Массивы(тип list)

А

1

2

3

4

5

Тип list

Тип int

A = [1, 2, 3, 4, 5]

for x in A:

print(x, type(x))

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print(A, type(A))

1 <class 'int'>

2 <class 'int'>

3 <class 'int'>

4 <class 'int'>

5 <class 'int'>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[1, 2, 3, 4, 5] <class 'list'>

Press any key to continue . . .

A = [1, 2, 3, 4, 5]

for x in A:

print(x, type(x))

x += 1

print(x, type(x))

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print(A, type(A))

1 <class 'int'>

2 <class 'int'>

2 <class 'int'>

3 <class 'int'>

3 <class 'int'>

4 <class 'int'>

4 <class 'int'>

5 <class 'int'>

5 <class 'int'>

6 <class 'int'>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[1, 2, 3, 4, 5] <class 'list'>

Press any key to continue . . .

Модель данных языка Python

Изменяемые объекты и неизменяемые объекты

Константный тип не изменяем

A = [['a'], ['b'], 'c', 'd', 'e']

for x in A:

print(x, type(x))

x += 'z'

print(x, type(x))

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print(A, type(A))

['a'] <class 'list'>

['a', 'z'] <class 'list'>

['b'] <class 'list'>

['b', 'z'] <class 'list'>

c <class 'str'>

cz <class 'str'>

d <class 'str'>

dz <class 'str'>

e <class 'str'>

ez <class 'str'>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[['a', 'z'], ['b', 'z'], 'c', 'd', 'e'] <class 'list'>

Press any key to continue . . .

Поэлементный доступ к списку

A = [1, 2, 3, 4, 5]

print(A, type(A))

for k in range(5):

print('Index', k, type(k))

print(A[k], type(A[k]))

A[k]=A[k]\*\*2

print(A[k], type(A[k]))

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print(A, type(A))

[1, 2, 3, 4, 5] <class 'list'>

Index 0 <class 'int'>

1 <class 'int'>

1 <class 'int'>

Index 1 <class 'int'>

2 <class 'int'>

4 <class 'int'>

Index 2 <class 'int'>

3 <class 'int'>

9 <class 'int'>

Index 3 <class 'int'>

4 <class 'int'>

16 <class 'int'>

Index 4 <class 'int'>

5 <class 'int'>

25 <class 'int'>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[1, 4, 9, 16, 25] <class 'list'>

Press any key to continue . . .

Записать массив из тысячи элементов

A = [0] \* 1000

print(A)

[0, 0, 0, 0, 0, 0,…, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Press any key to continue . . .

print("1, 2, 3, 4, 5, 0")

A = [0] \* 1000

top = 0

while True:

x = int(input())

if x == 0:

break

A[top] = x

top +=1

print("Arr", A)

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print("Top", top)

for k in range(4, -1, -1):

print(A[k])

1, 2, 3, 4, 5, 0

1

2

3

4

5

0

Arr [1, 2, 3, 4, 5, 0, 0, 0, …, 0, 0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Top 5

5

4

3

2

1

Press any key to continue . . .

top = 5

1

2

3

4

5

1000

Копирование массива

print("Размер массива")

N = int(input())

A = [0] \* N

B = [0] \* N

for k in range(N):

print("Ввод элемента массива", k)

A[k] = int(input())

for k in range(N):

B[k] = A[k]

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print("Массив А")

print(A)

print("Массив B")

print(B)

C = A

D = list(A)

C[0] = 'a'

print("Массив А")

print(A)

print("Массив B")

print(B)

print("Массив C")

print(C)

print("Массив D")

print(D)

Размер массива

4

Ввод элемента массива 0

4

Ввод элемента массива 1

3

Ввод элемента массива 2

2

Ввод элемента массива 3

1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Массив А

[4, 3, 2, 1]

Массив B

[4, 3, 2, 1]

Массив А

['a', 3, 2, 1]

Массив B

[4, 3, 2, 1]

Массив C

['a', 3, 2, 1]

Массив D

[4, 3, 2, 1]

Press any key to continue . . .

А

1

2

3

4

5

B

1

2

3

4

5

C

Алгоритм «Линейный поиск в массиве»

def arraySearch(A:list, N:int, x:int):

""" Осуществляет поиск числа x в массиве A

от O до N-1 индекса включительно.

Возвращает индекс элемента x в массиве A/

Или -1, если такого нет.

Если в массиве несколько одинаковых элементов,

равных x, то вернуть индекс первого элемента по счету.

