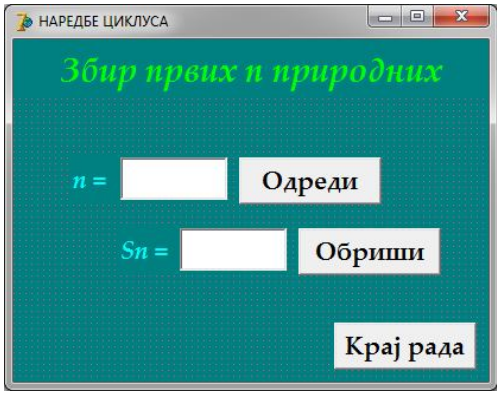
$s:=a_1$
$s:=s+a_2$
$s:=s+a_3$
...
$s:=s+a_n$

односно, свака сума се израчунава тако што се претходној суми дода следећи елемент. Ако уместо непознатих бројева ***a*** узмемо редом природне бројеве од 1 до ***n*** добићемо:

```
s:=1
s:=s+2
s:=s+3
...
s:=s+n
```

Овакав низ израза може заменити изразом:

```
s:=1;
for i:=2 to n do s:=s+i;
```



Тако смо, коначно, дошли до решења овог задатка. Битно је схватити да у изразу ***s:=s+i*** имамо ***s*** са леве стране знака додељивања које представља променљиву којој се додељује вредност и ***s*** са десне стране знака додељивања које представља променљиву која има своју вредност добијену у претходном кораку. Тако да, условно речено, ова два слова ***s*** не представљају исту ствар, само се на исти начин записују. Код исписивања програма изведену формулу ћемо модификовати због могућности да корисник програма за ***n*** упише негативан број или нулу. Зато ће почетна вредност за суму бити 0, а бројач ***i*** у циклусу ће почети од 1. Контролу уноса смо могли остварити и на други начин, на пример ако корисник унесе негативан број или 0 могли смо дефинисати ***n*** као 1 или неки други природан број (ово је, наравно, ствар избора програмера). Направимо сада форму за овај задатак као на слици и напишимо комплетан програм.

```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,i,s:integer;
begin n:=StrToIntDef(Edit1.Text,100);Edit1.Text:=IntToStr(n);
      s:=0;
      For i:=1 to n do
          s:=s+i;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
```

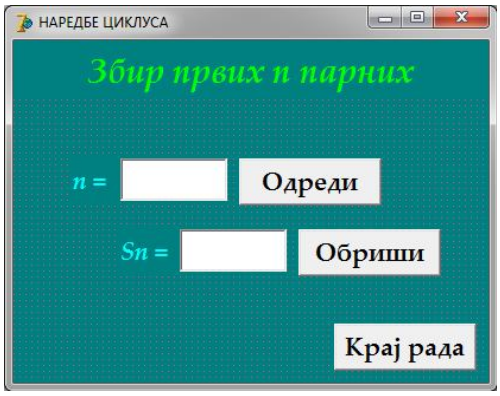
Написаћемо сада другу процедуру помоћу наредби *repeat* и *while*:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,s:integer;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<1
       then n:=1;       // контрола је неопходна због почетне вредности суме
      Edit1.Text:=IntToStr(n);
      s:=n;
      Repeat n:=n-1;    // циклус for аутоматски мења вредност бројача, а repeat не
             s:=s+n;
      Until n<=1;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,s:integer;
begin n:=StrToIntDef(Edit1.Text,1);
      // If n<1 then n:=1; ова контрола овде није неопходна јер је садржана у услову циклуса
      Edit1.Text:=IntToStr(n);
      s:=n;
      While n>1 do
      begin n:=n-1;     // циклус for аутоматски мења вредност бројача, а while не
            s:=s+n;
      end;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;,
      Button2.SetFocus;
end;
```

Приметили смо да је начин извршавања друга два циклуса нешто измењен. Наиме, сабирање се извршава уназад, тј. од ***n*** до 1. Ово је учињено да би се уштедела једна променљива и то не би требало битно да утиче на разумевање решења.

- ***Саставити програм који одређује збир првих n парних бројева.***

Разлика између овог и претходног задатака је само у томе што ћемо овде редом сабирати бројеве помножене са 2 да би били парни.




```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,s,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      s:=0;
      For i:=1 to n do
          s:=s+2*i;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Написаћемо сада другу процедуру помоћу наредби *repeat* и *while*:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,s:integer;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<1 then n:=1;       // контрола је неопходна због почетне вредности суме
      Edit1.Text:=IntToStr(n);
      s:=2*n;
      Repeat n:=n-1;
             s:=s+2*n;
      Until n<=1;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;

procedure TForm1.Button1Click(Sender: TObject);
var n,s:integer;
begin n:=StrToIntDef(Edit1.Text,1);
      // If n<1 then n:=1; ова контрола овде није неопходна јер је садржана у циклусу
      Edit1.Text:=IntToStr(n);
      s:=2*n;
      While n>1 do
      begin n:=n-1;
            s:=s+2*n;
      end;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

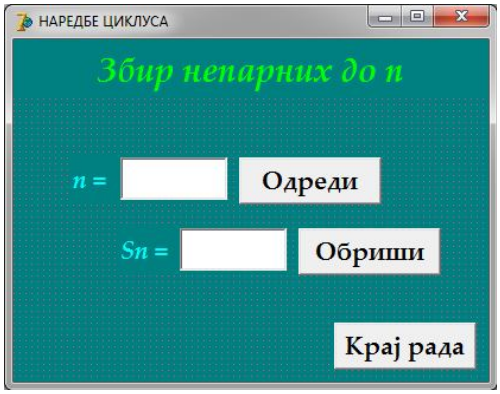
На сличан начин бисмо решили и сабирање првих ***n*** непарних бројева. Уместо ***s:=s+2*i***, односно, ***s:=s+2*n*** писали бисмо ***s:=s+2*i-1***, односно, ***s:=s+2*n-1*** и, наравно, у решењима са *repeat* и *while* почетна вредност за суму била би ***s:=2*n-1***.

- ***Саставити програм који одређује збир непарних бројева до n.***

Разлика између овог и претходног задатака је у томе што ћемо овде сабирати сваки други број, почев од броја један. Помоћу циклуса *For* пре сабирања броја треба проверити да ли је непаран, па ако јесте онда ћемо га сабирати са претходним непарним. У циклусима са *repeat* и *while* немамо потребу за овим тестирањем, већ ћемо подесити да корак буде 2, а почетна вредност бројача биће 1. Ако желимо решење као у претходном задатку, тј. без променљиве бројача, онда треба наместити да n буде непаран број, а затим сабирати уназад, сваки други број.

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,s,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      s:=0;
      For i:=2 to n do
          If Odd(i)
           then s:=s+i;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```



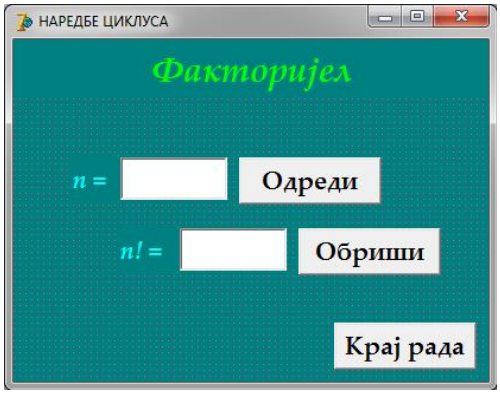
Написаћемо процедуре за непарне бројеве помоћу наредби *repeat* и *while*:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,s:integer;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<1 then n:=1;       // контрола је неопходна због изабране наредбе циклуса
      Edit1.Text:=IntToStr(n);
      If not Odd(n) then n:=n-1;    // намештамо да граница за збир буде непарна
      s:=0;
      Repeat s:=s+n; n:=n-2;
      Until n<1;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,s:integer;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<1 then n:=1;       // контрола је неопходна због почетне вредности суме
      Edit1.Text:=IntToStr(n);
      If not Odd(n) then n:=n-1;    // намештамо да граница за збир буде непарна
      s:=n;
      While n>1 do
      begin n:=n-2; s:=s+n;
      end;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Обратимо пажњу на разлику између решења са *repeat* и са *while*. Први циклус мора да се изврши бар једном и зато смо као почетну вредност узели 0, а у циклусу прво сабирали, па умањили вредност бројача. Други циклус прво тестира услов, па се тек ако треба извршава, зато је он ефикаснији (има једно извршење мање).
На сличан начин бисмо решили и сабирање парних до *n*. У циклусу *For* у услову треба додати *not*, а у решењима са *repeat* и *while*, пре уласка у циклус треба наместити да *n* буде паран број, тј. избрисати *not*.

- ***Саставити програм који одређује факторијел броја n.***

Факторијел неког броја је производ тог броја са свим природним бројевима мањим од њега. Формула за израчунавање факторијела може да се напише аналогно првом задатку: *факторијел* = $a_1 * a_2 * a_3 * \ldots * a_n$, односно, *факторијел* = 1 * 2 * 3 * ... * n, па ћемо и решење задатка написати аналогно решењу првог задатка, само ћемо уместо сабирања имати множење. Почетна вредност производа може бити 1 или сам тај број (а не сме бити 0 због множења јер би у том случају факторијел био увек 0), зависно од наредбе циклуса. По дефиницији факторијел броја 0 је 1, па и о томе треба водити рачуна. Код тестирања програма треба водити рачуна о граници целих бројева, па треба унети n<18. Граница се може померити ако уместо целих користимо реалну променљиву *f*, али не превише, јер факторијел пребрзо расте и прелази сва ограничења на рачунару. Направићемо форму за овај задатак као на слици и написати комплетан програм:

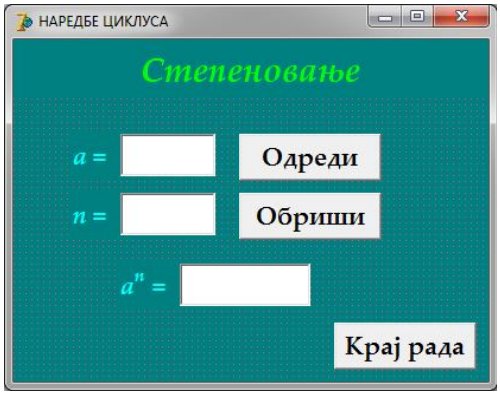


```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,f,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<0 then n:=0;
      Edit1.Text:=IntToStr(n);
      f:=1;
      For i:=2 to n do
          f:=f*i;
      Edit2.Text:=IntToStr(f);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Написаћемо другу процедуру помоћу наредби *repeat* и *while*:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,f:integer;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<0 then n:=0;
      Edit1.Text:=IntToStr(n);
      f:=1;
      Repeat f:=f*n; n:=n-1;
      Until n<1;
      Edit2.Text:=IntToStr(f);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,f:integer;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<0 then n:=0;
      Edit1.Text:=IntToStr(n);
      f:=1;
      While n>1 do
      begin n:=n-1; f:=f*n;
      end;
      Edit2.Text:=IntToStr(f);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који одређује n-ти степен броја b.***



Степеновање броја целобројним експонентом није ништа друго до производа, па решење овог задатка личи на факторијел с тим што је чинилац константа - основа степена:

степен = a * a * a * ... * a.

Најпре ћемо креирати форму као на слици (степен a^n добија се комбиновањем две лабеле, на једној пише *a*, а друга је провидна и на њој пише *n*, друга лабела је са ситнијим фонтом, мало подигнута и померена на десно), а затим написати комплетан програм:

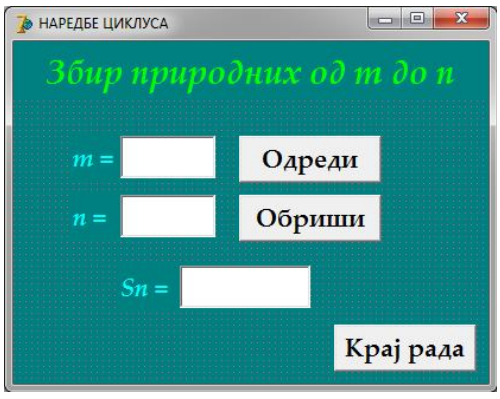
```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13 then Edit2.SetFocus;
end;
```

```
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
    b,s:real;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<0 then n:=0;
      Edit1.Text:=IntToStr(n);
      b:=StrToFloatDef(Edit2.Text,1);Edit2.Text:=FloatToStr(b);
      s:=1;
      For i:=1 to n do
          s:=s*b;
      Edit3.Text:=FloatToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Написаћемо ову процедуру помоћу *repeat* (мада је разлика само формалне природе, али имамо и додатан проблем, треба извршавати циклус само ако је експонент већи од 0, ово је код циклуса *for* било системски решено, па о томе нисмо морали да бринемо) и помоћу наредбе *while* (где разлике готово да и нема):

```
procedure TForm1.Button1Click(Sender: TObject);
var n:integer;
    b,s:real;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<0 then n:=0;
      Edit1.Text:=IntToStr(n);
      b:=StrToFloatDef(Edit2.Text,1);Edit2.Text:=FloatToStr(b);
      s:=1;
      If n>0 then // контрола је обавезна због начина извршавања циклуса
        Repeat s:=s*b;n:=n-1;
        Until n<1;
      Edit3.Text:=FloatToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n:integer;
    b,s:real;
begin n:=StrToIntDef(Edit1.Text,1);
      If n<0 then n:=0;
      Edit1.Text:=IntToStr(n);
      b:=StrToFloatDef(Edit2.Text,1);Edit2.Text:=FloatToStr(b);
      s:=1;
      While n>0 do
      begin n:=n-1;s:=s*b;
      end;
      Edit3.Text:=FloatToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који израчунава збир природних бројева између m и n, где су m и n цели позитивни бројеви.***



Задатак ћемо решити као и први, с тим што почетна вредност бројача неће бити 0, већ мањи од два уписана броја. То значи да морамо, пре уласка у циклус наместити да *m* буде мањи од уписаних бројева. Најпре ћемо креирати форму као на слици, а затим написати комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
```

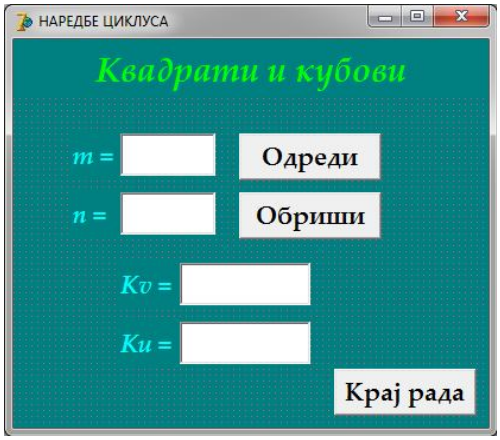
```
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var m,n,s,i:integer;
begin m:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(m);
      n:=StrToIntDef(Edit2.Text,2);Edit2.Text:=IntToStr(n);
      If m>n
       then begin s:=m;m:=n;n:=s; // замена вредности две променљиве
            end;
      s:=0;
      For i:=m to n do
          s:=s+i;
      Edit3.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Исту процедуру написаћемо и помоћу циклуса *repeat* и *while*. Погледајмо их и још једном уочимо разлику у начину размишљања код различитих врста циклуса. Наравно, могли смо и са ова два циклуса применити исти принцип као и код бројачког циклуса, али је занимљивије избећи алгоритам замене вредности две променљиве када се то може. Израз ***(n-m) div Abs(n-m)*** може имати вредност +1 или -1 у зависности од тога који је од два унета броја већи.

