

Лабораторная работа №7

Массивы.

Поиск в массиве

Используем цикл while:

```
import random # подключение библиотеки
from random import randint
n=10; x=5
mas = [randint(1,10) for i in range(n)] # инициализируем массив
i = 0
while i < n and mas[i] != x: # если элемент не равен
    i += 1
if i < n:
    print ( "mas[" , i, "]=", x, sep = "" )
else:
    print ( "Не нашли!" )
```

Используем цикл for:

```
import random
from random import randint
n=10;x=5
mas = [randint(1,10) for i in range(n)]

for i in range (n):
    if mas[i] == x:
        nomer = i
        break
if nomer >= 0:
    print ( "mas[" , nomer, "]=", x, sep = "" )
else:
    print ( "Не нашли!" )
```

В данном случае в переменной *nomer* сохраняется номер элемента массива с найденным значением.

Но на языке Python цикл for обладает уникальным свойством: у него есть блок else, который выполняется в том случае, если в цикле не применился оператор break.

Поэтому рассмотрим второй способ поиска, более простой:

```
import random
from random import randint
```

```

n=10;x=5
mas = [randint(1,10) for i in range(n)]

nomer = -1
for i in range (n):
    if mas[i] == x:
        print ( "mas[" , i, "]=", x, sep = "" )
        break

else:
    print ( "Не нашли!" )

```

Задание Python 7_1:

Дан массив. Необходимо подтвердить, что в массиве есть числа, кратные трем.

Задание Python 7_2:

Заполните массив случайными числами в диапазоне 0..4 и выведите на экран номера всех элементов, равных значению X (оно вводится с клавиатуры).

Поиск минимального или максимального элемента

```

import random
from random import randint
mas = [randint(1,10) for i in range(n)]

MaxEl = mas[0]
for i in range(1,n):
    if mas[i] > MaxEl:
        MaxEl = mas[i]

print (MaxEl)

```

В переменной *MaxEl* сохранится максимальный элемент массива.

Однако, для поиска максимального и минимального элементов массива в Python есть собственные функции *max*:

```

import random
from random import randint
mas = [randint(1,10) for i in range(n)]
MaxEl = max (mas)
print ( MaxEl )

```

Задание Python 7_3: Заполните массив случайными числами. Найдите номера первого минимального и последнего максимального элемента массива.

Сортировка массива в Python

МЕТОД ПУЗЫРЬКА

Сортировку массива в python будем выполнять [методом Пузырька](#):

```
import random
from random import randint

mas = [randint(1,10) for i in range(n)]
for i in range(n):
    print(mas[i],sep=" ")
print(" ")
for i in range(n-1):
    for j in range(n-2, i-1, -1):
        if mas[j+1] < mas[j]:
            mas[j], mas[j+1] = mas[j+1], mas[j]
for i in range(n):
    print(mas[i],sep=" ")
```

Задание Python 7_4:

Необходимо написать программу, в которой сортировка выполняется «методом камня» — самый «тяжёлый» элемент опускается в конец массива.

БЫСТРАЯ СОРТИРОВКА МАССИВА

Данную сортировку еще называют quick sort или сортировка Хоара (по имени разработчика — Ч.Э. Хоар).

```
import random
from random import randint
# процедура
def qSort ( A, nStart, nEnd ):
    if nStart >= nEnd: return
    L = nStart; R = nEnd
    X = A[(L+R)//2]
    while L <= R:
        while A[L] < X: L += 1 # разделение
        while A[R] > X: R -= 1
        if L <= R:
            A[L], A[R] = A[R], A[L]
            L += 1; R -= 1
```

```

        qSort ( A, nStart, R ) # рекурсивные вызовы
        qSort ( A, L, nEnd )

N=10
A = [randint(1,10) for i in range(N)]
print(A)
# вызов процедуры
qSort ( A, 0, N-1 )
print('отсортированный', A)

```

Задание Python 7_5: Необходимо написать программу, которая сортирует массив (быстрой сортировкой) по возрастанию первой цифры числа.

ВСТРОЕННЫЕ ФУНКЦИИ

mas.reverse() — стандартный метод для перестановки элементов массива в обратном порядке;

mas2 = sorted (mas1) — встроенная функция для сортировки массивов (списков);

Задание Python 7_6: Напишите программу, которая, не изменяя заданный массив, выводит номера его элементов в возрастающем порядке. Использовать вспомогательный массив номеров.

Задание Python 7_7: Напишите программу, которая сортирует массив и находит количество различных чисел в нём. Не использовать встроенные функции.

Задание Python 7_8: Дан массив. Назовем серией группу подряд идущих одинаковых элементов, а длиной серии — количество этих элементов. Сформировать два новых массива, в один из них записывать длины всех серий, а во второй — значения элементов, образующих эти серии.

Задание Python 7_9: Напишите вариант метода пузырька, который заканчивает работу, если на очередном шаге внешнего цикла не было перестановок. Не использовать встроенные функции.