ЛАБАРАТОРНАЯ РАБОТА № 7

Работа з радковымі велічынямі

Мэта: сфарміраваць асноўныя ўменні праграмавання з выкарыстаннем радкоў. Азнаёміць з аперацыямі прысвойвання, злучэння, адносін, убудаванымі працэдурамі і функцыямі апрацоўкі радкоў, прыкладамі выкарыстання радковых пераменных.

Даныя тыпу *string (радкі сімвалаў)*, як і лікавыя даныя, падраздзяляюцца на канстанты і пераменныя. Радковыя канстанты – гэта паслядоўнасць сімвалаў, узятых у апострафы. Радковыя канстанты могуць быць апісаны ў раздзеле канстант *const='99';*. У мове рэалізуюцца таксама пераменныя тыпу *string*, якія з'яўляюцца пашырэннем стандарту мовы Паскаль. Апісанне радковых пераменных мае выгляд:

var t:string[10]; s:string;.

У Turbo Pascal пераменныя тыпу *string[n]* займаюць $n+1$ байт. Радковыя пераменныя аналагічныя масівам тыпу *char*. Іх адрозненнем з'яўляецца тое, што лік сімвалаў (ці бягучая даўжыня радка) можа дынамічна мяняцца ад нуля да зададзенага верхняга значэння n. Як і ў масівах, да асобных сімвалаў радка можна звярнуцца з дапамогай індэксаў у квадратных дужках: *s[5]*. Нулявы індэкс вызначае пазіцыю, у якой змяшчаецца бягучая даўжыня радка.

Дапускаецца таксама выкарыстанне тыпізаваных канстант радковага тыпу:

const s1:string[6]:='Весна'; s1:string[2]:=#67#68;(тоже что і 'CD')
s1:string[6]:='1'; .

Радковыя даныя могуць удзельнічаць у радковых выразах, якія складаюцца з радковых канстант, пераменных тыпу *string*, знакаў аперацый і ўбудаваных функцый. Пры гэтым над радковымі данымі дапускаюцца наступныя аперацыі: прысвойванне, злучэнне і адносіны.

Аперацыя прысвойвання. Агульны выгляд гэтай аперацыі наступны:

імя_радковай_пераменнай:=радковы_выраз;.

Напрыклад: *s1:='кі'; s2:='інфарматыка і ВТ'; s1:=s2;*.

Калі даўжыня радковага выразу перабольшвае максімальную даўжыню радковай пераменнай, то ўсе лішнія сімвалы справа адкідваюцца.

Асобныя сімвалы радковых пераменных можна прысвойваць сімвальным пераменным (пераменным тыпу *char*).

Аперацыя злучэння. Гэта аперацыя прымяняецца для злучэння некалькіх радкоў у адзін выніковы. Для абазначэння аперацыі злучэння выкарыстоўваецца знак "+".

Аперацыі адносін. Аперацыі адносін (=,<>,<,>) ажыццяўляюць параўнанне дзвюх радковых велічынь і маюць прыярытэт больш нізкі, чым аперацыя злучэння. Параўнанне радкоў выконваецца злева направа да першага несупадаючага сімвала, і той радок лічыцца большым, у якім першы несупадаючы сімвал мае большы нумар у кодавай табліцы ПЭВМ. Вынік выканання аперацый адносін над радковымі велічынямі заўсёды мае лагічны тып і прымае значэнне *true* (праўдзіва) ці *false* (ілжыва). Радкі лічацца роўнымі, калі яны цалкам супадаюць па бягучай, а не аб'яўленай даўжыні і змяшчаюць адны і тыя ж сімвалы. Неабходна таксама улічваць, што аднолькавыя вялікія і малыя літары ў радках вызначаюць розныя значэнні радковых пераменных.

Для апрацоўкі радковых даных выкарыстоўваюцца прыведзеныя ніжэй стандартныя **радковыя працэдуры**:

delete(s,p,k) – выдаленне k сімвалаў у радку s, пачынаючы з пазіцыі p;

insert(s1,s2,p) – устаўка радка $s1$ у радок $s2$, пачынаючы з пазіцыі p;

str(N,s) – пераўтварэнне лікавага значэння N у радок s;

val(s,N,c) пераўтварае значэнне s у велічыню цэлалікавага ці рэчавага тыпу і змяшчае вынік у N, c – нумар месца збою.