```
procedure TForm1.Button1Click(Sender: TObject);
var m,n,s:integer;
begin m:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(m);
      n:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(n);
      s:=n;
      If m<>n
       then Repeat s:=s+m;
                   m:=m+(n-m)div Abs(n-m);
            Until m=n;
      Edit3.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);
var m,n,s:integer;
begin m:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(m);
      n:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(n);
      s:=n;
      While m<>n do
        begin s:=s+m;
              m:=m+(n-m)div Abs(n-m);
        end;
      Edit3.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Могли смо увести и помоћну променљиву која добија вредност ***p:=(n-m) div Abs(n-m)*** пре уласка у циклус, а у циклусу уместо ***m:=m+(n-m)div Abs(n-m)*** онда стоји ***m:=m+p*** тада би се алгоритам мало убрзао или, уместо овог израза, могли смо користити наредбу: ***If n>m then p:= -1 else p:=1;***

- ***Саставити програм који израчунава збир квадрата парних и збир кубова непарних бројева између m и n, где су m и n цели позитивни бројеви.***



За разлику од претходног задатка, овде ћемо имати две променљиве које представљају резултат. Почетна вредност ће за обе бити 0, а у циклусу ће се тестирати да ли је број паран или непаран и у зависности од тога ће се одређивати збир. Најпре ћемо креирати форму као на слици, а затим написати комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit4.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
```

```
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var m,n,p,q,i:integer;
begin m:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(m);
      n:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(n);
      If m>n then
       begin s:=m;m:=n;n:=s;
       end;
      p:=0;q:=0;
      For i:=m to n do
          If Odd(i)
           then q:=q+i*i*i
           else p:=p+i*i;
      Edit3.Text:=IntToStr(p);
      Edit4.Text:=IntToStr(q);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

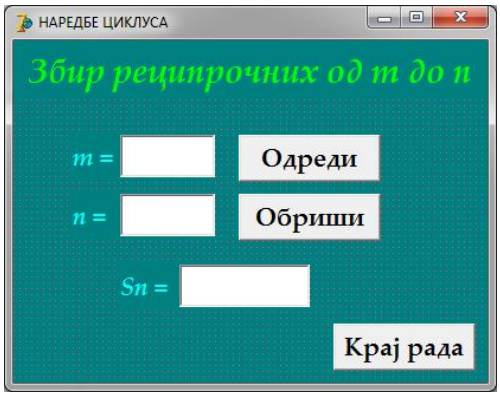
Исту процедуру написаћемо и помоћу циклуса *repeat* и *while*. Погледајмо их и још једном уочимо разлику у начину размишљања код различитих врста циклуса.

```
procedure TForm1.Button1Click(Sender: TObject);
var m,n,p,q,r:integer;
begin m:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(m);
      n:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(n);
      p:=0;q:=0;
      If m>n
       then r:=-1
       else r:=1;
      If Odd(n)
       then q:=n*n*n
       else p:=n*n;
      If m<>n then
       Repeat If Odd(m)
                then q:=q+m*m*m
                else p:=p+m*m;
              m:=m+r;
       Until m=n;
      Edit3.Text:=IntToStr(p);
      Edit4.Text:=IntToStr(q);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);
var m,n,p,q,r:integer;
begin m:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(m);
      n:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(n);
      p:=0;q:=0;
      r:=(n-m) div Abs(n-m);
      If Odd(n)
       then q:=n*n*n
       else p:=n*n;
      While m<>n do
      begin If Odd(m)
             then q:=q+m*m*m
             else p:=p+m*m;
           m:=m+r;
      end;
      Edit3.Text:=IntToStr(p);
      Edit4.Text:=IntToStr(q);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који израчунава збир реципрочних вредности бројева између m и n, где су m и n цели позитивпи бројеви.***

Пошто треба сабирати реципрочне вредности, јасно је да је сума реалан број. Даље је лако. Треба наместити да први број буде мањи, а затим, редом, сабирати реципрочне бројеве од првог до другог. Најпре ћемо креирати форму као на слици, а затим написати комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
```



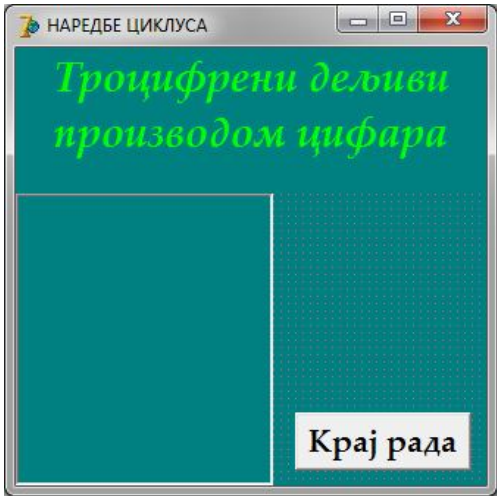

```
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var m,n,i:integer;
    s:real;
begin m:=Abs(StrToIntDef(Edit1.Text,1));  // Искључујемо негативне бројеве
      n:=Abs(StrToIntDef(Edit2.Text,1));  // Искључујемо негативне бројеве
      If m>n
       then begin s:=m;m:=n;n:=s;
            end;
      If m=0            // Контролом мањег броја искључујемо могућност дељења нулом
       then m:=1;
      Edit1.Text:=IntToStr(m);
      Edit2.Text:=IntToStr(n);
      s:=0;
      For i:=m to n do
          s:=s+1/i;
      Edit3.Text:=Format('%15.13f',[s]);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Функцијом ***Abs*** искључили смо негативне вредности граничних бројева, а дефинисали смо и да границе не могу бити 0. Прво ограничење није неопходно, јер се могу сабирати и негативни бројеви, али се поставља питање математичке исправности сабирања ако су границе супротног знака, а дељење нулом се изостави из коначног резултата (На пример: ако се сабирају реципрочне вредности бројева од -3 до 5 треба сабрати реципрочне вредности свих бројева, тј. не може се прескочити 0, а реципрочна вредност за 0 није дефинисана). Функцијом ***Format*** дефинисали смо да се резултат испише на 15 места као реалан број са 13 децимала. Исту процедуру написаћемо и помоћу циклуса *repeat* и *while*. Погледајмо их и још једном уочимо разлику у начину размишљања код различитих врста циклуса.

```
procedure TForm1.Button1Click(Sender: TObject);
var m,n,p:integer;
    s:real;
begin m:=Abs(StrToIntDef(Edit1.Text,1));
      If m=0
       then m:=1;
      Edit1.Text:=IntToStr(m);
      n:=Abs(StrToIntDef(Edit2.Text,1));
      If n=0
       then n:=1;
      Edit2.Text:=IntToStr(n);
      s:=1/n;
      p:=(n-m)div Abs(n-m);
      If m<>n then
       Repeat s:=s+1/m;m:=m+p;
       Until m=n;
      Edit3.Text:=Format('%15.13f',[s]);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;

procedure TForm1.Button1Click(Sender: TObject);
var m,n,p:integer;
    s:real;
begin m:=Abs(StrToIntDef(Edit1.Text,1));
      If m=0
       then m:=1;
      Edit1.Text:=IntToStr(m);
      n:=Abs(StrToIntDef(Edit2.Text,1));
      If n=0
       then n:=1;
      Edit2.Text:=IntToStr(n);
      s:=1/n;
      p:=(n-m)div Abs(n-m);
      While m<>n do
       begin s:=s+1/m;m:=m+p;
```

```
       end;
      Edit3.Text:=Format('%15.13f',[s]);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који исписује све троцифрене бројеве који су дељиви производом својих цифара (водити рачуна да не дође до дељења са нулом).***



Код овог задатака нећемо имати унос улазних података јер треба проверити услов за све троцифрене бројеве. Зато ће и форма бити нешто другачија, једноставнија. Комплетан програм ће се аутоматски извршавати, без тастера *Одреди* и *Обриши*. За испис резултата користићемо мемо поље које ће бити само за читање. Тестирање ћемо започети од броја 111 јер сви претходни троцифрени бројеви садрже цифру 0 и не могу бити дељиви производом својих цифара, који је 0, па дељење нема смисла. У прозору ***Object Tree View*** ћемо двокликом селектовати објекат ***Form1*** (или једним кликом на форму селектовати објекат, а затим у ***Object Inspector***-у на листићу ***Events*** два пута кликнути у поље вредности догађаја ***OnCreate*** и написати процедуру:

```
procedure TForm1.FormCreate(Sender: TObject);
var i,c1,c2,c3,p:integer;
begin Memo1.Clear;
      For i:=111 to 999 do
      begin c1:=i mod 10;
            c2:=i div 10 mod 10;
            c3:=i div 100;
            p:=c1*c2*c3;
            If (p>0)and(i mod p=0) then Memo1.Text:=Memo1.Text+IntToStr(i)+' ';
      end;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin Application.Terminate;
end;
```

Ако желимо да се бројеви исписују један испод другог, уместо команде:

```
Memo1.Text:=Memo1.Text+IntToStr(i)+' ';
```

унос у мемо поље остваримо командом:

```
Memo1.Lines.Add(IntToStr(i)); или Memo1.Lines.Append(IntToStr(i));
```

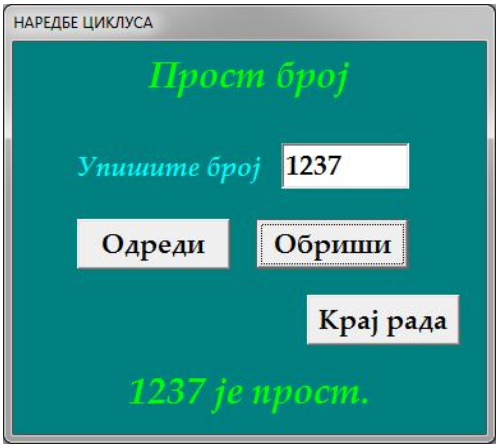
Написаћемо ову процедуру и помоћу наредби *repeat* и *while*, али разлика је готово безначајна:

```
procedure TForm1.FormCreate(Sender: TObject);
var i,c1,c2,c3,p:integer;
begin Memo1.Clear; i:=111;
      Repeat c1:=i mod 10;
             c2:=i div 10 mod 10;
             c3:=i div 100;
             p:=c1*c2*c3;
             If (p>0)and(i mod p=0) then Memo1.Lines.Text:=Memo1.Lines.Text+IntToStr(i)+' ';
             i:=i+1;
      until i=1000;
end;
procedure TForm1.FormCreate(Sender: TObject);
var i,c1,c2,c3,p:integer;
begin Memo1.Clear; i:=111;
      While i<1000 do
      begin c1:=i mod 10;
            c2:=i div 10 mod 10;
            c3:=i div 100;
            p:=c1*c2*c3;
            If (p>0)and(i mod p=0) then Memo1.Lines.Text:=Memo1.Lines.Text+IntToStr(i)+' ';
            i:=i+1;
      end;
end;
```

- ***Саставити програм који одређује да ли је број прост (број је прост ако осим себе самог и јединице нема других делилаца).***

Пошто су сви бројеви дељиви са 1 и самим собом те делиоце нећемо ни проверавати. У решавању задатка можемо користити бројач делилаца - целобројну променљиву која ће се увећавати за 1 кад год нађемо неки број који је делиоц уписаног броја. Ако је тај број различит од 0 број није прост. Друга могућност је логичка променљива. Уписани број ћемо делити бројевима од 2 до половине уписаног броја, ако је неки од бројева дели уписани број логичка променљива ***p*** ће добити вредност ***false***, односно, број није прост (почетна вредност променљиве ***p*** је ***true***). Најпре ћемо креирати форму као на слици, а затим, написати комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
    p:boolean;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      p:=true;
      For i:=2 to n div 2 do
        If n mod i=0
         then p:=false;
      If p
       then Label3.Caption:=Edit1.Text+' је прост.'
       else Label3.Caption:=Edit1.Text+' није прост.';
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```



Овај алгоритам је неефикасан јер, на пример, ако је број паран и већи је од 2 не може бити прост, а ми ћемо, свеједно, делити са свим бројевима до половине унетог броја, да бисмо на крају написали да број није прост. Али то је проблем бројачких циклуса који се не могу прекидати пре извршења задатог броја циклуса. Написаћемо процедуре за одређивање простог броја помоћу наредби *repeat* и *while* коришћењем бржег алгоритма:

Ако је број мањи од 4 он је прост, ако је број паран он није прост, у осталим случајевима тестираћемо потенцијалне делиоце све док не нађемо на првог или док не стигнемо до корена броја који се тестира.

Корен унетог броја је највећи потенцијални делилац који тестирамо јер сваки делилац већи корена, да би се добио почетни број, треба помножити неким мањим од корена, а њега смо већ тестирали. Овим смо убрзали излазак из процедуре и ако је број прост и ако није јер има много мање проверавања него са *for*.

```
procedure TForm1.Button1Click(Sender: TObject);
var d,n:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      Label3.Caption:='Broj nije prost';
      If n<4
       then Label3.Caption:='Broj je prost'
       else If Odd(n) then
             begin d:=1;
                   Repeat d:=d+2;
                   until (d>Sqrt(n))or(n mod d=0);
                   If d>Sqrt(n)
                    then Label3.Caption:='Broj je prost';
             end;
      Edit1.ReadOnly:=true;Button2.SetFocus;
end;
```

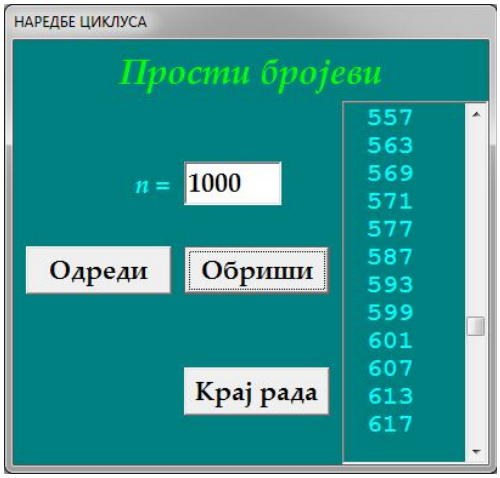
или:

```
procedure TForm1.Button1Click(Sender: TObject);
var d,n:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      Label3.Caption:='Broj nije prost';
      If n<4
       then Label3.Caption:='Broj je prost'
       else If Odd(n) then
             begin d:=3;
                   While (d<=Sqrt(n))and(n mod d<>0) do d:=d+2;
                   If d>Sqrt(n)
                          then Label3.Caption:='Broj je prost';
             end;
      Edit1.ReadOnly:=true;Button2.SetFocus;
end;
```

- ***Саставити програм који одређује све просте бројеве до n.***

У решавању овог задатка ћемо искористити претходни задатак тако што ћемо у циклусу од 1 до уписаног броја проверавати да ли је број прост и уместо да исписујемо поруку уписиваћемо просте бројеве у мемо поље. За исписивање простих бројева користићемо формат наредбу, ако желимо да резултат буде прегледан. Најпре ћемо креирати форму, а затим ћемо написати комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,i,b:integer;
    p:boolean;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For b:=1 to n do
      begin p:=true;
            For i:=2 to b div 2 do
                p:=b mod i=0;
            If p then Memo1.Lines.Add(Format('%5d',[b]));
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```



Ако желимо да се бројеви исписују један поред другог, уместо:

```
Memo1.Lines.Add(Format('%5d',[b]));
```

унос у мемо поље остваримо командом:

```
Memo1.Text:=Memo1.Text+Format('%5d ',[b]);
```

И овде стоји примедба да алгоритам са два *for* циклуса није ефикасан, с тим да се то овде још више осећа јер се не тестира један већ n бројева, па је разлика у брзини извршења овог алгоритма у односу на алгоритам са циклусима са условом много приметнија. Први циклус *for* је мања сметња ефикасности. Написаћемо ову процедуру помоћу наредби циклуса *repeat* и *for*:

```
procedure TForm1.Button1Click(Sender: TObject);
var d,n,b:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For b:=1 to n do
      begin d:=1;
            Repeat d:=d+1;
            until (d>Sqrt(b))or(b mod d=0);
            If d>Sqrt(b)
             then Memo1.Lines.Add(Format('%5d',[b]));
       end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или са две наредбе циклуса *repeat*:

```
procedure TForm1.Button1Click(Sender: TObject);
var d,n,b:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      If n>0
       then Memo1.Lines.Add(Format('%5d',[1]));
      If n>1
       then Memo1.Lines.Add(Format('%5d',[2]));
      If n>2
       then begin b:=3;
                  Repeat d:=1;
                   Repeat d:=d+1;
                   until (d>Sqrt(b))or(b mod d=0);
                   If d>Sqrt(b)
                    then Memo1.Lines.Add(Format('%5d',[b]));
                   b:=b+2;
                  Until b>n;
            end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или помоћу наредби циклуса *while* и *for*:

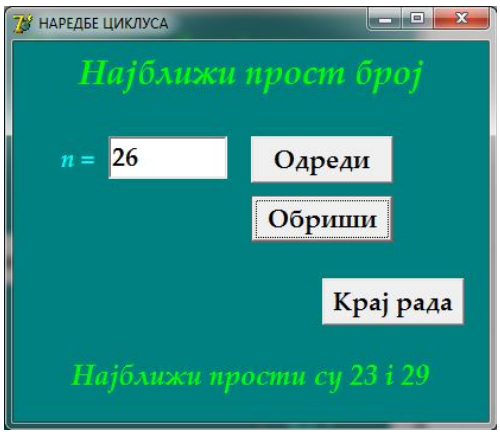
```
procedure TForm1.Button1Click(Sender: TObject);
var d,n,b:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For b:=1 to n do
      begin d:=2;
            While (d<=Sqrt(b))and(b mod d<>0) do
                   d:=d+1;
```

```
            If d>Sqrt(b)
             then Memo1.Lines.Add(Format('%5d',[b]));
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или са две наредбе циклуса *while*:

```
procedure TForm1.Button1Click(Sender: TObject);
var d,n,b:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      If n>0
       then Memo1.Lines.Add(Format('%5d',[1]));
      If n>1
       then Memo1.Lines.Add(Format('%5d',[2]));
      If n>2
       then begin b:=3;
                  While b<=n do
                  begin d:=2;
                        While (d<=Sqrt(b))and(b mod d<>0) do
                              d:=d+1;
                        If d>Sqrt(b)
                         then Memo1.Lines.Add(Format('%5d',[b]));
                        b:=b+2;
                  end;
            end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који одређује прост број најближи уписаном броју.***



Овај задатак је мало теже решити помоћу циклус *for* (али није немогуће). Зато ће овде бити приказана решења са друга два циклуса. Идеја је да се нађе први прост број мањи и већи од уписаног броја, а затим да се одреди који је од њих ближи уписаном броју. Ово се може урадити на више различитих начина, али најефикаснији и најефектнији је:

полазимо од уписаног броја, тестирамо да ли је прост, ако јесте то је тражени број, а ако није

смањимо га за 1 и повећамо га за 1, па оба овако добијена броја тестирамо, ако је један од њих

прост - то је тражени број, ако су оба броја проста у том случају потпуно је исправно исписати

мањи или већи од њих или исписати оба броја, ако се у задатку не каже прецизно, ако ниједан није

прост понављамо поступак све док не нађемо бар један прост број.

