

Поиск элементов массива.

Схема поиска.

If < условие поиска > then

Условию предшествует цикл просмотра.

Поиск элементов массива осуществляется по различным условиям, изменения в программе будут зависеть от количества условий и, соответственно, будут содержать несколько условных операторов.

Задача.

Найти все нечетные по значению элементы массива, стоящие на местах с четными номерами. Массив состоит из n целых чисел. Вывести на печать значения элементов и их индекс. Ввод элементов массива производится с клавиатуры.

Математическая постановка задачи.

Дано: $a[n]$;

Надо: вывести на печать четные элементы и их индексы (i).

Математическая модель.

```
If  $a[2*i] \bmod 2 \neq 0$  then writeln('a[' $i$ ']=',  $a[i]$ , 'i=':8,  $i$ );
```

```
program massiv4;  
{‘ Описание элементов массива ‘};  
var  
a: array [1 .. 1000] of integer;  
l, n : integer;  
begin  
{‘ Ввод размера массива ‘};  
write(‘ Введите размер массива ‘);  
readln(n);  
{‘ Ввод значений элементов массива ‘};  
for i := 1 to n do  
begin  
write(‘ Введите a[‘,i,’ ‘);  
read(a[i]);  
end;  
{‘ Поиск нужных элементов ‘};  
for i:= 1 to n div 2 do  
If a[2*i] mod 2 <> 0 then {‘вывод результатов на печать ‘}  
writeln(‘a[‘,i,’]=’,a[i]:8, ‘i=’,i);  
end.
```

**Поиск отрицательных элементов массива,
стоящих на нечетных местах.**

If (a[i] < 0) and (i mod 2 <> 0) then

или

If a[2*i-1] < 0 then

Поиск максимального значения элемента и его индекса.

Математическая постановка задачи.

Дано: $a[10]$;

Надо: \max , i_{\max} .

Математическая модель.

$\max := a[1]$; $i_{\max} := 1$;

If $a[i] > \max$ then begin $\max := a[i]$; $i_{\max} := i$; end;

```
Program elmax;  
{‘Описание элементов массива и других переменных ’};  
var  
a: array[1..10] of integer;  
i, max, imax : integer;  
begin  
{‘Ввод элементов массива’};  
for i:=1 to 10 do  
read(a[ i ]);  
{‘Подготовка данных к обработке’};  
max:=a[1]; imax:=1;  
{‘Обработка элементов массива’};  
for i:=2 to 10 do  
If a[i] > max then begin max:=a[i]; imax:=i; end;  
writeln(‘max=’, max, ‘imax=’, imax);  
readln;  
end.
```

Поиск минимального значения элемента и его индекса.

Математическая постановка задачи.

Дано: $a[10]$;

Надо: \min , imin .

Математическая модель.

$\min := a[1]$; $\text{imin} := 1$;

If $a[i] < \min$ then begin $\min := a[i]$; $\text{imin} := i$; end;

Проиллюстрируем использование одномерных массивов при решении задач.

