

ОБРАБОТКА НА РЕДИЦИ ОТ ЧИСЛА

Румяна Жекова

МГ ‘Баба Тонка’, Русе 7000, ул. Иван Вазов 18
rgekova@abv.bg

Обработката на редици (от числа) може да се раздели на 3 основни вида: всички елементи се обработват по един и същ начин, един или няколко, но предварително известен брой елементи (обикновено предварително) и неопределен брой елементи се обработват по различен начин.

Ключови думи: редица, елемент, начин, цикъл, брой

Въпреки многообразието от учебници и учебни помагала по информатика считам, че днес са много малко помощните материали за учителите по този предмет. Надявам се следващите думи от статията да бъдат полезни за някои от настоящите и бъдещи преподаватели.

В часа по информатика се стига до понятието цикъл. Дава се определение. И нататък? Прави се преглед на различните видове цикли и основните понятия, свързани с тях, разликите между отделните конструкции: условие и управляваща променлива; тялото на цикъла се изпълнява поне веднъж или може да не се изпълни изобщо; наличието на предусловие означава, че клетките, които участват в това условие трябва вече (преди цикъла) да са получили някаква стойност; възможността от безкраен цикъл (зацикляне); прекъсването на цикъл; преход в края на цикъл... След това?

Циклите са свързани с редици от данни, например числа, независимо дали тези числа се въвеждат или изчисляват, независимо дали получените числа се извеждат или представляват само елемент от друга дейност. В началото може да се поговори за редиците, при които предварително е известен броя на елементите – тези редици ще се обработват с цикъл с управляваща променлива. След това се споменава за редиците, при които краят се задава чрез някакво друго условие: нулев елемент; отрицателно число; елемент, който е по-голям (по-малък) от дадена стойност.

Следващото разделяне на работата с редици от числа (без да се използват масиви или динамични променливи), което предлагам малко по-подробно, може да бъде следното:

1. Първи тип – всички елементи от редицата се обработват по един и същ начин. Задачи: Намиране на сума, средно аритметично, произведение на всички въведени числа или на числата от даден интервал (положителни, отрицателни, между две дадени числа); преброяване на числата отговарящи на да-

ден признак (делимост или друг); изчисляване на A^N , където N е цяло число; изчисляване на $N!$, $N!!$ и така нататък.

<pre> Program P1_1; { Pascal 7.0 } Var N, K, F : Longint; BEGIN Writeln('Изчисляване на факториел'); Write('Въведете число /от 0 до 10/: '); Readln(N); F := 1; For K := 1 To N Do F := F * K; Writeln(N, ' ! = ', F); END. </pre>	<pre> #include <iostream.h> // C++ void main() { int n,f; cout <<"Изчисляване на факториел\n"; cout <<"Въведете число /от 0 до 10/: "; cin >> n; f = 1; for (int k = 1; k <= n ; k++) f=f*k; cout <<"\n" <<n <<"!=" <<f; } </pre>
--	--

<pre> Program P1_2; { Pascal 7.0 } Var N, K : Integer; A, AN : Real; BEGIN Writeln('Число на цяла степен'); Write('Въведете основа: '); Readln(A); Write('степенен показател: '); Readln(N); AN := 1; Write(A, ' на степен ', N, ' е равно на '); If N < 0 Then Begin If A = 0 Then Begin Writeln('Невъзможно действие!'); Exit End Else Begin A:=1/A; N := -N End; End; For K := 1 To N Do AN := AN * A; Writeln(AN) END. </pre>	<pre> #include <iostream.h> // C++ void main() { int n; double a,an; cout <<"Число на цяла степен "; cout <<"\n Въведете основа: "; cin >> a; cout <<"\n степенен показател: "; cin >> n; an = 1; cout<<"\n"<<a<<" на степен "<<n <<"=" ; if (n < 0) { if (a == 0) { cout <<"\n Невъзможно действие!"; return () } else { a = 1/a; n = -n;} } for (int k=1 ; k<=n ; k++) an = an * a; cout <<"\n" <<an ; } </pre>
--	--

Стандартния алгоритъм за решаване на задачите от този тип е: Всички действия се извършват в един цикъл. В него в една и съща клетка се въвежда (изчислява) всеки елемент от редицата (ако елементът за работа не е известен), прави се или не проверка и числото (евентуално само това число, което отговаря на условието) се преброява, сумира, умножава със съдържанието на една и съща клетка. В тази клетка преди започването на цикъла се поставя неутрална стойност (най-често нула или единица). Все пак след като се въве-

де нов елемент, стойността на стария се губи безвъзвратно. След приключване на цикъла не се въвеждат (изчисляват) повече елементи от редицата.