У решењу које ћемо приказати исписиваће се оба броја ако су прости. Најпре ћемо креирати форму, а затим написати комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var a,b,d,n:integer;
    r:string;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      a:=n-1; b:=n+1; r:='';
      Repeat d:=1;a:=a+1;
             Repeat d:=d+1;
             until (d>Sqrt(a))or(a mod d=0);
             If d>Sqrt(a)           // већи број је прост
              then r:=IntToStr(a);
             d:=1;b:=b-1;
             Repeat d:=d+1;
             until (d>Sqrt(b))or(b mod d=0);
```

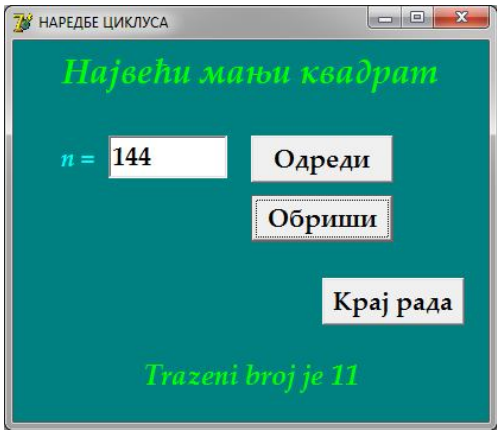
```
            If d>Sqrt(b)           // мањи број је прост
             then If (Length(r)>0)and(a<>b)          // и мањи и већи број је прост
                   then r:='Најближи прости су '+IntToStr(b)+' i '+r
                   else r:='Најближи прост је '+IntToStr(b) // уписани или мањи број је прост
             else If Length(r)>0
                   then r:='Најближи прост је '+r;   // мањи број је прост
      until Length(r)>0;
      Label3.Caption:=r;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,d,n:integer;
    r:string;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      a:=n-1;b:=n+1;r:='';
      While Length(r)=0 do
      begin d:=2;a:=a+1;
            While (d<=Sqrt(n))and(n mod d<>0) do
                  d:=d+1;
            If d>Sqrt(a)
             then r:=IntToStr(a);
            d:=2;b:=b-1;
            While (d<=Sqrt(n))and(n mod d<>0) do
                  d:=d+1;
             If d>Sqrt(b)           // мањи број је прост
              then If (Length(r)>0)and(a<>b)          // и мањи и већи број је прост
                    then r:='Најближи прости су '+IntToStr(b)+' i '+r
                    else r:='Најближи прост је '+IntToStr(b) // уписани или мањи број је прост
              else If Length(r)>0
                    then r:='Најближи прост је '+r;   // мањи број је прост
      end;
      Label3.Caption:=r;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Јасно је да је ово само један од начина да се дође до решења. Пробајте да задатак решите и другачије.

- ***Саставити програм који одређује највећи број чији је квадрат мањи од уписаног броја.***



Пустићемо циклус од 1 до уписаног броја и тестирати да ли је квадрат бројача мањи од уписаног броја и ако јесте исписаћемо га. Последњи уписани број је тражени. Решење са циклусом ***for*** није програмерски сасвим коректно јер, у ствари, исписује све квадрате мање од уписаног броја, а на крају се види само последњи (који јесте највећи и тражени) јер се сви они исписују у исту лабелу. Коректно решење може да се добије без примене циклуса по формули:

```
i:=Trunc(Sqrt(n));
```

Међутим, овде се трудимо да схватимо циклусе, па отуда и овакво решење. Решења са циклусима ***repeat*** и ***while*** су коректнија. Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;
```

```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For i:=1 to n do
       If i*i<n then
        Label3.Caption:='Trazeni broj je '+IntToStr(i);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

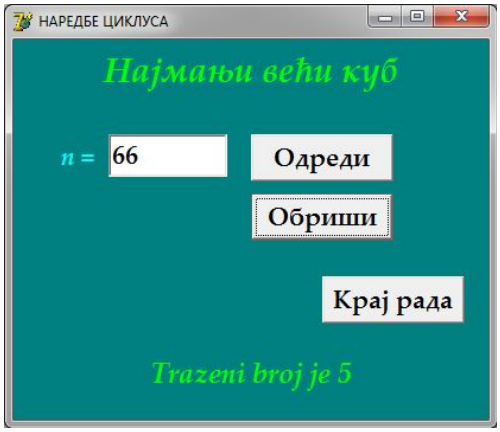
Написаћемо ову процедуру и помоћу наредби *repeat* и *while*, али разлика је готово безначајна:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      i:=n+1;
      Repeat i:=i-1
      until i*i<n;
      Label3.Caption:='Trazeni broj je '+IntToStr(i);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      i:=n;
      While i*i>=n do i:=i-1;
      Label3.Caption:='Trazeni broj je '+IntToStr(i);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који одређује најмањи број чији је куб већи од уписаног броја.***



Пустићемо циклус од уписаног броја до 1 и тестирати да ли је куб бројача већи од уписаног броја и ако јесте исписаћемо га. Последњи уписани број је тражени. Решење са циклусом *for* није програмерски сасвим коректно јер, у ствари, исписује све кубове веће од уписаног броја, а на крају се види само последњи (који јесте најмањи и тражени) јер се сви они исписују у исту лабелу. Коректно решење може да се добије без примене циклуса по формули:

```
i:=Round(Exp(1/3*log(n)+0.51);
```

Међутим овде се трудимо да схватимо циклусе, па отуда и овакво решење. Решења са циклусима *repeat* и *while* су коректна. Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
            then Button1.Click;
end;

procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For i:=n downto 1 do
       If i*i*i>n then Label3.Caption:='Trazeni broj je '+IntToStr(i);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Написаћемо ову процедуру и помоћу наредби *repeat* и *while*, али разлика је готово безначајна:

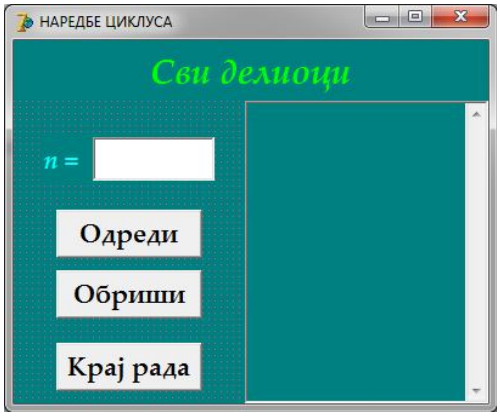
```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      i:=0;
      Repeat i:=i+1;
      Until i*i*i>n;
      Label3.Caption:='Trazeni broj je '+IntToStr(i);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      i:=0;
```

```
      While i*i*i<=n do
            i:=i+1;
      Label3.Caption:='Trazeni broj je '+IntToStr(i);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који исписује све делиоце уписаног броја***



У циклусу *for* од 1 до уписаног броја тестираћемо да ли је неки од бројача делилац уписаног броја и ако јесте уписаћемо га у мемо поље. Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For i:=1 to n do
        If n mod i=0
         then Memo1.Lines.Add(IntToStr(i));
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Много брже се долази до решења ако се исписују парови делилаца, али онда они нису уређени у растући поредак. Ако нам то не смета, онда је ово сасвим прихватљиво решење:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For i:=1 to Trunc(Sqrt(n)) do
        If n mod i=0
         then Memo1.Lines.Add(Format('%5d,%5d',[i,n div i]));
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Решење се може добити и применом неког од циклуса са условом мада је решење са *for* најефикасније:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      i:=1;
      Repeat If n mod i=0
              then Memo1.Lines.Add(IntToStr(i));
             i:=i+1;
      Until i>=n;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

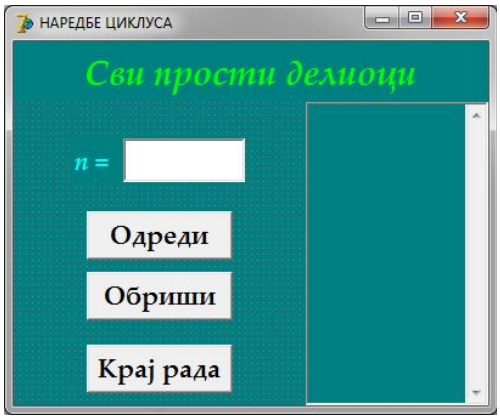
```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      i:=0;
      While i<n do
      begin i:=i+1;
            If n mod i=0
             then Memo1.Lines.Add(IntToStr(i));
      end;
      Edit1.ReadOnly:=true;Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      i:=0;
      While i<Sqrt(n) do
```

```
      begin i:=i+1;
            If n mod i=0
             then Memo1.Lines.Add(Format('%5d,%5d',[i,n div i]));
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који исписује све просте делиоце уписаног броја.***



Задатак је јако тешко решити само помоћу циклуса *for* циклуса, зато ћемо га решавати уз помоћ два циклуса са условом. Као у претходном задатку треба одредити делиоце уписаног броја у првом циклусу, а затим проверити да ли су они прости бројеви у другом циклусу, па ако јесу исписати их у мемо поље. Међутим, овакво решење није ефикасно. Уместо провере да ли је делилац прост, можемо унети број делити делиоцем све док је он њиме дељив. На тај начин елиминишемо све оне делиоце који су целобројни умножак било ког другог делиоца, односно, елиминишемо све делиоце који нису прости бројеви. Најпре ћемо креирати форму, а затим написати и комплетан програм:

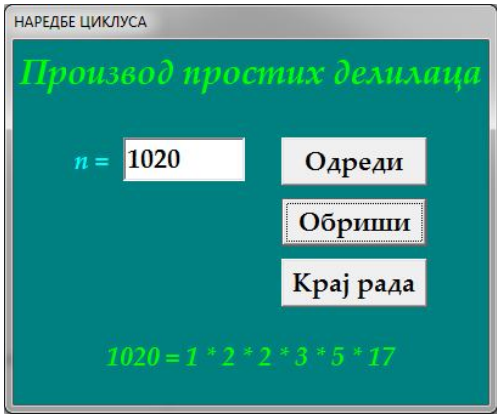
```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,d:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      Memo1.Lines.Add('1');
      d:=2;
      Repeat If n mod d=0 then
             begin Memo1.Lines.Add(IntToStr(d));
                   Repeat n:=n div d;
                   until n mod d<>0;
             end;
             d:=d+1;
      Until d>n;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,d:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      Memo1.Lines.Add('1');
      d:=2;
      While d<=n do
        If n mod d=0
         then begin Memo1.Lines.Add(IntToStr(d));
                    While n mod d=0 do n:=n div d;
              end
         else d:=d+1;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који уписани број исписује као производ простих делилаца.***

Задатак је јако тешко решити помоћу циклуса ***for*** зато ћемо га решити само помоћу циклуса са условом, прво уз помоћ циклуса ***repeat***. У претходном задатку користили смо метод одређивања простих делилаца дељењем док је број дељив са тим делиоцем, а исписивали смо га само једном. Сада ћемо, сваки пут када се уписани број дели исписивати делилац све док је могуће делити број. Да би све то личило на производ додаћемо * између сваког исписа делиоца. Резултат ћемо исписати у лабели на дну форме. У решењу је коришћена стринг променљива ***p*** која се могла изоставити ако би се уместо ње писало ***Label3.Caption***, али овако је решење лепше. Најпре ћемо креирати форму, а затим написати и комплетан програм:

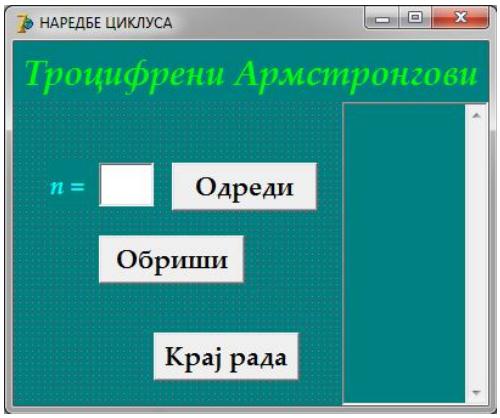



```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var b,d:integer;
    p:string;
begin b:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(b);
      p:=IntToStr(b)+' = 1';
      If b>1 then
      begin d:=2;
            Repeat If b mod d=0 then
                    Repeat b:=b div d;
                           p:=p+' * '+IntToStr(d);
                    Until b mod d<>0;
                  d:=d+1;
            Until b=1;
      end;
      Label3.Caption:=p;
      Edit1.ReadOnly:=true;Button2.SetFocus;
end;
```

Једноставније и елегантније решење је помоћу циклуса ***while***:

```
procedure TForm1.Button1Click(Sender: TObject);
var b,d:integer;
begin b:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(b);
      Label3.Caption:=IntToStr(b)+' = 1';
      d:=2;
      While b<>1 do
      begin While b mod d=0 do
            begin b:=b div d;
                  Label3.Caption:=Label3.Caption+' * '+IntToStr(d);
            end;
            d:=d+1;
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који одређује све Армстронгове бројеве до n.***



Неки број је Армстронгов ако је збир свих његових цифара степенованих бројем цифара једнак том броју. Решење које овде приказујемо важи само за троцифрене бројеве, тј. овде ћемо Армстронговим звати троцифрене бројеве који су једнаки збиру кубова својих цифара. Зато ћемо едит ограничити на дужину 3. Касније ћемо, још једном, решавати овај задатак и тада ће бити приказано решење које се може применити на све бројеве. За сваки број у циклусу ћемо одредити цифре, њихове кубове и проверити да ли је збир кубова једнак почетном броју. Ако јесте додаћемо тај број у мемо поље. Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,b,c1,c2,c3,z:integer;
```

```
begin n:=StrToIntDef(Edit1.Text,999);
      If n<100
       then n:=999;
      Edit1.Text:=IntToStr(n);
      For b:=100 to n do
      begin c1:=b mod 10;
            c2:=b div 10 mod 10;
            c3:=b div 100;
            z:=c1*c1*c1+c2*c2*c2+c3*c3*c3;
            If b=z
             then Memo1.Lines.Add(IntToStr(b));
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Јасно је да је овакво решење коректно, али су решења са циклусима са условом неупоредиво лепша и ефектнија:

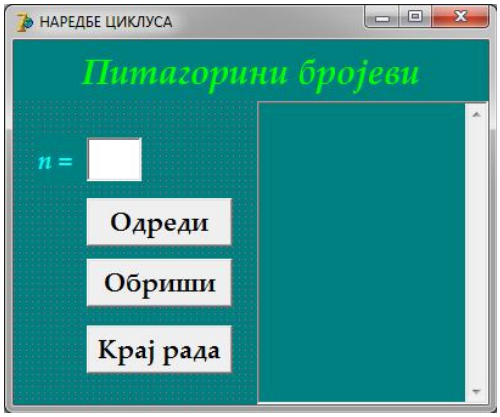
```
procedure TForm1.Button1Click(Sender: TObject);
var n,b,p,c,z:integer;
begin n:=StrToIntDef(Edit1.Text,999);
      If n<100
       then n:=999;
      Edit1.Text:=IntToStr(n);
      b:=100;
      Repeat p:=b;z:=0;
             Repeat c:=p mod 10;
                    z:=z+c*c*c;
                    p:=p div 10;
             until p=0;
             If b=z
              then Memo1.Lines.Add(Format('%4d',[b]));
             b:=b+1;
      until b=n;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,b,p,c,z:integer;
begin n:=StrToIntDef(Edit1.Text,999);
      If n<100
       then n:=999;
      Edit1.Text:=IntToStr(n);
      b:=1;
      While b<>n do
      begin p:=b;z:=0;
            While p<>0 do
            begin c:=p mod 10;
                  z:=z+c*c*c;
                  p:=p div 10;
            end;
            If b=z
             then Memo1.Lines.Add(Format('%4d',[b]));
            b:=b+1;
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који одређује Питагорине бројеве мање од n.***

Три броја су Питагорина ако могу бити мерни бројеви страница правоуглог троугла. То значи да ћемо за три броја проверити да ли важи Питагорина теорема. Ако важи исписаћемо ту тројку у мемо поље. Приликом формирања сваког од три циклуса морамо водити рачуна да је хипотенуза највећа, а да катете не могу бити целобројне и једнаке. Такође треба водити рачуна и да не дође до понављања тројки, тј. тројка 3, 4, 5 је иста као и тројка 4, 3, 5 и треба да се испише само једна од њих. Најпре ћемо креирати форму, а затим написати и комплетан програм:




```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
```

```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,a,b,c:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For a:=1 to n-2 do
       For b:=a+1 to n-1 do
        For c:=b+1 to n do
         If a*a+b*b=c*c
          then Memo1.Lines.Add(Format('%3d %3d %3d',[a,b,c]));
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Овакво решење је коректно и најједноставније, али написаћемо и решења са циклусима са условом:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,a,b,c:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      a:=1;
      Repeat b:=a+1;
             Repeat c:=b+1;
                    Repeat If a*a+b*b=c*c
                            then Memo1.Lines.Add(Format('%3d %3d %3d',[a,b,c]));
                           c:=c+1;
                    until c>n;
                    b:=b+1;
             until b>n-1;
             a:=a+1;
      until a>n-2;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,a,b,c:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      a:=1;
      While a<=n-2 do
      begin b:=a+1;
            While b<=n-1 do
            begin c:=b+1;
                  While c<=n do
                  begin If a*a+b*b=c*c
                         then Memo1.Lines.Add(Format('%3d %3d %3d',[a,b,c]));
                        c:=c+1;
                  end;
                  b:=b+1;
            end;
            a:=a+1;
      end;
      Edit1.ReadOnly:=true;Button2.SetFocus;
end;
```

- ***Саставити програм који одређује све савршене бројеве до n.***

Савршени бројеви су бројеви који су једнаки збиру својих правих делилаца. То значи, да бисмо решили овај задатак треба одредити све праве делиоце неког броја (делиоце који су различити од самог тог броја) и проверити да ли је њихов збир једнак почетном броју. Ако јесте исписаћемо га у мемо поље. Најпре ћемо креирати форму, а затим написати и комплетан програм:


```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
```

```
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,b,d,s:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For b:=1 to n do
      begin s:=1;
            For d:=2 to b div 2 do
              If b mod d=0
               then s:=s+d;
            If s=b
             then Memo1.Lines.Add(Format('%4d',[b]));
      end;
      Edit1.ReadOnly:=true;Button2.SetFocus;
end;
```

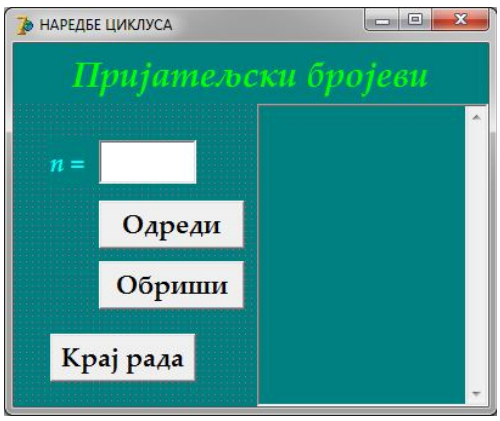
Овакво решење је коректно и најједноставније, али написаћемо и решења са циклусима са условом:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,b,d,s:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      b:=1;
      Repeat s:=1;d:=2;
             Repeat If b mod d=0
                     then s:=s+d;
                    d:=d+1;
             until d>n div 2;
             If s=b
              then Memo1.Lines.Add(Format('%4d',[b]));
             b:=b+1;
      until b>n;
      Edit1.ReadOnly:=true;Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,b,d,s:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      b:=1;
      While b<=n do
      begin s:=1;d:=2
            While d<=b div 2 do
            begin If b mod d=0
                   then s:=s+d;
                  d:=d+1;
            end;
            If s=b
             then Memo1.Lines.Add(Format('%4d',[b]));
            b:=b+1;
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који одређује парове пријатељских бројева до n.***



Два броја су пријатељска ако је збир правих делилаца сваког једнак оном другом броју. Да бисмо решили овај задатак пустићемо да један број буде слободно изабран. Затим ћемо одредити његов збир правих делилаца и њега прогласити потенцијалним бројем пријатељем првог броја. Да би он стварно то и био одредићемо његов збир правих делилаца и проверити да ли је једнак почетном броју. Ако јесте то су два пријатељска броја. Код исписивања треба водити рачуна да се парови не понављају. То ћемо урадити додавањем услова да први број буде мањи од другог. Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var n,a,d,s,b:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      For a:=1 to n do
      begin b:=1;
            For d:=2 to a div 2 do
              If a mod d=0 then b:=b+d;
            s:=1;
            For d:=2 to b div 2 do
              If b mod d=0 then s:=s+d;
            If (s=a)and(a<b) then Memo1.Lines.Add(Format('%5d %5d',[a,b]));
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

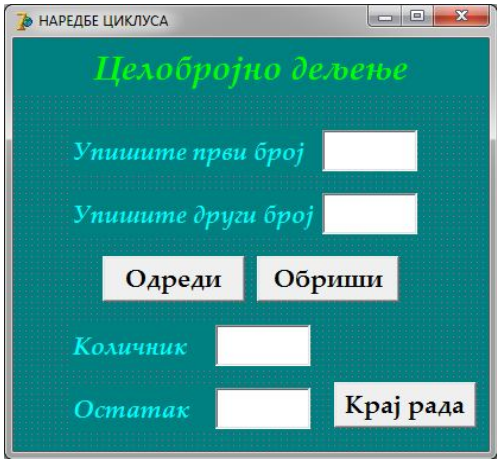
Овакво решење је коректно и најједноставније, али написаћемо и решења са циклусима са условом:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,a,d,s,b:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      a:=1;
      Repeat b:=1;d:=2;
             Repeat If a mod d=0 then b:=b+d;
                    d:=d+1;
             until d>a div 2;
             s:=1;d:=2;
             Repeat If b mod d=0 then s:=s+d;
                    d:=d+1;
             until d>b div 2;
             If (s=a)and(a<b) then Memo1.Lines.Add(Format('%5d %5d',[a,b]));
             a:=a+1;
      until a>n;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,a,d,s,b:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      a:=1;
      While a<=n do
      begin b:=1;d:=2;
            While d<=a div 2 do
            begin If a mod d=0 then b:=b+d;
                  d:=d+1;
            end;
            s:=1;d:=2;
            While d<=b div 2 do
            begin If b mod d=0 then s:=s+d;
                  d:=d+1;
            end;
            If (s=a)and(a<b) then Memo1.Lines.Add(Format('%5d %5d',[a,b]));
            a:=a+1;
      end;
      Edit1.ReadOnly:=true;Button2.SetFocus;
end;
```

- ***Саставити програм који одређује количник и остатак при целобројном дељењу два уписана цела броја користећи само операције сабирања и одузимања.***



Од првог броја одузимаћемо други. Сваки пут када одузмемо други број, количник ћемо увећати за један. Када више не буде могло да се одузима, односно, када први број постане мањи од другог, број који је остао од почетног представља остатак при целобројном дељењу. Задатак је немогуће решити помоћу циклуса *for* јер треба од првог броја одузимати други све док се не добије резултат који је мањи од првог броја, број одузимања је непознат, па се не може поставити граница за бројач. Зато следе само решења са циклусима *repeat* и *while*. Најпре ћемо креирати форму, а затим написати и комплетан програм:

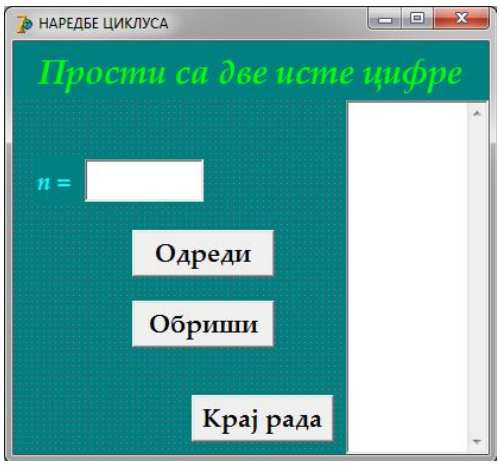
```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit4.Clear;
      Edit1.SetFocus;
end;
```

```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c:integer;
begin a:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(b);
      c:=0;
      If a>b
       then Repeat a:=a-b;c:=c+1;
            until a<b;
      Edit3.Text:=IntToStr(c);
      Edit4.Text:=IntToStr(a);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c:integer;
begin a:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(b);
      c:=0;
      While a>=b do
      begin a:=a-b;c:=c+1;
      end;
      Edit3.Text:=IntToStr(c);
      Edit4.Text:=IntToStr(a);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који исписује све просте бројеве мање од 1000 са две исте цифре.***

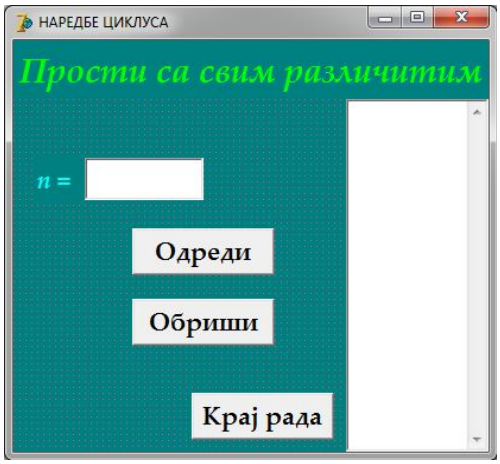


Јасно је да, осим 11, нема двоцифрених простих бројева са две исте цифре јер су сви такви дељиви баш са 11. Зато ћемо тестирање почети од 101, а 11 ћемо исписати као почетну вредност садржаја мемо поља. Задатак ћемо проширити тако да граница не буде број 1000, већ да се може изабрати било која граница. Јасно је да се до решења може доћи на више различитих начина, а овде ће бити приказано једно од њих. За сваки број већи од 100 проверићемо да ли је прост, па ако јесте одредићемо његове цифре и од њих формирати скуп цифара. Код сваког додавања цифре у скуп, прво ћемо проверити да ли се цифра налази у скупу. Ако је цифра у скупу онда је испуњен услов задатка, па ћемо га додати у мемо поље и одмах прелазимо на тестирање следећег простог броја. Задатак ћемо решити комбиновањем различитих наредби циклуса јер се тако добија најефикасније решење. Циклусом *repeat* ћемо проћи кроз непарне бројеве од 100 до уписаног броја, а циклусима *while* ћемо одређивати да ли је број прост и да ли има бар две исте цифре. Задатак нећемо решавати помоћу бројачког циклуса јер је спорији (зато што не можемо да искључимо из тестирања парне бројеве). Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var n,a,b,c,d:integer;
    p:boolean;
    sc:set of 1..9;
begin n:=StrToIntDef(Edit1.Text,1000);Edit1.Text:=IntToStr(n);
      If n>10 then Memo1.Lines.Add('11');
      If n>100
       then begin b:=101
                  Repeat d:=2;a:=b;
                         While (a mod d<>0)and(d<=Sqrt(a)) do d:=d+1;
                         If d>Sqrt(a)
                          then begin c:=a mod 10;
                                     a:=a div 10;
                                     sc:=[c];
                                     p:=false;
                                     While (a<>0)and(not p) do
                                     begin c:=a mod 10;
                                           a:=a div 10;
                                           If c in sc
                                            then p:=true
                                            else sc:=sc+[c];
                                     end;
                                     If p then Memo1.Lines.Add(IntToStr(b));
                               end;
                         b:=b+2;
                  until b>n;
            end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који исписује све просте бројеве мање од 1000 са свим различитим цифрама.***

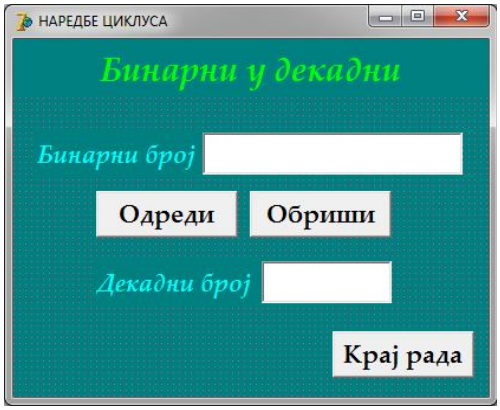


Јасно је да, осим 11, сви други прости бројеви до 100 не могу имати исте цифре. Зато ћемо тестирање почети од 101, а све претходне просте ћемо исписати као почетну вредност садржаја мемо поља. Задатак ћемо проширити тако да граница не буде број 1000, већ да се може изабрати било која граница. Јасно је да се до решења може доћи на више различитих начина, а овде ће бити приказано једно од њих. За сваки број већи од 100 проверићемо да ли је прост, па ако јесте одредићемо његове цифре и од њих формирати скуп цифара. Код сваког додавања цифре у скуп, прво ћемо проверити да ли се цифра налази у скупу. Ако је цифра у скупу онда није испуњен услов задатка, па га нећемо додати у мемо поље, већ тражимо следећи прост број. Задатак ћемо решити са три *while* циклуса јер се тако добија најефикасније решење. Првим циклусом ћемо проћи кроз непарне бројеве од 3 до уписаног броја. Другим циклусом ћемо одређивати да ли је број прост, а трећим да ли има истих цифара. Задатак нећемо решавати помоћу бројачког циклуса јер је спорији (зато што не можемо да искључимо из тестирања парне бројеве). Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,a,b,c,d:integer;
    p:boolean;
    sc:set of 1..9;
begin n:=StrToIntDef(Edit1.Text,1000);Edit1.Text:=IntToStr(n);
      If n>0 then Memo1.Lines.Add(IntToStr(1));
      If n>1 then Memo1.Lines.Add(IntToStr(2));
      b:=3;
      While b<=n do
      begin d:=2;a:=b;
            While (a mod d<>0)and(d<=Sqrt(a)) do d:=d+1;
```

```
            If d>Sqrt(a)
             then If (b<100)and(b<>11)
                   then Memo1.Lines.Add(IntToStr(b))
                   else begin c:=a mod 10;
                              a:=a div 10;
                              sc:=[c];
                              p:=true;
                              While (a<>0)and(p) do
                              begin c:=a mod 10;
                                    a:=a div 10;
                                    If c in sc
                                     then p:=false
                                     else sc:=sc+[c];
                              end;
                              If p then Memo1.Lines.Add(IntToStr(b));
                        end;
            b:=b+2;
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који бинарни број претвара у декадни.***



Задатак ћемо решити тако што ћемо сваку цифру бинарног броја множити бројем 2 степенованим редним бројем места бинарне цифре и сабирати тако добијене резултате. Да не бисмо стално понављали операцију степеновања користићемо променљиву која ће се у сваком кораку циклуса множити са 2 (практично то је степен броја 2, али се не рачуна сваки пут када нам треба већ се само претходна вредност увећава 2 пута и тако се добија на ефикасности програма). Подразумева се да ћемо бинарне цифре читати са десна у лево, односно, од цифре најнижег реда. Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0','1',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    b,s,i:integer;
begin a:=Edit1.Text;
      s:=1;b:=0;
      For i:=Length(a) downto 1 do
      begin b:=b+StrToInt(a[i])*s;
            s:=s*2;
      end;
      Edit2.Text:=IntToStr(b);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

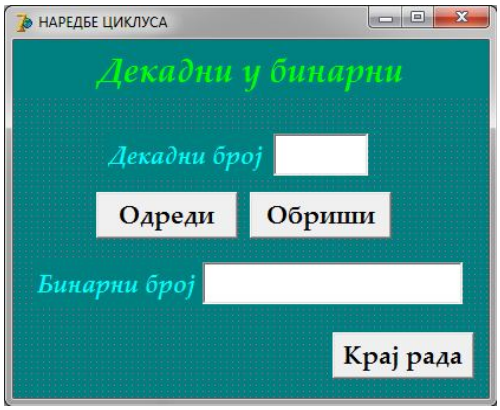
```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    b,s,i:integer;
begin a:=Edit1.Text;
      s:=1;b:=0;
      i:=Length(a);
      If i>0 then
       Repeat b:=b+StrToInt(a[i])*s;
              s:=s*2;
              i:=i-1;
       until i<1;
      Edit2.Text:=IntToStr(b);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    b,s,i:integer;
begin a:=Edit1.Text;
      s:=1;b:=0;
      i:=Length(a);
      While i>0 do
      begin b:=b+StrToInt(a[i])*s;
            s:=s*2;
            i:=i-1;
      end;
      Edit2.Text:=IntToStr(b);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који декадни број претвара у бинарни.***

