

Алгоритми за търсене в масив на Паскал

Публикувано от elena_pazardjik на 21.08.2012

1. Алгоритъм за търсене в масив.

Едно от основните и най-често използвани действия, когато се обработва структура от данни е търсене на елемент по зададено условие. Това предполага намирането на един или няколко елемента, така че те да отговарят на критерия за търсене. Ако структурата, която ще бъде претърсвана се състои от еднотипни елементи е удобно тя да се представи като масив.

а) последователно търсене - при този метод на действие се обхожда цялата структура от данни;

Задача 1:

Да се състави програма, която въвежда максимум 50 цели числа, след което извежда положителните.

За решаването ще бъде необходимо да се дефинира едномерен масив - X, който ще съдържа максимум 50 елемента от тип целочислен. В началото на програмата ще трябва да се осигури възможност за въвеждане на действителния брой елементи - N, с които ще се работи (осигурено е въвеждането на броя задължително да попадне в интервала 1 - 50). След като бъдат въведени стойности в индексирани елементи на масива, той ще трябва да бъде обходен отново за да се отсеят положителните. Това се явява критерия за търсене - съответния елемент да е по-голям от нула: $X > 0$. Самата проверка ще бъде като условие в кратка форма на условен оператор. Предвидено е действие само ако се намери елемент, отговарящ на условието.

VAR x : array [1 .. 50] of integer ;

```

i, N : 1 .. 50 ;
BEGIN
  repeat
  write ( ' Въведете броя на елентите: ' ) ;
  readln ( N ) ;
  until ( N >=1 ) and ( N <= 50 ) ;
  for i := 1 to N do
  begin
    write ( ' x[ ' , i , ' ]= ' ) ;
    readln ( x ) ;
  end ;
  for i := 1 to N do
  if x > 0 then write( x );
  readln ;
END.

```

б) *ускорено търсене - при този метод на действие, обхождането на структурата от данни се прекратява при достигане на елемент, отговарящ на поставеното условие;*

Задача 2:

Да се състави програма, която въвежда максимум 50 цели числа, след което проверява дали числото 10 е сред въведените.

Отново ще имаме едномерен масив X с максимум 50 елемента, които са от тип целочислен, N ще бъде действителния брой елементи. При обхождането на масива за откриване на числото 10 ще бъде използван оператор за цикъл с условие. Целта на това използване е да се прекрати търсенето ако бъде намерен елемент, отговарящ на условието $X = 10$. Ако такъв не се намери, цикълът ще трябва да се прекрати при достигане края на масива $i = N \Rightarrow$ условието за край ще бъде образувано от две прости условия.

```

VAR x : array [1 .. 50] of integer ;
i, N : 1 .. 50 ;
BEGIN
  repeat
  write ( ' Въведете броя на елентите: ' ) ;
  readln ( N ) ;
  until ( N >=1 ) and ( N <= 50 ) ;
  for i := 1 to N do
  begin

```

```

write ( ' x[ ', i , ' ]= ' );
readln ( x );
end ;
i := 0 ;
repeat
i := i + 1 ;
until ( x = 10 ) or ( i = n ) ;
if x = 10 then writeln ('Числото 10 се среща в масива')
else writeln ('Числото 10 не се среща в масива');
readln ;
END.

```

2. Алгоритъм за броене в масив.

Този тип алгоритъм е много близък до алгоритъм за търсене. Допълнителното в случая е необходимостта от променлива (обикновено *br*), в която трябва да се натрупа броя на елементите, отговарящи на зададеното условие. Преди да се започне обхождане на структурата за търсене на елементите, на *br* се дава първоначална стойност 0, след това при всяко срещане на елемент, отговарящ на критерия стойността на *br* се увеличава с едно: *br* := *br* + 1 ; След прекратяване на обхождането *br* ще трябва да се изведе за да бъде съобщен резултата.

Задача 3:

Да се състави програма, която въвежда максимум 100 цели числа, след което намира и извежда броя на тези, които се делят на 3.

```

VAR x : array [1 .. 100] of integer ;
i, N : 1 .. 100 ;
br : 0 .. 100 ;
BEGIN
repeat
write ( ' Въведете броя на елементите: ' ) ;
readln ( N ) ;
until ( N >= 1 ) and ( N <= 100 ) ;
for i := 1 to N do
begin

```

```

write ( ' x[ ' , i , ' ]= ' );
readln ( x );
end ;
br := 0 ;
for i := 1 to N do
if x mod 3 = 0 then br := br + 1 ;
writeln ( 'Броят на числата, делищи се на 3 е: ' , br ) ;
readln ;
END.

```

3. Алгоритъм за намиране на максимален / минимален по стойност елемент в масив.

Приема се, че първия елемент от структурата отговаря на условието и се запамятава в помощна променлива $\min / \max$ ($\min = X[1] / \max = X[1]$). След това се претърсва структурата от втория елемент до края. Ако се намери елемент със стойност по-малка / по-голяма от стойността на $\min / \max$, то променяме $\min / \max$ ($\min = X / \max = X$). *Така след цялостното обхождане в двете променливи ще бъдат запаметени най-малката и най-голяма стойност.*

Задача 4:

Да се състави програма, която въвежда максимум 60 числа, след което намира и извежда най-малката и най-голяма въведена стойност.

```

VAR x : array [1 .. 60] of real ;
i, N : 1 .. 60 ;
min, max : real ;
BEGIN
  repeat
  write ( ' Въведете броят на елементите: ' ) ;
  readln ( N ) ;
  until ( N >= 1 ) and ( N <= 60 ) ;
  for i := 1 to N do

```

```
begin
write ( ' x[ ' , i , ' ]= ' );
readln ( x );
end ;
min := x[1] ;
max := x[1] ;
for i := 2 to N do
begin
if x < min then min := x ;
if x > max then max := x ;
end ;
writeln ('Най-малката въведена стойност е: ' , min) ;
writeln ('Най-голямата въведена стойност е: ' , max) ;
readln ;
END.
```