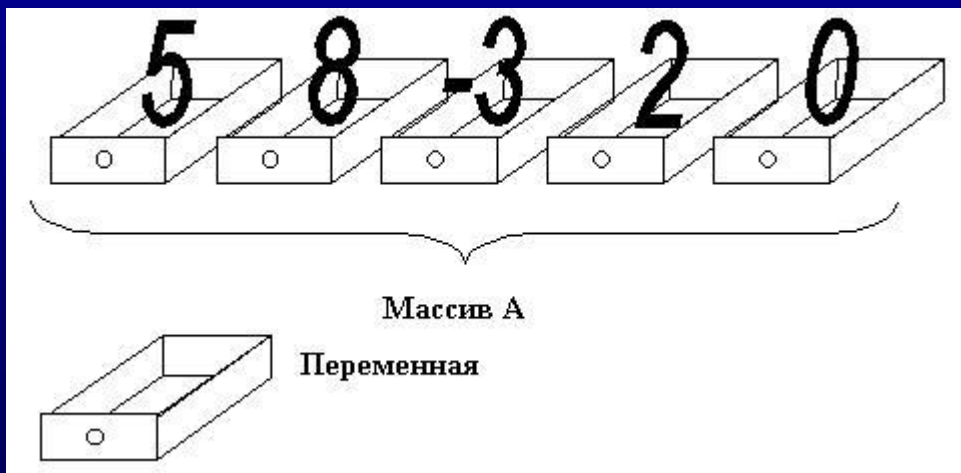


Рис.1. Исходные массив и переменная

В качестве минимального элемента возьмем первый элемент массива,

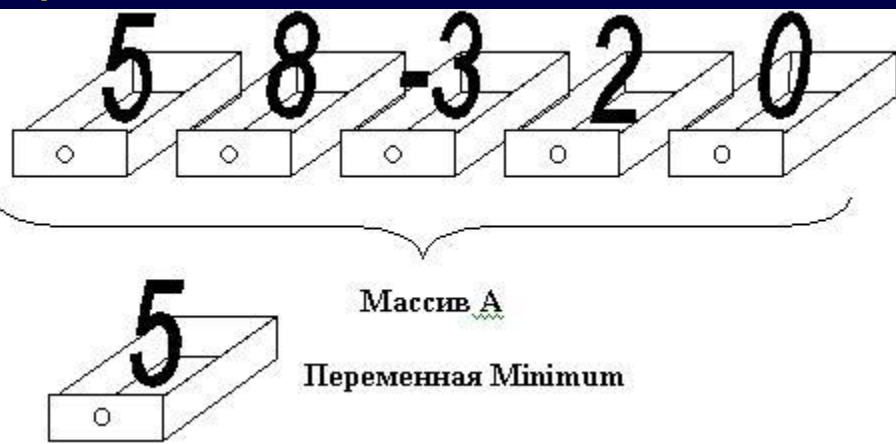


Рис.2. В качестве минимального берем первый элемент

После завершения просмотра всех элементов массива переменная Minimum будет содержать значение минимального элемента массива:

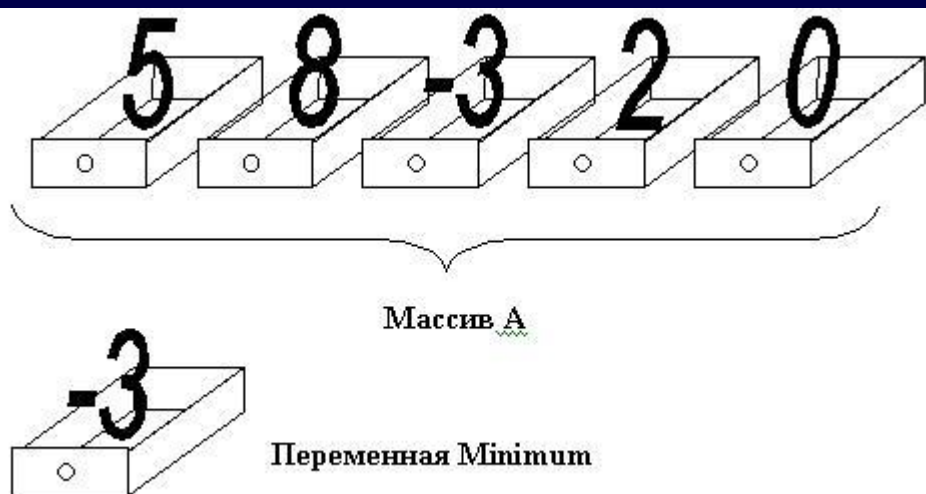


Рис.3. Переменная Minimum содержит минимальный элемент


```
Program elmin;  
{‘Описание элементов массива и других переменных ’};  
var  
a: array[1..10] of integer;  
i, min, imin : integer;  
begin  
{‘Ввод элементов массива’};  
for i:=1 to 10 do  
read(a[ i ]);  
{‘Подготовка данных к обработке’};  
min:=a[1]; imin:=1;  
{‘Обработка элементов массива’};  
for i:=2 to 10 do  
If a[i] < min then begin min:=a[i]; imin:=i; end;  
writeln(‘min=’, min, ‘imin=’, imin);  
readln;  
end.
```

Технология решения задачи 25 ЕГЭ

Задачи этого типа направлены на обработку массива по некоторым критериям. Полностью программа состояла бы из:

- 1) Описания массива и типов используемых переменных, организация ввода данных (*уже есть*)
- 2) Инициализации начальных значений некоторых переменных (*требуется задать!*)
- 3) Обработки данных (*требуется организовать!*)
- 4) Вывода данных (*требуется организовать!*).