Задатак ћемо решити тако што ћемо декадни број делити са 2 и бележити остатке све док количник у неком кораку не постане 0. Када се добијени низ остатака прочита с десна у лево добија се бинарни број - решење. Јасно је да се овај задатак не може решити уз помоћ циклуса *for* јер не можемо знати колико пута ће се делити. Најпре ћемо креирати форму, а затим написати и комплетан програм:



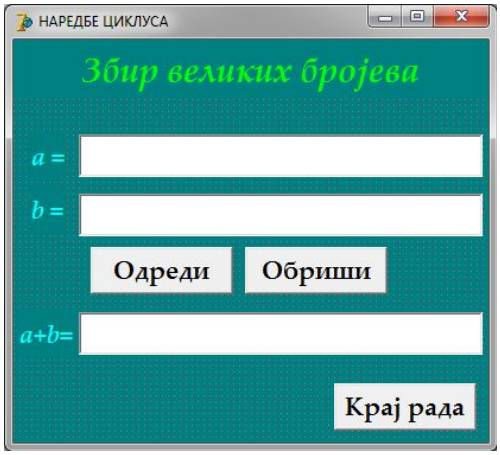

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    b:integer;
begin b:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(b);
      a:='';
      If b>0 then
      Repeat a:=IntToStr(b mod 2)+a;
             b:=b div 2;
      until b<1;
      Edit2.Text:=a;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    b:integer;
begin b:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(b);
      a:='';
      While b>0 do
      begin a:=IntToStr(b mod 2)+a;
            b:=b div 2;
      end;
      Edit2.Text:=a;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који одређује збир два цела броја са двадесет и више цифара.***

Из услова задаткавиди се да ћемо сабирке унети као стрингове и затим карактер по карактер с десна у лево претварати у бројеве, сабирати и формирати цифре резултата водећи при томе рачуна о евентуалном преносу (када је збир две цифре већи од 9). Цифре резултата ћемо, такође, претварати у стринг и тако памтити резултат. Да бисмо добили најједноставније решење најпре ћемо одредити која реч је краћа (ако нису исте дужине), а затим ћемо ту реч допунити нулама са леве стране (да се збир не би променио) тако да се добију две речи једнаке дужине. Задатак се може решити применом било које наредбе циклуса, а најефикаснији је са *for*. Најпре ћемо креирати форму, а затим написати и комплетан програм:




```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit1.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c:string;
    i,d,p:integer;
begin a:=Edit1.Text;
      b:=Edit2.Text;
      If Length(a)>Length(b)
       then begin c:=a;a:=b;b:=c;
            end;
      For i:=1 to Length(b)-Length(a) do a:='0'+a;
      c:='';p:=0;
      For i:=Length(a) downto 1 do
      begin d:=StrToInt(a[i])+StrToInt(b[i])+p;
            c:=IntToStr(d mod 10)+c;
            p:=d div 10;
      end;
      If p>0 then c:=IntToStr(p)+c;
      Edit3.Text:=c;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c:string;
    i,d,p:integer;
begin a:=Edit1.Text;
      b:=Edit2.Text;
      If Length(a)>Length(b)
       then Repeat b:='0'+b;
            until Length(b)=Length(a);
       else If Length(b)>Length(a)
             then Repeat a:='0'+a;
                  until Length(b)=Length(a);
      c:='';p:=0;
      i:=Length(a);
      Repeat d:=StrToInt(a[i])+StrToInt(b[i])+p;
             c:=IntToStr(d mod 10)+c;
             p:=d div 10;
             i:=i-1;
      until i=0;
      If p>0 then c:=IntToStr(p)+c;
      Edit3.Text:=c;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

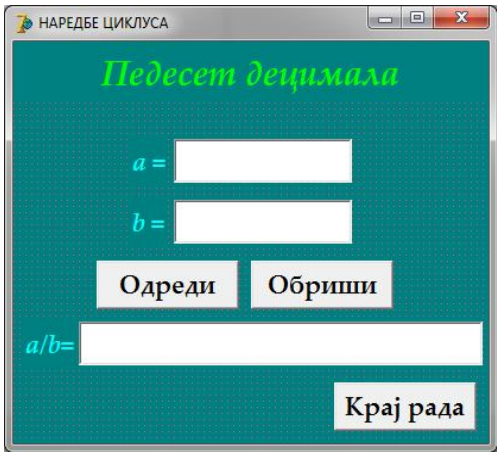
или:

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c:string;
    i,d,p:integer;
begin a:=Edit1.Text;
      b:=Edit2.Text;
      While Length(a)>Length(b) do b:='0'+b;
      While Length(b)>Length(a) do a:='0'+a;
      c:='';p:=0;
      i:=Length(a);
      While i>0 do
      begin d:=StrToInt(a[i])+StrToInt(b[i])+p;
            c:=IntToStr(d mod 10)+c;
            p:=d div 10;
            i:=i-1;
```

```
      end;
      If p>0 then c:=IntToStr(p)+c;
      Edit3.Text:=c;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Занимљиво би било одредити разлику и производ оваквих бројева.

- ***Саставити програм који одређује количник два броја на најмање педесет децимала.***



Да бисмо могли да испишемо број са педесет децимала морамо резултат дељења да конвертујемо у стринг. Почетна вредност за количник биће целобројни количник унетих бројева конвертован у стринг, а затим ћемо у тај стринг додати децимални зарез и целобројне количнике остатка при дељењу помноженог са десет и другог броја. Поступак понављамо све док не добијемо одговарајући број децимала. Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit1.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var c:string;
    a,b,i,n:integer;
begin a:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(b);
      n:=50;
      c:=IntToStr(a div b)+'.';
      For i:=1 to n do
      begin a:=a mod b*10;
            c:=c+IntToStr(a div b);
      end;
      Edit3.Text:=c;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var c:string;
    a,b,n:integer;
begin a:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(b);
      n:=50;
      c:=IntToStr(a div b)+'.';
      Repeat a:=a mod b*10;
             c:=c+IntToStr(a div b);
             n:=n-1;
      until n=0;
      Edit3.Text:=c;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

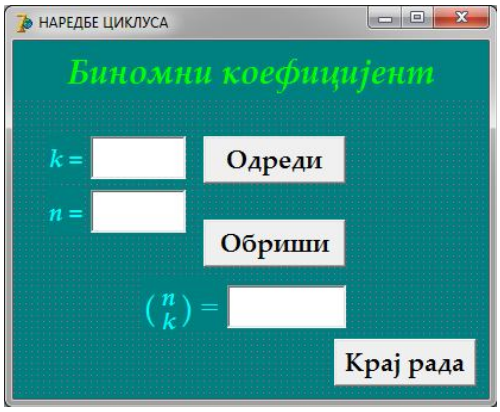
или:

```
procedure TForm1.Button1Click(Sender: TObject);
var c:string;
    a,b,n:integer;
begin a:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(b);
      n:=50;
```

```
      c:=IntToStr(a div b)+'.';
      While n>0 do
      begin a:=a mod b*10;
            c:=c+IntToStr(a div b);
            n:=n-1;
      end;
      Edit3.Text:=c;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

У задатку је дефинисано да количник има 50 децимала (***n:=50;***), променом вредности промењиве ***n*** лако се може дефинисати жељени број децимала. Програмерски боље решење је додати нови едит у који ће се уписивати вредност промењиве ***n***, односно, колико децимала желимо. Такође, да би задатак био коректно решен, требало би проверити да други уписани број није 0 (јер делити нулом нема смисла).

- ***Саставити програм који одређује к-ти биномни коефицијент степена n.***



У решењу задатка користићемо формулу у развијеном облику. Да бисмо добили мало уштеде на времену и меморији, количник на десној страни једнакости разбићемо на низ чинилаца који чини један елемент изнад разломачке црте подељен једним елементом испод разломачке црте (и то редом који су написани), односно, применићемо формулу:

$$\binom{n}{k}=\frac{n\cdot(n-1)\cdot(n-2)\cdot\ldots\cdot(n-k+1)}{k\cdot(k-1)\cdot(k-2)\cdot\ldots\cdot3\cdot2\cdot1}=\prod_{i=1}^{k}\frac{n-i+1}{i}.$$

Најпре ћемо креирати форму (формулу *ен над ка* смо исписали у три лабеле), а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,k,i:integer;
    b:real;
begin n:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(n);
      k:=StrToIntDef(Edit2.Text,0);
      If k>n then k:=n;
      Edit2.Text:=IntToStr(k);
      b:=1;
      For i:=1 to k do b:=b*(n-i+1)/i;
      Edit3.Text:=IntToStr(Trunc(b));
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

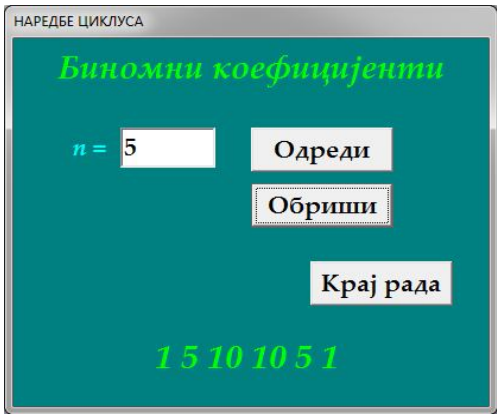
```
procedure TForm1.Button1Click(Sender: TObject);
var n,k,i:integer;
    b:real;
begin n:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(n);
      k:=StrToIntDef(Edit2.Text,0);
      If k>n
       then k:=n;
      Edit2.Text:=IntToStr(k);
      b:=1;i:=0;
      If k>0 then
       Repeat i:=i+1;
```

```
              b:=b*(n-i+1)/i;
       until i>=k;
      Memo1.Lines.Add(IntToStr(Trunc(b)));
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,k,i:integer;
    b:real;
begin n:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(n);
      k:=StrToIntDef(Edit2.Text,0);
      If k>n
       then k:=n;
      Edit2.Text:=IntToStr(k);
      b:=1;i:=0;
      While i<k do
      begin i:=i+1;
            b:=b*(n-i+1)/i;
      end;
      Memo1.Lines.Add(IntToStr(Trunc(b)));
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који исписује биномне коефицијенте степена n.***



Сваки појединачни биномни коефицијент ћемо одређивати као у претходном задатку, при чему ћемо променљиву *k* мењати у циклусу од нуле до унетог степена. Исписивање коефицијената ћемо решити помоћу стринг променљиве која ће се мењати у циклусу по *k* слепљивањем управо одређеног коефицијента на претходне. На крају ћемо стринг исписати у лабели. Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Label3.Caption:='';
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,k,i:integer;
    b:real;
    a:string;
begin n:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(n);
      a:='';
      For k:=0 to n do
      begin b:=1;
            For i:=1 to k do b:=b*(n-i+1)/i;
            a:=a+IntToStr(Trunc(b))+' ';
      end;
      Label3.Caption:=a;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,k,i:integer;
    b:real;
    a:string;
begin n:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(n);
      a:='';k:=0;
      Repeat b:=1;i:=0;
             If k>0 then
              Repeat i:=i+1;
                     b:=b*(n-i+1)/i;
              until i>=k;
             a:=a+IntToStr(Trunc(b))+' ';
             k:=k+1;
      until k>n;
      Label3.Caption:=a;
```

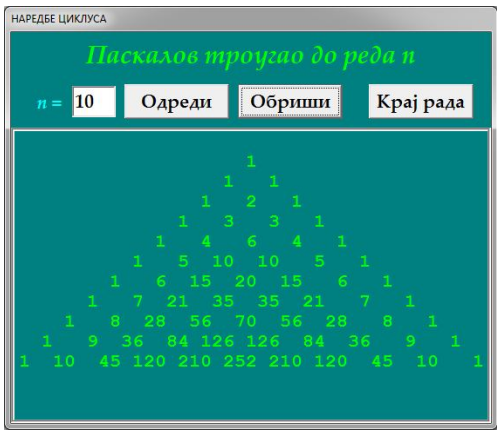
```
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,k,i:integer;
    b:real;
    a:string;
begin n:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(n);
      a:='';k:=0;
      While k<=n do
      begin b:=1;i:=0;
            While i<k do
            begin i:=i+1;
                  b:=b*(n-i+1)/i;
            end;
            a:=a+IntToStr(Trunc(b))+' ';
            k:=k+1;
      end;
      Label3.Caption:=a;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који исписује Паскалов троугао до реда n.***

Разлика између овог и претходног задатка је у томе што се у претходном унето ***n*** односи на један ред Паскаловог троугла који треба исписати (коефицијенти н-тог степена бинома), а овде се односи на број редова троугла које треба исписати, односно, треба одредити и исписати редове (коефицијенте бинома степенованог бројевима) од 0 до ***n***. Навешћемо без детаљног објашњавања решење помоћу свих наредби циклуса. Најпре ћемо креирати форму, а затим написати и комплетан програм:




```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
```

```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var n,p,k,i:integer;
    b:real;
    a:string;
begin p:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(p);
      For n:=0 to p do
      begin a:='1';
            For k:=1 to n do
            begin b:=1;
                  For i:=1 to k do b:=b*(n-i+1)/i;
                  a:=a+Format('%4.0f',[b]);
            end;
            Memo1.Lines.Add(a);
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

или:

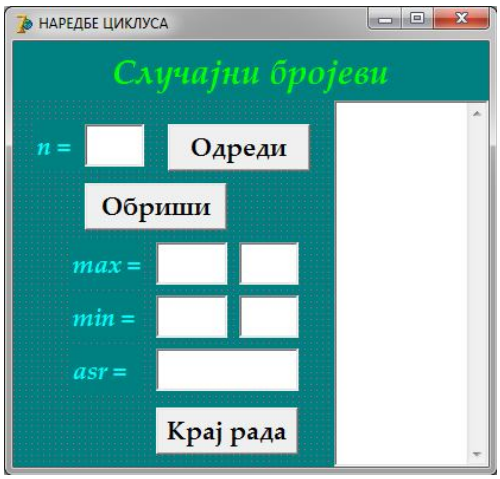
```
procedure TForm1.Button1Click(Sender: TObject);
var n,p,k,i:integer;
    b:real;
    a:string;
begin p:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(p);
      n:=0;
      Repeat a:='1';k:=1;
             Repeat b:=1;i:=0;
                    If k>0 then
                     Repeat i:=i+1;
                            b:=b*(n-i+1)/i;
                     until i>=k;
```

```
                    a:=a+Format('%4.0f',[b]);
                    k:=k+1;
              until k>n;
              Memo1.Lines.Add(a);
              n:=n+1;
       until n>p;
       Edit1.ReadOnly:=true;
       Button2.SetFocus;
end;
```

