

Программирование на языке Си

Алгоритмы обработки массивов

Поиск в массиве

Найти элемент, равный X:

```
i = 0
while A[i] != X:
    i += 1
print ( "A[" , i , "]" = " , X , sep = " " )
```



Что плохо?

```
i = 0
while i < N and A[i] != X:
    i += 1
if i < N:
    print ( "A[" , i , "]" = " , X , sep = " " )
else:
    print ( "Не нашли!" )
```



Что если такого нет?

Поиск в массиве

Вариант с досрочным выходом:

номер найденного
элемента

```
nX = -1
for i in range ( N ):
    if A[i] == X:
        nX = i
        break
if nX >= 0:
    print ( "A[" , nX, "]" = " , X, sep = " " )
else:
    print ( "Не нашли!" )
```

досрочный
выход из цикла

Поиск в массиве

Варианты в стиле Python:

```
for i in range ( N ) :  
    if A[i] == X:  
        print ( "A[" , i , "]" = " , X , sep = " " )  
        break  
else:  
    print ( "Не нашли!" )
```

если не было досрочного выхода из цикла

```
if X in A:  
    nX = A.index (X)  
    print ( "A[" , nX , "]" = " , X , sep = " " )  
else:  
    print ( "Не нашли!" )
```

Максимальный элемент

```
M = A[0]
for i in range(1, N):
    if A[i] > M:
        M = A[i]
print ( M )
```



Если `range(N)` ?

Варианты в стиле Python:

```
M = A[0]
for x in A:
    if x > M:
        M = x
```



Как найти его номер?

```
M = max ( A )
```

Максимальный элемент и его номер

```
M = A[0] ; nMax = 0
for i in range(1, N):
    if A[i] > M:
        M = A[i]
        nMax = i
print ( "A[" , nMax , "]" = " , M , sep = " " )
```



Что можно улучшить?



По номеру элемента можно найти значение!

```
nMax = 0
for i in range(1, N):
    if A[i] > A[nMax]:
        nMax = i
print ( "A[" , nMax , "]" = " , A[nMax] , sep = " " )
```

Максимальный элемент и его номер

Вариант в стиле Python:

```
M = max (A)
nMax = A.index (M)
print ( "A[" , nMax , "]" = " , M , sep = " " )
```

номер заданного
элемента (первого из...)