Однако, в 25 задаче учащемуся необходимо не писать программу полностью, а «дописать» её в рамках уже организованного ввода, а также заданного количества переменных и их типов.

Что может обрабатываться в массиве?

элементы массива по одному

пары элементов массива

тройки элементов массива

последовательности элементов

Обработка данных происходит посредством циклического перебора элементов массива и поиска по некоторому комбинированному условию, которое учащемуся необходимо выявить на основе анализа условия задачи.

(например, элемент массива не делится на 2 и кратен 3).

Примеры формализации некоторых типичных для задачи

25

составляющих комбинированного условия:

Признак

Пример его формального описания для
некоторого элемента массива A
(на языке Паскаль)

принадлежность к положительным
/отрицательным числам

$A[i]>0 / A[i]<0$

четность/нечетность

$A[i] \bmod 2 = 0$

наличие указанного числа знаков в
числе (только двузначное или др.)

(трехзначное натуральное число)
 $(A[i] \geq 100) \text{ and } (A[i] \leq 999)$

наличие указанной цифры в конце
(начале) числа

(средняя цифра трехзначного числа
– 5)
 $(A[i] \bmod 100) \text{ div } 10 = 5$

кратность некоторому числу

(делится на 7 нацело) $A[i] \bmod 7 = 0$

окончание числа на указанные
цифры

(оканчивается на 5) $A[i] \bmod 10 = 5$

соответствие указанному значению
абсолютного значения элементов
массива

(элемент массива больше элемента
слева и элемента справа – поиск
«пика» в тройке элементов)
 $(A[i-1] < A[i]) \text{ and } (A[i] > A[i+1])$

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов:

6; 2; 9; -3; 6 ответ: 4.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Паскаль

```
const
```

```
N = 20;
```

```
var
```

```
a: array [1..N] of integer;
```

```
i, j, k: integer;
```

```
begin
```

```
for i := 1 to N do
```

```
  readln(a[i]);
```

```
  ...
```

```
end.
```

Опишите ... алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3.

```
k := 0;
```

```
for i := 1 to N-1 do
```

```
  if (a[i] mod 3=0) or (a[i+1] mod 3=0)
```

```
  then k:=k+1;
```

```
  writeln(k);
```

Паскаль

```
const
N=2000;
var
a: array [1..N] of integer;
i, j, k: integer;
begin
for i:=1 to N do
readln(a[i]);

...
end.
```

дан массив, содержащий 2000 неотрицательных целых чисел. Пиком называется не крайний элемент массива, который больше обоих своих соседей. Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите «пика нет».

Например, в массиве из шести элементов, равных соответственно 5, 9, 2, 18, 3, 7, есть два пика – 9 и 18, максимальный пик равен 18.

Паскаль

```
const
N=2000;
var
a: array [1..N] of integer;
i, j, k: integer;
begin
for i:=1 to N do
readln(a[i]);

...
end.
```

Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите «пика нет».

```
k:=0;
```

```
If k=0 then
writeln('пика нет') else
writeln(k);
```


Паскаль

```
const
N=2000;
var
a: array [1..N] of integer;
i, j, k: integer;
begin
for i:=1 to N do
readln(a[i]);

...
end.
```

```
k:=0;//макс пик
```

```
if k=0 then writeln('пика нет')
else writeln(k);
```

Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите «пика нет».

```
for i:=2 to N-1 do
begin
if (a[i]>a[i-1]) and
(a[i]>a[i+1]) and (a[i]>k)
then k:=a[i];
and;
```

Паскаль

```
const
N=2000;
var
a: array [1..N] of integer;
i, j, k: integer;
begin
for i:=1 to N do
readln(a[i]);

...
end.
```

Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите «пика нет».

Паскаль ABC.net

```
k := 0;
for i := 2 to N-1 do
begin
if (a[i]>a[i-1]) and (a[i]>a[i+1]) and
(a[i]>k)
then k:=a[i];
and;
if k=0 then writeln('пика нет')
else writeln(k);
```

Надо помнить:

- 1. При нахождении **максимального** значения элемента, необходимо в подготовке данных к обработке присвоить максимальному элементу самое **маленькое значение (0, -1)**.**
- 2. При нахождении **минимального** значения элемента, необходимо в подготовке данных к обработке присвоить минимальному элементу **самое большое значение +1**.**