или:

```
procedure TForm1.Button1Click(Sender: TObject);
var n,p,k,i:integer;
    b:real;
    a:string;
begin p:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(p);
       n:=0;
       While n<=p do
       begin a:='1';k:=1;
             While k<=n do
             begin b:=1;i:=0;
                   While i<k do
                   begin i:=i+1;
                         b:=b*(n-i+1)/i;
                   end;
                   a:=a+Format('%4.0f',[b]);
                   k:=k+1;
             end;
             Memo1.Lines.Add(a);
             n:=n+1;
       end;
       Edit1.ReadOnly:=true;
       Button2.SetFocus;
end;
```

- ***Саставити програм који на случајан начин формира n бројева, а затим одређује највећи и најмањи број, њихове редне бројеве и аритметичку средину.***



Задатак демонстрира рад са низом бројева, када није потребно чувати вредности свих елемената низа. Генерисаћемо први број и прогласити га највећим и најмањим, а затим ћемо формирати остале бројеве и упоређивати са вредностима највећег и најмањег. Кад год наиђемо на већи, прогласићемо нови највећи. Кад год наиђемо на мањи, прогласићемо нови најмањи. Чуваћемо редне бројеве највећег и најмањег елемента. Без обзира на величину сабраћемо генерисани број са свим претходним. Аритметичку средину ћемо одредити дељењем збира са ***n***. Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
       Edit2.Clear;Edit3.Clear;Edit4.Clear;
       Edit5.Clear;Edit6.Clear;
       Memo1.Clear;
       Edit1.SetFocus;
end;
```

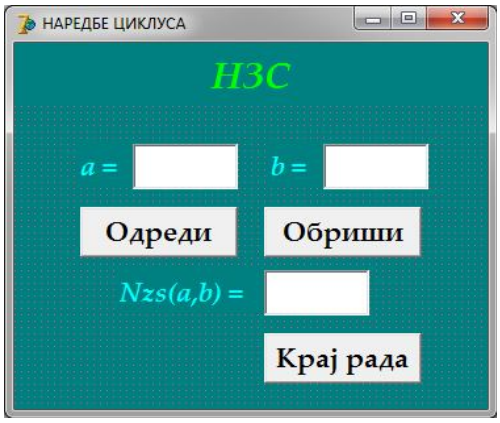
```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
        then key:=#27
        else If key=#13
              then Button1.Click;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var x,m,rx,rm,s,b,n,i:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
       Randomize;
       If n>0 then
       begin b:=Random(1000);
             Memo1.Lines.Add(Format('%2d.%4d',[1,b]));
             s:=b;
             x:=b;rx:=1;
             m:=b;rm:=1;
             For i:=2 to n do
             begin b:=Random(1000);s:=s+b;
                   If b>x
                     then begin x:=b;rx:=i;
                          end;
```

```
                If b<m
                  then begin m:=b;rm:=i;
                       end;
                Memo1.Lines.Add(Format('%2d.%4d',[i,b]));
            end;
            Edit2.Text:=IntToStr(x);Edit3.Text:=IntToStr(rx);
            Edit4.Text:=IntToStr(m);Edit5.Text:=IntToStr(rm);
            Edit6.Text:=FloatToStr(s/n);
      end;
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Циклус се користи само за бројање формираних елемената зато задатак нећемо решавати циклусима са условом.

- ***Саставити програм који за два уписана броја одређује најмањи заједнички садржалац.***



Постоји неколико алгоритама за одређивање најмањег заједничког садржалца. Овде су приказана три:

- за нзс - бројач узимамо први број, а затим у циклусу проверавамо да ли је бројач дељив са оба броја, ако јесте то је нзс, ако није повећавамо бројач за један .

- за нзс узимамо један од уписаних бројева и проверавамо да ли је други број његов делилац, ако јесте то је нзс, ако није потенцијални нзс увећавамо за први број све док не нађемо прави.

- за нзс узимамо производ уписаних бројева, а затим тај производ смањујемо за један све до једног од уписаних бројева и проверавамо да ли је дељив са оба уписана броја, па ако јесте то је нови нзс који исписујемо. Последњи исписани нзс је заиста то.

Други алгоритам је најефикаснији и њега треба запамтити.

Ако знамо нзд, онда нзс добијамо из формуле: ***nzs * nzd = a * b***.

Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);       // први алгоритам
var a,b,nzs,p:integer;
begin a:=StrToIntDef(Edit1.Text,3);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,4);Edit2.Text:=IntToStr(b);
      If a>b
       then begin p:=a;a:=b;b:=p;
            end;
      nzs:=b;
      While (nzs mod a<>0)or(nzs mod b<>0) do nzs:=nzs+1;
      Edit3.Text:=IntToStr(nzs);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);       // други алгоритам
var a,b,nzs,p:integer;
begin a:=StrToIntDef(Edit1.Text,3);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,4);Edit2.Text:=IntToStr(b);
      If a>b
       then begin p:=a;a:=b;b:=p;
            end;
      nzs:=b;             // ако пођемо од већег броја добијамо на ефикасности алгоритма
      While nzs mod a<>0 do
```

```
            nzs:=nzs+b;
      Edit3.Text:=IntToStr(nzs);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);       // трећи алгоритам
var a,b,nzs,p:integer;
begin a:=StrToIntDef(Edit1.Text,3);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,4);Edit2.Text:=IntToStr(b);
      p:=a*b+1;
      Repeat p:=p-1;
             If (p mod a=0)and(p mod b=0) then nzs:=p;
      until nzs=b;
      Edit3.Text:=IntToStr(nzs);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који за два уписана броја одређује највећи заједнички делилац.***

Постоји неколико алгоритама за одређивање највећег заједничког делиоца. Овде су приказана три:

- за нзд узимамо број 1, а затим у циклусу повећавамо бројач до једног од два унета броја и проверавамо да ли је делилац оба броја, па ако јесте постављамо њега за нови нзд. Последњи од таквих бројева је тражени нзд.

- за нзд узимамо један од уписаних бројева и проверавамо да ли је делилац другог броја, па ако јесте то је нзд, а ако није потенцијални нзд смањујемо га за један све док не нађемо број који је делилац оба броја.

- за нзд узимамо мањи од уписаних бројева и проверавамо да ли је делилац већег, ако није од већег одузимамо мањи и мањи проглашавамо потенцијалим нзд и поступак понављамо све док не добијемо делилац већег броја који представља нзд.

Трећи алгоритам је најефикаснији, па га треба запамтити.

Ако знамо нзс, онда нзд добијамо из формуле: ***nzs * nzd = a * b***.

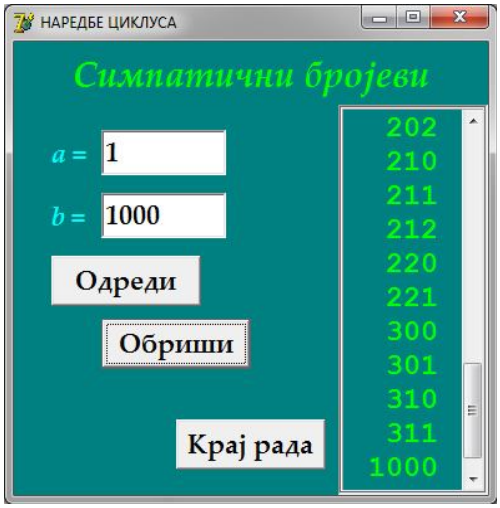
Најпре ћемо креирати форму, а затим написати и комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);       // први алгоритам
var a,b,nzd,p:integer; // i:integer;
begin a:=StrToIntDef(Edit1.Text,3);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,4);Edit2.Text:=IntToStr(b);
      If a>b
       then begin p:=a;a:=b;b:=p;
            end;
      p:=0;
      Repeat p:=p+1;
             If (a mod p=0)and(b mod p=0) then nzd:=p;
      until p=b;
      // решење са for је нешто мало једноставније, али су оба решења прилично неефикасна
      // For i:=1 to a do
      //  If (a mod i=0)and(b mod i=0)
      //   then nzd:=i;
      Edit3.Text:=IntToStr(nzd);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);       // други алгоритам
var a,b,nzd:integer;
begin a:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(b);
      nzd:=a;
      If b mod nzd<>0
       then Repeat nzd:=nzd-1;
            until (a mod nzd=0)and(b mod nzd=0);
      Edit3.Text:=IntToStr(nzd);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);       // трећи алгоритам
var a,b,p:integer;
begin a:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(b);
      If a<>b then
       Repeat If a>b
                then a:=a-b
                else b:=b-a;
       until a=b;
      Edit3.Text:=IntToStr(a);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

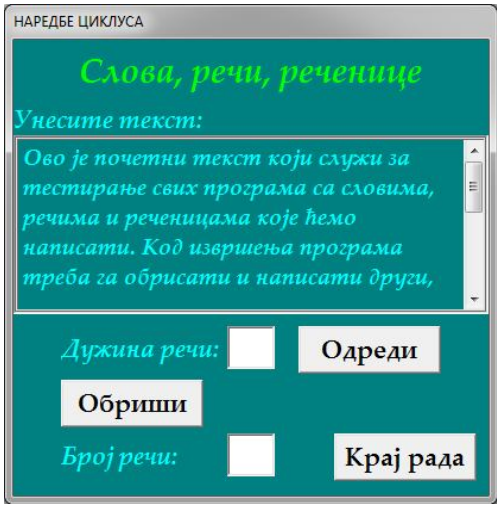
- ***Саставити програм који исписује све симпатичне бројеве од a до b.***

За природан број ***n*** кажемо да је симпатичан ако важи:
$S(n^2) = S(n)*S(n)$, где је ***S(x)*** збир цифара природног броја ***x***.
Задатак ћемо решити комбинацијом циклуса ***for*** и ***repeat***, а на сличан начин би се могло решити и неком другом комбинацијом наредби циклуса. Одредићемо збир цифара неког броја, а затим и збир цифара квадрата тог броја и проверити услов задатка, тј. да ли је збир цифара квадрата броја једнак квадрату збира цифара тог броја. Најпре ћемо креирати форму, а затим написати и комплетан програм:




```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Memo1.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,i,p,z1,z2:integer;
begin a:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(b);
      Memo1.Clear;
      For i:=a to b do
      begin z1:=0; p:=i;
            Repeat c:=p mod 10;z1:=z1+c;
                   p:=p div 10;
            until p=0;
            z2:=0; p:=i*i;
            Repeat c:=p mod 10;z2:=z2+c;
                   p:=p div 10;
            until p=0;
            If z2=z1*z1
             then Memo1.Lines.Add(Format('%4d',[i]));
      end;
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

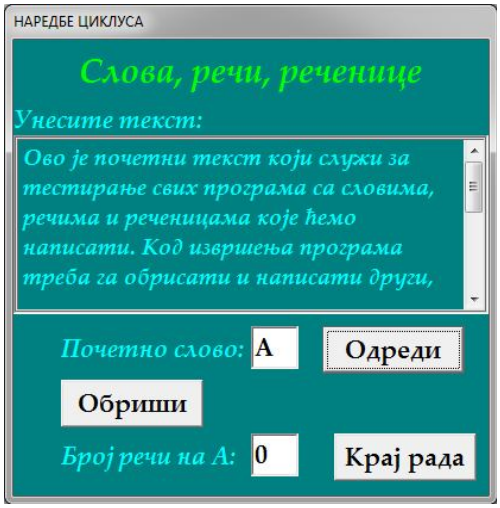
- ***Саставити програм који за текст написан у MetoBox-у одређује број речи дужине 5 карактера.***



Граничник речи може бити размак, зарез, тачка, знак узвика или знак питања, али и ознака краја реда (#13) и ознака прекида реда (#10) и све ове карактер (ова последња два нису видљива) треба пронаћи у тексту, а граница речи је први од њих. Функцијом *Pos()* ћемо одредити позицију сваког од њих и пронаћи најмањи број који представља крај речи. Овај поступак ћемо користити у свим задацима где се траже речи са неким карактеристикама, а мало модификован ће бити користан и за одређивање реченице или неког специјалног низа карактера. Променљивој ***r*** ћемо доделити ту реч и проверити да ли она има одговарајућу дужину. У задатку се траже речи са 5 карактера, али решење које приказујемо је општије и корисник може да упише дужину коју жели, тако да је алгоритамски коректније. Бројач ***d*** се увећава када је пронађена реч одговарајуће дужине и исписује се на крају програма. Услов за крај претраживања је да је остатак текста дужине 0 карактера и да има размака у тексту. Овај други услов обезбеђује рад програма и у случају када се текст састоји из само једне речи, па нема ниједног од карактера који ограничавају реч. Процедуром *Edit1KeyPress()* смо заштитили први едит од уноса слова и знакова, а процедуром *Edit1Enter()* смо дефинисали да се садржај оба едита брише када се курсор постави у први. Зато процедуру која се извршава када се кликне тастер *Обриши* не треба користити ако желимо само да променимо дужину речи да бисмо проверили колико има речи те друге дужине. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;
      Edit2.Clear;
      Memo1.SetFocus;
end;
procedure TForm1.Edit1Enter(Sender: TObject);
begin Edit1.Clear;
      Edit2.Clear;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    a,b,d,n:integer;
begin n:=StrToIntDef(Edit1.Text,5);Edit1.Text:=IntToStr(n);
      s:=Memo1.Text+' ';
      d:=0;
      While (Length(s)>0)and(Pos(' ',s)<>0) do
      begin a:=Pos(' ',s);
            b:=Pos(#13,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(#10,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(',',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('.',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('!',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('?',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            r:=Copy(s,1,a-1); // формира реч
            Delete(s,1,a);    // брише реч из текста
            If a=n+1          // проверава да ли је реч жељене дужине
             then d:=d+1;     // броји речи жељене дужине
      end;
      If Length(s)=n+1        // проверава да ли је остатак текста реч жељене дужине
       then d:=d+1;           // ако јесте убраја и њу
      Edit2.Text:=IntToStr(d);// исписује резултат
      Memo1.ReadOnly:=true;
end;
```

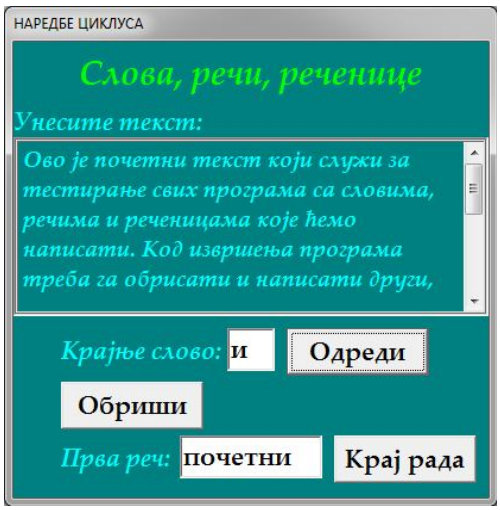
- ***Саставити програм који за текст написан у MemoBox-у одређује број речи које почињу великим словом A.***