Пры рабоце з радковымі велічынямі выкарыстоўваюцца наступныя **стандартныя функцыі:**

length(s) – вызначае дліну радка s;

copy(s,p,c) – з пазіцыі p радка s выразае c сімвалаў;

pos(s1,s2) – вызначае нумар месца першага з'яўлення радка $s1$ у радку $s2$;

upcase(ch) – пераўтварае малую літару ў вялікую (толькі лацінскія);

concat(s1,s2,…,sn) – выконвае склейку радкоў $s1$, …, sn у парадку, указаным у дужках (гл. аперацыю "+").

Прыклад 1. Вызначым, ці ёсць у тэксце спалучэнне літар "ab":

```
program slr;
    const N=10;
    var S,V:string[N]; I,M,k,f:integer;
BEGIN
    writeln('Увядзіце радок сімвалаў'); readln(S);
    m:=pos('ab',s);
    if m=0 then writeln('няма') else writeln('ёсць');
END.
```

Прыклад 2. У дадзеным сказе літару "о" выдалім, а літару "а" заменім на "*":

```
program qvit1;
    const N=30;
    var S:string[N]; f:string; i,M:integer;
BEGIN
    writeln('Увядзіце радок сімвалаў'); readln(S);
    f:=''; {пустата – два апострафы без прабела}
    for i:=1 to length(S) do if S[I]<>'o' then f:=f+s[i];
    s:=f;
    for i:=1 to length(s) do if s[i]='a' then s[i]:=’*’;
    writeln(s);
END.
```

Прыклад 3. Вызначым суму лічбаў натуральнага ліку:

```
program ghit;
    const N=30;
    var S:string[N]; i,K,f,j:integer;
BEGIN
    K:=0; writeln(' Увядзіце лік); readln(S);
    for i:=1 to length(s) do begin val(s[i],f,j); K:=K+f; end;
    writeln(K);
END.
```

Прыклад 4. У дадзеным сказе заменім спалучэнне літар "ab" шматкроп’ем:

```
program slr;
    const N=10;
    var S,V,T:string;
        I,M,k,f:integer;
BEGIN
    writeln('Увядзіце радок сімвалаў'); readln(S);
    t:=''; i:=1;
    while i<=length(s) do
          if copy(s,i,2)='ab' then begin t:=t+'...'; i:=i+2 end
                             else begin t:=t+s[i]; i:=i+1 end;
    writeln(T);
END.
```

Прыклад 5. Дадзены тэкст да 10 слоў, у якім словы (ад 1 да 8 літар) могуць быць аддзелены адной ці некалькімі зорачкамі. Выдалім усе зорачкі ў пачатку і ў канцы і пакінем паміж словамі адну зорачку, усе словы, што ўтрымліваюць тры сімвалы, заменім на слова "тры":

```
program gfhd;
```

```
    const s='****a*bc***efg**k*';
    var t,w,b:string;
        i,k,n:integer; sl:array[1..10] of string[8];
BEGIN
    t:=s;
    writeln('------зыходны тэкст------'); writeln(s);
    {выдаляем усе зорачкі ў пачатку тэксту}
    while(t[1]='*') and (length(t)>0) do delete(t,1,1);
    {выдаляем усе зорачкі ў канцы тэксту}
    while (t[length(t)]='*') do delete(t,length(t),1);
    k:=0;{лічыльнік зорачак паміж словамі}
    n:=0;{ лічыльнік слоў}
    b:='';
    t:=t+'*'; {дадаем * для атрымання апошняга слова}
    { атрыманне масіву слоў з тэксту}
    for i:=1 to length(t) do
        if t[i]<>'*' then begin b:=b+t[i]; k:=0; end
                     else begin k:=k+1;
                                if k=1 then begin n:=n+1; sl[n]:=b; b:=''; end;
                          end;
    writeln('-----вывад усіх слоў на экран-----');
    for i:=1 to n do writeln(sl[i]);
    {замяняем словы}
    t:='';
    for i:=1 to n do
        if length(sl[i])=3 then t:=t+'тры'+'*'
                           else t:=t+sl[i]+'*';
    {выдаляем зорачку ў канцы тэксту}
    delete(t,length(t),1);
    writeln('------адказ------');
    writeln(t);
END.
```