2. Втори тип – един или няколко, но предварително ясен брой елементи от редицата се обработват по различен начин преди самия цикъл. Задачи: най-малко, най-голямо по абсолютна стойност число; числото, което е най-близо до предварително зададена стойност; редицата на Фибоначи; аритметична и геометрична прогресия; проверка за подреденост в дадена редица (растяща, строго растяща, намаляваща, строго намаляваща, равни числа, неподредена редица); ...

```
Program P2_1; { Pascal 7.0 }
Var N, l : Integer; A, Min : Real;
BEGIN
Writeln('Намиране на min от N числа');
Write('Въведете броя на числата: ');
Readln(N);
Write('Въведете число: '); Readln(A);
Min := A;
  For l := 2 To N Do Begin
    Write('Въведете число: '); Readln(A);
    If A < Min Then Min := A End;
  Writeln('Най-малкото число е: ',Min)
END.
```

```
#include <iostream.h> // C++
void main()
{ int n; double a,min;
  cout <<"Намиране на min от N числа ";
  cout <<"\n Въведете броя на числата: ";
  cin >> n;
  cout <<"\n Въведете число: ";cin >> a;
  min = a;
  for ( int k=2; k<=n; k++)
  { cout <<"\n Въведете число: ";cin >> a;
    if ( a < min ) min = a; }
  cout <<"\nНай-малкото число е:" <<min;
}
```

```
Program P2_2; { Pascal 7.0 }
Var A, B, C, N, l : Integer;
BEGIN
Writeln('N-то число на Фибоначи');
Write
  ('Въведете номера на числото: ');
Readln(N);
A := 1; B := 1;
For l := 3 To N Do
Begin C := A + B ; A := B; B := C; End;
If N < 3 Then C := 1;
Writeln
(N, ' -тото число на Фибоначи-',C)
END.
```

```
#include <iostream.h> // C++
void main()
{ int n,a,b,c;
  cout <<" N-то число на Фибоначи ";
  cout<<"\n"
  <<"Въведете номера на числото:";
  cin >> n;
  a = 1; b=1;
  for ( int k=3; k<=n; k++)
  { c = a + b; a = b ; b = c };
  if ( n < 3 ) c = 1;
  cout <<"\n"
  <<n <<"-тото число на Фибоначи-" <<c;
}
```

При този тип задачи стандартния алгоритъм отново включва само един цикъл. Преди него обаче се въвеждат (изчисляват) едно, две или повече, но предварително известен брой елементи и има клетки (най-често за сравне-

ние), чиято стойност се зарежда преди цикъла със стойността на елемент от редицата. В цикъла освен въвеждане (изчисляване) в една и съща клетка има сравнение, “остаряване” на данните (прехвърляне на стойности в други клетки). Нали след въвеждане на нов елемент, стойността на стария се губи безвъзвратно. След приключване на цикъла не се въвеждат (изчисляват) повече елементи от редицата.

3. Трети тип – няколко, но предварително неизвестен брой елементи от редицата се обработват по различен начин. Задачи: най-малкото положително число, най-голямото отрицателно число; най-малкото число по-голямо от предварително зададена стойност; броя на отрицателните числа след първата срещната нула; ...

<pre> Program P3_1; { Pascal 7.0 } Var N, K, J : Integer; A, Min : Real; BEGIN Writeln ('Намиране на най-малкото ', 'положително от N дадени числа'); Write('Въведете броя на числата: '); Readln(N); For K := 1 To N Do Begin Write('Въведете число: '); Readln(A); If A > 0 Then Break; End; If A <= 0 Then Begin Writeln('Няма положителни числа. '); Exit End; For J := K + 1 To N Do Begin Write('Въведете число: '); Readln(A); If A > 0 Then If A < Min Then Min := A End; Writeln('Търсеното число е ',Min) END. </pre>	<pre> #include <iostream.h> // C++ void main() { int n,k,j; double a,min; cout <<"Намиране на най-малкото " <<"положително от N дадени числа" ; cout <<"\n Въведете броя на числата. "; cin >> n; for (k = 1; k <= n ; k++) { cout <<"\n Въведете число: "; cin >>a; if (a > 0) break ;} if (a <= 0) { cout <<"\nНяма положителни числа. "; return () ; } for (j = k+1; j <= n ; j++) { cout <<"\n Въведете число: "; cin >>a; if (a > 0) if (a < min) min = a ; } cout <<"Търсеното число е " << min ; } </pre>
---	---

Стандартния алгоритъм вече включва два цикъла. В първия се въвеждат (изчисляват) елементи от редицата, за да може някои клетки (най-често за сравнение), необходими за работата на алгоритъма, да се заредят преди цикъла с подходящи стойности, зависещи от стойността на елемент от редицата. Останалите елементи от редицата (ако има такива) се въвеждат (изчисляват) във втория цикъл в една и съща клетка и следва сравнение (много често от две условия), както и други обработки. След приключване на двата цикъла не се въвеждат (изчисляват) повече елементи от тази редица.

Едва след тези основни моменти може да се пристъпи към по-сложни задачи изискващи по-сериозни самостоятелни разсъждения.

Все пак реалните типови задачи много често са свързани с обработка на големи редици (бази) от данни и програмистите трябва да са наясно с идеите за различните им обработки.

Литература:

1. Азълов П., Златарова Ф., Сборник информатика с Паскал в примери, тестове и задачи, София, Просвета, 1996 г
2. Василев Цв., Христова Пл., Марков М., Григорова К., Ръководство за упражнения по Програмиране и използване на компютрите – II част, Русе, Русенски университет "Ангел Кънчев", 2005 г
3. Тодорова М., Програмиране на C++ част 1., София, СИЕЛА, 2004 г