Одређиваћемо речи као и у претходном задатку, а затим ћемо проверити да ли је прво слово изабрани карактер. У задатку се тражи број речи које почињу великим словом A, а приказано решење је општије и дозвољава кориснику да упише било који карактер и провери да ли и колико има речи које почињу тим карактером. Решење је *Case Sensitive*, односно разликује мала и велика слова. Лако се може укинути ово ограничење заменом услова ***r[1]=c*** условом ***UpCase(r[1])=UpCase(c)***. Услов за престанак претраживања је да се задато слово више не налази у тексту. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;Edit2.Clear;
      Memo1.SetFocus;
end;
procedure TForm1.Edit1Enter(Sender: TObject);
begin Edit1.Clear;Edit2.Clear;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If key=#13 then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    c:char;
    a,b,d:integer;
begin c:=Edit1.Text[1];Label4.Caption:='Број речи на '+c+':';
      s:=Memo1.Text+' ';d:=0;
      While Pos(c,s)<>0 do
      begin a:=Pos(' ',s);
            b:=Pos(#13,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(#10,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(',',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('.',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('!',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('?',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            r:=Copy(s,1,a-1);Delete(s,1,a);
            If (Length(r)>0)and(r[1]=c) then d:=d+1;
      end;
      Edit2.Text:=IntToStr(d);
end;
```

- ***Саставити програм који за текст написан у MemoBox-у одређује прву реч која завршава малим словом a.***

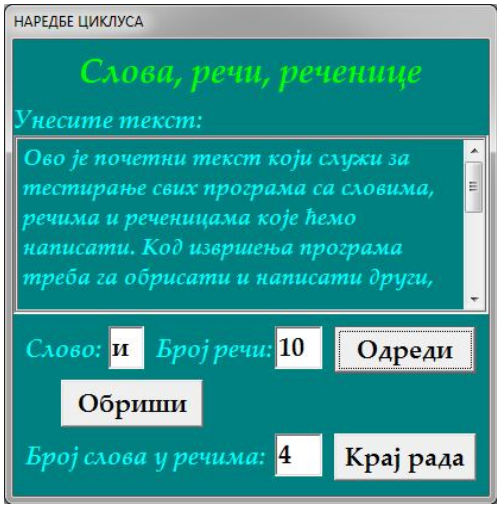


Разлика између овог и претходног задатка је у томе што се овде тражи само прва реч која завршава задатим карактером. То значи да треба увести променљиву показивач која ће променити вредност оног тренутка када нађемо реч која задовољава услов задатка. Ова променљива мора да учествује у услову циклуса који претражује речи јер претраживање мора да се прекине чим се реч пронађе. Други услов је да је комплетан текст претражен, тј. да је остатак текста дужине 0 карактера. Услови да нешто буде реч су они исти као у и претходним задацима. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;Edit2.Clear;
      Memo1.SetFocus;
end;
```

```
procedure TForm1.Edit1Enter(Sender: TObject);
begin Edit1.Clear;Edit2.Clear;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If key=#13 then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    c:char;
    a,b,d:integer;
begin c:=Edit1.Text[1];
      s:=Memo1.Text+' ';d:=0;
      While (Pos(c,s)<>0)and(d=0) do
      begin a:=Pos(' ',s); b:=Pos(#13,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(#10,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(',',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('.',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('!',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('?',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            r:=Copy(s,1,a-1);Delete(s,1,a);
            If (Length(r)>0)and(r[Length(r)]=c)and(d=0) then d:=1;
      end;
      If d=0
       then Edit2.Text:='Nema reci.'
       else Edit2.Text:=r;
end;
```

- ***Саставити програм који за текст написан у MemoBox-у одређује број појављивања симбола * у првих пет речи.***

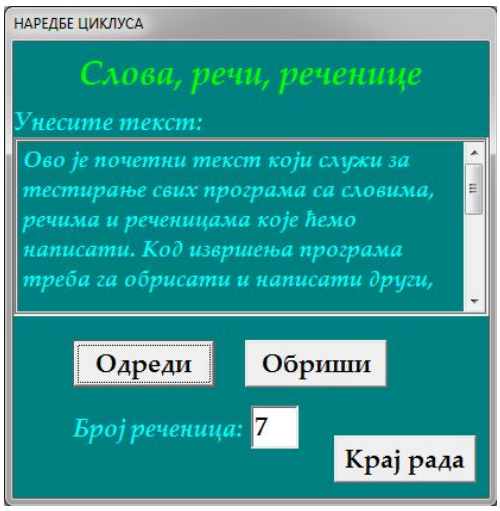


Треба одредити првих пет речи и видети да ли и колико пута се у њима појављује симбол *. Решење које се приказује је општије и кориснику дозвољава да упише карактер по избору и број речи у којима се тражи уписани карактер. Променљива која броји речи мора да учествује у услову циклуса за претраживање да би се он прекинуо када се изврши тестирање задатог броја речи. Други услов је да је комплетан текст претражен, тј. да је остатак текста дужине 0 карактера. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;Edit2.Clear;Edit3.Clear;
      Memo1.SetFocus;
end;
procedure TForm1.Edit1Enter(Sender: TObject);
begin Edit1.Clear;Edit2.Clear;Edit3.Clear;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If key=#13 then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    c:char;
    a,b,d,n,i:integer;
begin c:=Edit1.Text[1];
      d:=StrToIntDef(Edit2.Text,5);Edit2.Text:=IntToStr(d);
      s:=Memo1.Text+' ';n:=0;
      While (Length(s)>0)and(d>0) do
      begin a:=Pos(' ',s);
            b:=Pos(#13,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(#10,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
```

```
            b:=Pos(',',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('.',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('!',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('?',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            r:=Copy(s,1,a-1);Delete(s,1,a);d:=d-1;
            If (Length(r)>0)and(Pos(c,r)>0)
             then For i:=1 to Length(r) do
                      If r[i]=c then n:=n+1;
      end;
      Edit3.Text:=IntToStr(n);
end;
```

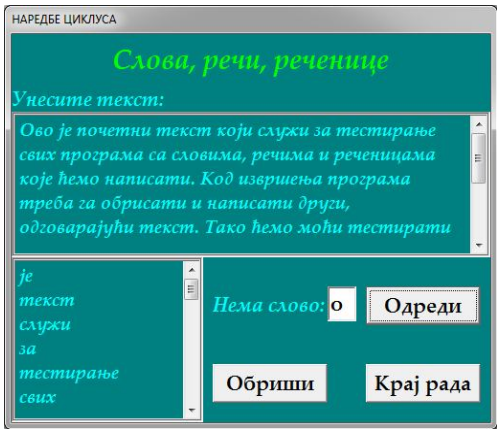
- ***Саставити програм који за текст написан у MemoBox-у одређује број реченица у тексту (крај реченице се означава симболима . ! ?).***



Задатак је нешто једноставнији од претходних јер је скуп карактера који дефинишу крај реченице, а чије постојање треба утврдити, мањи. Једино о чему треба водити рачуна је да не прогласимо реченицом и вишеструко узастопно понављање знакова интерпункције. Променљива ***a*** само на почетку има вредност дужине текста, а надаље, кроз циклус, њена вредност је позиција краја реченице. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;
      Memo1.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    a,b,d:integer;
begin s:=Memo1.Text+' ';
      a:=Length(s);d:=0;
      While (Length(s)>0)and(a<>0) do
      begin a:=Pos('.',s);
            b:=Pos('?',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('!',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            r:=Copy(s,1,a);Delete(s,1,a);
            If Length(r)>0 then d:=d+1;          // не броје се дуплирани граничници
      end;
      Edit1.Text:=IntToStr(d);
end;
```

- ***Саставити програм који за текст написан у MemoBox-у у други MemoBox исписује све речи које не садрже симбол *.***



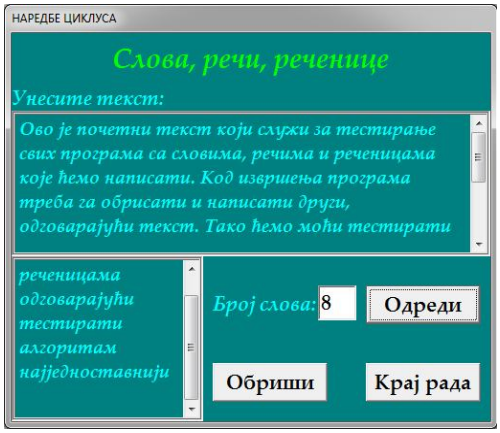
Као и у неким ранијим задацима, одређиваћемо речи, затим проверавати да ли се задати карактер налази у њима. Решење које је приказано је општије од траженог условима задатка јер дозвољава кориснику да изабере карактер који ће се тражити у речима. Мемо поље у које исписујемо пронађене речи се брише чим се промени карактер који се тражи. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;Memo2.Clear;
      Memo1.SetFocus;
end;
procedure TForm1.Edit1Enter(Sender: TObject);
begin Edit1.Clear;Memo2.Clear;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If key=#13 then Button1.Click;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    c:char;
    a,b:integer;
begin s:=Memo1.Text+' ';
      c:=Edit1.Text[1];
      While Length(s)>0 do
      begin a:=Pos(' ',s);
            b:=Pos(#13,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(#10,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(',',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('.',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('!',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('?',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            r:=Copy(s,1,a-1);
            Delete(s,1,a);
            If (Length(r)>0)and(Pos(c,r)=0)
             then Memo2.Lines.Add(r);
      end;
end;
```

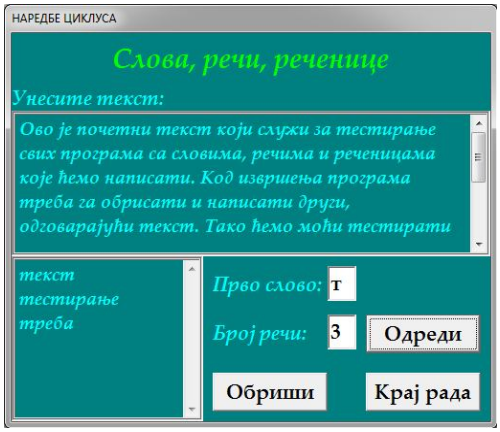
- ***Саставити програм који за текст написан у MemoBox-у у други MemoBox исписује све речи које имају дужину више од три слова.***

Као у неким ранијим задацима одређиваћемо речи, а онда ћемо проверавати да ли су задате дужине. Решење које је приказано општије је од траженог у условима задатка јер дозвољава кориснику да изабере дужину речи које ће се тражити. Мемо поље у које исписујемо пронађене речи се празни чим се промени дужина речи које се траже. Након креирања форме написаћемо комплетан програм:




```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;Memo2.Clear;
      Memo1.SetFocus;
end;

procedure TForm1.Edit1Enter(Sender: TObject);
begin Edit1.Clear;Memo2.Clear;
end;

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If key=#13
       then Button1.Click;
end;

procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    a,b,n:integer;
begin s:=Memo1.Text+' ';
      n:=StrToIntDef(Edit1.Text,3);Edit1.Text:=IntToStr(n);
      While Length(s)>0 do
      begin a:=Pos(' ',s);
            b:=Pos(#13,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(#10,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(',',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('.',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('!',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('?',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            r:=Copy(s,1,a-1);
            Delete(s,1,a);
            If Length(r)>n
             then Memo2.Lines.Add(r);
      end;
end;
```

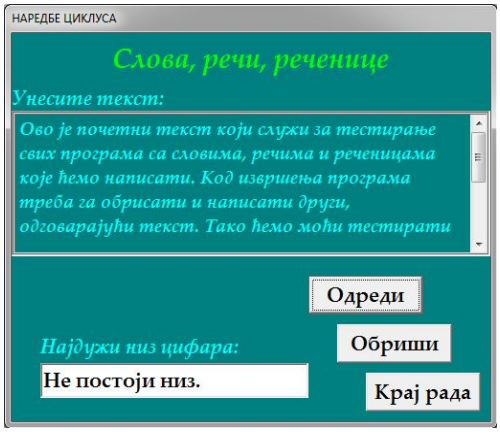
- ***Саставити програм који за текст написан у MemoBox-у у други MemoBox исписује првих пет речи које почињу великим словом А.***



Већ смо показали како се у тексту траже речи које почињу задатим карактером, сада ћемо то решење проширити додатним условима. Уместо да бројимо речи, исписујемо их. У приказаном решењу бира се слово којима речи треба да почињу и, такође, колико таквих речи се тражи. Подразумева се да може да се деси и да нема таквих речи. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;Edit2.Clear;Memo2.Clear;
      Memo1.SetFocus;
end;
procedure TForm1.Edit1Enter(Sender: TObject);
begin Edit1.Clear;Edit2.Clear;Memo2.Clear;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If key=#13 then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    c:char;
    a,b,n:integer;
begin s:=Memo1.Text+' ';
      c:=Edit1.Text[1];
      n:=StrToIntDef(Edit2.Text,5);Edit2.Text:=IntToStr(n);
      While (Length(s)>0)and(n>0) do
      begin a:=Pos(' ',s);
            b:=Pos(#13,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(#10,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(',',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('.',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('!',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('?',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            r:=Copy(s,1,a-1);
            Delete(s,1,a);
            If (Length(r)>0)and(r[1]=c)
             then begin n:=n-1;Memo2.Lines.Add(r);
                  end;
      end;
end;
```

- ***Саставити програм који у уписаном стрингу одређује најдужи подниз који се састоји само из цифара.***

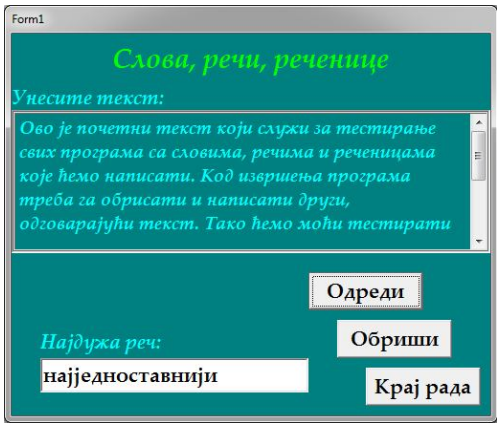


Овде нам нису потребне речи. Потребно је наћи низ узастопних цифара и одредити његову дужину. Може се десити да у тексту има више низова цифара и у том случају треба одредити који је најдужи, или да цифара нема у тексту који анализирамо, у том случају исписујемо одговарајућу поруку. Низ цифара може да почне било којом цифром, па је услов почетка низа је компликованији. Зато смо услов за одређивање почетка низа цифара представили бројачким циклусом. Да бисмо утврдили да у тексту нема цифара увели смо бројач ***n*** који је на почетку бројачког циклуса 0, тј. претпостављамо да цифре постоје у тексту, а затим, кад год утврдимо да нека цифра не постоји, бројач увећавамо за један. Ако је ***n=10*** значи да нисмо пронашли ниједну цифру у тексту и треба испразнити текст да бисмо изашли из претраживања. У

противном одређујемо дужину низа и проверавамо да ли је он најдужи. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;
      Memo1.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    c:char;
    a,b,n,m:integer;
begin s:=Memo1.Text+' ';
      m:=0;r:='';
      While Length(s)>0 do
      begin a:=Length(s);
            n:=0;
            For c:='0' to '9' do
            begin b:=Pos(c,s);
                  If (b>0)and(b<a) then a:=b;
                  If b=0 then n:=n+1;
            end;
            If n=10
             then Delete(s,1,a)
             else begin Delete(s,1,a-1);
                        n:=1;
                        While s[n] in ['0'..'9'] do n:=n+1;
                        If n-1>m
                         then begin m:=n-1;
                                    r:=Copy(s,1,n-1);
                              end;
                        Delete(s,1,n-1);
                  end;
      end;
      If m>0
       then Edit1.Text:=r
       else Edit1.Text:='Не постоји низ.';
end;
```

- ***Саставити програм који у уписаном стрингу одређује најдужу реч (низ карактера ограничен размацима).***



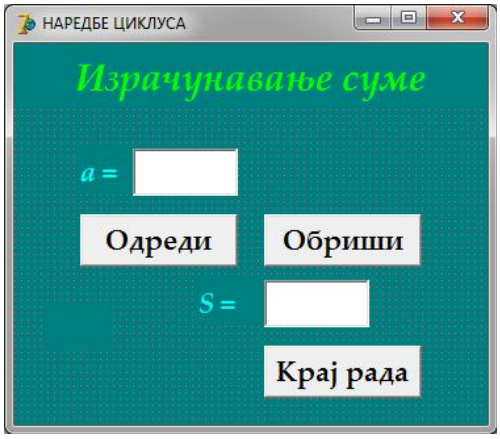
Задатак је само наизглед сличан претходном. Међутим, овде прво треба одредити реч (као у неким претходним решењима), а затим и њену дужину, па онда треба проверити и да ли је она најдужа реч. Такође, треба предвидети и могућност да текст буде празан, тј. да уопште нема речи. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Memo1.Clear;Memo1.ReadOnly:=false;
      Edit1.Clear;
      Memo1.SetFocus;
end;
procedure TForm1.Button1Click(Sender: TObject);
var s,r:string;
    a,b,n:integer;
begin s:=Memo1.Text+' ';
      n:=0;r:='';
      While Length(s)>0 do
      begin a:=Pos(' ',s);
            b:=Pos(#13,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(#10,s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos(',',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('.',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('!',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
            b:=Pos('?',s);
            If (a>0)and(b>0)and(b<a) then a:=b;
```

```
            If a-1>n
             then begin n:=a-1;
                        r:=Copy(s,1,a-1);
                  end;
            Delete(s,1,a);
      end;
      If n>0
       then Edit1.Text:=r
       else Edit1.Text:='Не постоји реч.';
end;
```

- ***Саставити програм који израчунава вредност суме:***
  $s=0-1+2-3+...+(-1)^n n$, ***где је број n уписани природни број.***

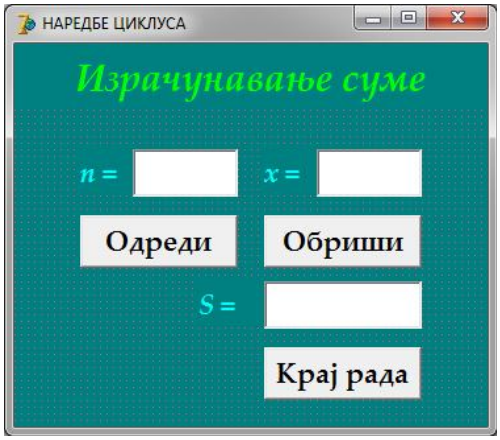
Задаци са одређивањем вредности неког израза су, углавном, једноставни. У овом случају, ради се о наизменичном сабирању и одузимању природних бројева до ***n***. Решићемо задатак као да се ради о сабирању, а промену знака дефинисаћемо помоћу промењиве ***z*** која ће се у сваком кораку циклуса множити са -1. Након креирања форме написаћемо комплетан програм:




```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var n,s,i,z:integer;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      s:=0; z:=1;
      For i:=1 to n do
      begin z:=-z;
            s:=s+z*i;
      end;
      Edit2.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који израчунава вредност суме:***
  $s=\sin(x)+\sin^2(x)+\sin^3(x)+...+\sin^n(x)$, ***где је број n уписани природни број.***

Број x може бити реалан. Вредност функције *sin(x)* израчунаћемо испред циклуса и сачувати у некој промењљивој, а у циклусу ћемо радити само са том промењљивом коју ћемо множити потребан број пута и сабирати. Након креирања форме написаћемо комплетан програм:



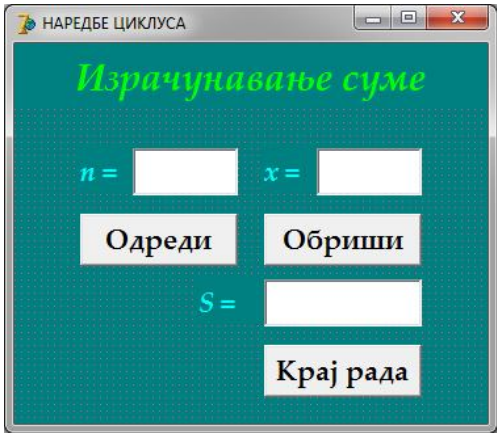

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',',','.',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
```

```
var n,i:integer;
    s,p,z,x:real;
begin n:=StrToIntDef(Edit1.Text,1);Edit1.Text:=IntToStr(n);
      x:=StrToFloatDef(Edit2.Text,1);
      Edit2.Text:=FloatToStr(x);
      s:=0; z:=1;
      p:=sin(x);
      For i:=1 to n do
      begin z:=z*p; s:=s+z;
      end;
      Edit3.Text:=FloatToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који израчунава вредност суме:***

$$S_n = \frac{0!}{x^0} + \frac{1!}{x^1} + \frac{2!}{x^2} + \cdots + \frac{n!}{x^n}, \text{ где је } n \in N \text{ и } n \le 16, x \in R \text{ и } 1 < x < n.$$

По дефиницији 0! је 1. За израчунавање факторијела и степена користићемо помоћне променљиве да бисмо задатак решили коришћењем само једног циклуса. Обавезна је контрола уноса броја n због ограничења целобројног типа података (факторијел много брзо расте и прелази границу за целе бројеве). Након креирања форме написаћемо комплетан програм:



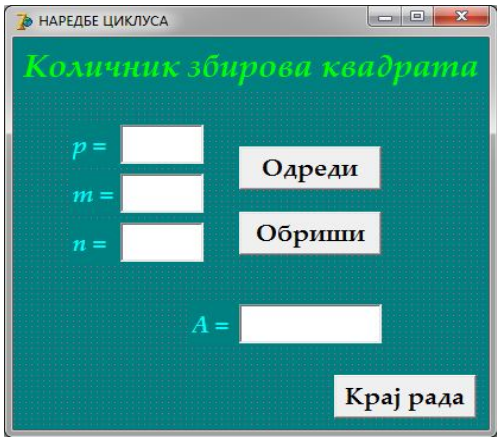

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;
      Edit1.SetFocus;
end;

procedure TForm1.Edit1Exit(Sender: TObject);
var p:integer;
begin p:=StrToIntDef(Edit1.Text,5);
      If p>16
       then begin Edit1.Clear;
                  Edit1.SetFocus;
            end;
end;

procedure TForm1.Edit2Exit(Sender: TObject);
var p:integer;
    r:real;
begin p:=StrToIntDef(Edit1.Text,5);
      r:=StrToFloatDef(Edit2.Text,1);
      If (r<1)or(r>p)
       then begin Edit2.Clear;
                  Edit2.SetFocus;
            end;
end;

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;

procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',',','.',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;

procedure TForm1.Button1Click(Sender: TObject);
var n,i,f:integer;
    s,p,x:real;
begin n:=StrToIntDef(Edit1.Text,5);Edit1.Text:=IntToStr(n);
      x:=StrToFloatDef(Edit2.Text,1);Edit2.Text:=FloatToStr(x);
      s:=1; f:=1; p:=1;
      For i:=1 to n do
      begin f:=f*i;      // одређује факторијел
            p:=p*x;      // одређује степен
            s:=s+f/p;    // израчунава суму
      end;
      Edit3.Text:=FloatToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Много је ефектније и ефикасније следеће решење (јер не рачуна ни факторијеле и ни степене; идеја је да се они напишу у развијеном облику, па се онда формирају производи количника):

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer;
    s,p,x:real;
begin n:=StrToIntDef(Edit1.Text,5);Edit1.Text:=IntToStr(n);
      x:=StrToFloatDef(Edit2.Text,1);Edit2.Text:=FloatToStr(x);
      s:=1; p:=1;
      For i:=1 to n do
      begin p:=p*i/x;
            s:=s+p;
      end;
      Edit3.Text:=FloatToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који израчунава вредност израза:***
  $a=\dfrac{m^2+(m+1)^2+(m+2)^2+\ldots+(n-2)^2+(n-1)^2+n^2}{1^2+2^2+3^2+\ldots+(p-2)^2+(p-1)^2+p^2}$***, где су m, n, природни бројеви.***

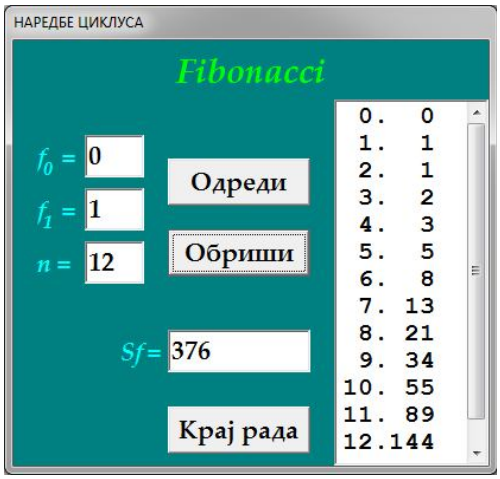
Јасно је да мора бити $m < n$ и $p > 0$. То се мора обезбедити када корисник уноси податке. Задатак је најједноставније решити са два независна циклуса (мада је могуће користити и циклусе са условом). У првом се рачуна вредност бројиоца, а у другом вредност имениоца. Затим се та два резултата поделе. Након креирања форме написаћемо комплетан програм:




```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit3.ReadOnly:=false;
      Edit4.Clear;
      Edit3.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit3.SetFocus;
end;
procedure TForm1.Edit3KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Edit1Exit(Sender: TObject);
var p:integer;
begin p:=StrToIntDef(Edit1.Text,0);
      If p<=0
       then begin Edit1.Clear;
                  Edit1.SetFocus;
            end;
end;
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,i,n,s:integer;
begin p:=StrToIntDef(Edit1.Text,5);Edit1.Text:=IntToStr(p);
      m:=StrToIntDef(Edit2.Text,2);
      n:=StrToIntDef(Edit3.Text,3);
      If m>n
       then begin a:=m;m:=n;n:=a;
            end;
      Edit2.Text:=IntToStr(m);
      Edit3.Text:=IntToStr(n);
      a:=0;
      For i:=m to n do a:=a+i*i;
      b:=0;
      For i:=1 to p do b:=b+i*i;
```

```
      Edit4.Text:=FloatToStr(a/b);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
      Button2.SetFocus;
end;
```

- ***Саставити програм који израчунава суму првих n елемената Фибоначијевог низа (Фибоначијев низ почиње са $f_0=0$ и $f_1=1$, а сваки следећи елемент се израчунава као збир претходна два).***



Фибоначијев низ је један од значајнијих феномена у математици који је нашао примену и у уметности и у физици, а може се пронаћи и у хемији, биологији, архитектури и готово у свим областима људског деловања. Ово је само један од примера употребе низа. Задатак је решен тако да корисник може променити почетне вредности за формирање низа. Након креирања форме написаћемо комплетан програм:

```
procedure TForm1.Button3Click(Sender: TObject);
begin Application.Terminate;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.Clear;Edit1.ReadOnly:=false;
      Edit2.Clear;Edit2.ReadOnly:=false;
      Edit3.Clear;Edit3.ReadOnly:=false;
      Edit4.Clear;Memo1.Clear;
      Edit1.Text:='0';Edit2.Text:='1';
      Edit3.SetFocus;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit2.SetFocus;
end;
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Edit3.SetFocus;
end;
procedure TForm1.Edit3KeyPress(Sender: TObject; var Key: Char);
begin If not (key in ['0'..'9',#8,#13,#128])
       then key:=#27
       else If key=#13
             then Button1.Click;
end;
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,i,n,s:integer;
begin a:=StrToIntDef(Edit1.Text,0);Edit1.Text:=IntToStr(a);
      b:=StrToIntDef(Edit2.Text,1);Edit2.Text:=IntToStr(b);
      n:=StrToIntDef(Edit3.Text,2);Edit3.Text:=IntToStr(n);
      Memo1.Lines.Add(Format('%2d.%3d',[0,a]));
      Memo1.Lines.Add(Format('%2d.%3d',[1,b]));
      s:=a+b;
      For i:=2 to n do
      begin c:=a+b;
            Memo1.Lines.Add(Format('%2d.%3d',[i,c]));
            a:=b;b:=c;
            s:=s+c;
      end;
      Edit4.Text:=IntToStr(s);
      Edit1.ReadOnly:=true;Edit2.ReadOnly:=true;Edit3.ReadOnly:=true;
      Button2.SetFocus;
end;
```

Задаци за самосталан рад

- Саставити програм који израчунава збир квадрата бројева између m и n, где су m и n цели позитивни бројеви.
- Саставити програм који израчунава збир кубова бројева између m и n, где су m и n цели позитивни бројеви.
- Саставити програм који исписује све троцифрене бројеве који су дељиви са 6 и нису дељиви ни са 4 ни са 9.
- Саставити програм који исписује све троцифрене бројеве који су дељиви производом, а нису дељиви збиром својих цифара.
- Саставити програм који исписује све просте бројеве мање од n са свим различитим цифрама.
- Саставити програм који одређује нзс и нзд за три уписана броја.
- Саставити програм који одређује збир простих бројева од a до b.
- Саставити програм који на случајан начин формира n целих бројева од -500 до 500, а затим одређује број негативних, број позитивних, број парних, број непарних.
- Саставити програм који на случајан начин одређује n троцифрених бројева, а затим одређује број непарних и њихов збир и аритметичку средину парних.
- Уписују се бројеви све док се не упише 0. Саставити програм који одређује број са највећим и најмањим збиром цифара, њихове редне бројеве и укупан број цифара свих уписаних бројева. У случају да више елемената има исти (највећи или најмањи) збир цифара исписује се најмањи од њих.
- Уписују се бројеви све док се не упише 0. Саставити програм који одређује најмањи парни, највећи непарни уписани број и њихове редне бројеве.
- Саставити програм који одређује да ли је уписана реч палиндромна.
- Саставити програм који исписује слова која се у речи појављују тачно једном.
- Саставити програм који исписује слова која се у речи појављују више пута.
- Саставити програм који одређује број самогласника у уписаној речи и исписује их.
- Саставити програм који одређује број сугласника у уписаној речи и исписује их.
- Саставити програм који одређује број самогласника који се у уписаној речи појављују више пута.
- Саставити програм који одређује број сугласника који се у уписаној речи појављују тачно два пута.
- Саставити програм који од унете речи формира словну пирамиду одузимајући по једно слово са леве и десне стране наизменично.
- Саставити програм који од унете речи формира словни пешчани сат одузимајући по једно слово са леве и десне стране наизменично, а затим враћајући слова истим редом.
- Саставити програм који од унете речи формира словно вретено одузимајући по једно слово са леве и десне стране наизменично од средине према врховима вретрена.
- Саставити програм који симулира бацање коцкице за јамб n пута, а затим исписује број појављивања сваке стране.
- Саставити програм који одређује разлику два цела броја са двадесет и више цифара.
- Саставити програм који арапски број мањи од 4000 претвара у римски.
- Саставити програм који римски број претвара у арапски.
- Саставити програм који број у систему са основом m претвара у број у систему са основом n.
- Саставити програм који одређује производ два цела броја са двадесет и више цифара.
- Саставити програм који одређује збир бројева од m до n чија је последња цифра 7, а прва није 3.
- Саставити програм који одређује са колико јединица се декадни број n пише у бинарном бројном систему.
- Саставити програм који исписује све двоцифрене бројеве који се могу написати као збир квадрата два броја.
- Саставити програм који исписује све троцифрене бројеве чија је средња цифра једнака суми прве и последње.

- ***Саставити програм који исписује све троцифрене бројеве који су једнаки збиру факторијела својих цифара.***
- ***Саставити програм који за текст написан у МетоВох-у одређује:***
  - Прву реч која почиње великим словом К
  - Број речи које почињу и завршавају истим словом
  - Број великих слова у тексту
  - Број појављивања речи Delphi и број појављивања речи Pascal
- ***Саставити програм који за текст написан у МетоВох-у у други МетоВох исписује:***
  - Све речи које завршавају малим словом е
  - Све речи из текста у обрнутом поретку слова
  - Све речи из текста са великим почетним словом
  - Све речи које почињу и завршавају истим словом
  - Исти текст, али из кога су избрисани сви симболи *
  - Исти текст, али у коме су слова с замењена симболом *
- ***Саставити програм који одређује вредност суме:***

$$a = 1 + \frac{1}{1+\frac{1}{2}} + \frac{1}{1+\frac{1}{2}+\frac{1}{3}} + \ldots + \frac{1}{1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n}}$$

- ***Саставити програм који одређује вредност израза:***

$$a = 1 + \cfrac{1}{2 + \cfrac{1}{3 + \cfrac{1}{\ldots + \cfrac{1}{n}}}}$$

- ***Саставити програм који исписује цифарску пирамиду:***

```
          1
        1 2 1
      1 2 3 2 1
    1 2 3 4 3 2 1
      ⋮
```

- ***Саставити програм који одређује редни број првог елемента низа задатог формулом:*** $a_n = \frac{x^n}{n!}, x \in R, n \in N$ ***који је мањи од произвољног, унапред задатог броја*** ε.