"""

#pass

for k in range(N):

if A[k] == x:

return k

return -1

def testArraySearch():

print("\_\_\_\_\_\_test1\_\_\_\_\_\_\_\_")

A1 = [1, 2, 3, 4, 5]

m = arraySearch(A1, 5, 8)

if m == -1:

print("#test1-ok")

else:

print("#test1-fail")

print("\_\_\_\_\_\_test2\_\_\_\_\_\_\_\_")

A2 = [-1, -2, -3, -4, -5]

m = arraySearch(A2, 5, -3)

if m == 2:

print("#test2-ok")

else:

print("#test2-fail")

print("\_\_\_\_\_\_test3\_\_\_\_\_\_\_\_")

A3 = [10, 20, 30, 10, 10]

m = arraySearch(A3, 5, 10)

if m == 0:

print("#test3-ok")

else:

print("#test3-fail")

testArraySearch()

\_\_\_\_\_\_test1\_\_\_\_\_\_\_\_

#test1-ok

\_\_\_\_\_\_test2\_\_\_\_\_\_\_\_

#test2-ok

\_\_\_\_\_\_test3\_\_\_\_\_\_\_\_

#test3-ok

Press any key to continue . . .

Алгоритм «Обращение в массиве»

А

1

2

3

4

5

B

1

2

3

4

5

Что будет если делать на одном массиве?

def invertArray(A:list, N:int):

""" Обращение массива (поворот задом-наперед)

в рамках индексов от 0 до N-1

"""

#pass

B = [0]\*len(A)

for k in range(N):

#B[k] = A[N-1-k]

A[k] = A[N-1-k]

#return B

return A

def testInvertArray():

print("\_\_\_\_\_\_test1\_\_\_\_\_\_\_\_")

A1 = [1, 2, 3, 4, 5]

print(A1)

A1= invertArray(A1, 5)

print(A1)

if A1 == [5, 4, 3, 2, 1]:

print("#test1-ok")

else:

print("#test1-fail")

print("\_\_\_\_\_\_test2\_\_\_\_\_\_\_\_")

A2 = [0, 0, 0, 0, 0, 10]

print(A2)

A2 = invertArray(A2, 6)

print(A2)

if A2 == [10, 0, 0, 0, 0, 0]:

print("#test2-ok")

else:

print("#test2-fail")

testInvertArray()

\_\_\_\_\_\_test1\_\_\_\_\_\_\_\_

[1, 2, 3, 4, 5]

[5, 4, 3, 4, 5]

#test1-fail

\_\_\_\_\_\_test2\_\_\_\_\_\_\_\_

[0, 0, 0, 0, 0, 10]

[10, 0, 0, 0, 0, 10]

#test2-fail

Press any key to continue . . .

Будет перетирание элементов массива.

Инверсия элементов массива

Вариант1

def invertArray(A:list, N:int):

""" Обращение массива (поворот задом-наперед)

в рамках индексов от 0 до N-1

"""

#pass

B = [0]\*len(A)

for k in range(N):

B[k] = A[N-1-k]

#A[k] = A[N-1-k]

return B

#return A

def testInvertArray():

print("\_\_\_\_\_\_test1\_\_\_\_\_\_\_\_")

A1 = [1, 2, 3, 4, 5]

print(A1)

A1= invertArray(A1, 5)

print(A1)

if A1 == [5, 4, 3, 2, 1]:

print("#test1-ok")

else:

print("#test1-fail")

print("\_\_\_\_\_\_test2\_\_\_\_\_\_\_\_")

A2 = [0, 0, 0, 0, 0, 10]

print(A2)

A2 = invertArray(A2, 6)

print(A2)

if A2 == [10, 0, 0, 0, 0, 0]:

print("#test2-ok")

else:

print("#test2-fail")

testInvertArray()

\_\_\_\_\_\_test1\_\_\_\_\_\_\_\_

[1, 2, 3, 4, 5]

[5, 4, 3, 2, 1]

#test1-ok

\_\_\_\_\_\_test2\_\_\_\_\_\_\_\_

[0, 0, 0, 0, 0, 10]

[10, 0, 0, 0, 0, 0]

#test2-ok

Press any key to continue . . .

Вариант 2

def invertArray(A:list, N:int):

""" Обращение массива (поворот задом-наперед)

в рамках индексов от 0 до N-1

"""

#pass

for k in range(N//2):

A[k], A[N-1-k] = A[N-1-k], A[k]

return A

def testInvertArray():

print("\_\_\_\_\_\_test1\_\_\_\_\_\_\_\_")

A1 = [1, 2, 3, 4, 5]

print(A1)

A1= invertArray(A1, 5)

print(A1)

if A1 == [5, 4, 3, 2, 1]:

print("#test1-ok")

else:

print("#test1-fail")

print("\_\_\_\_\_\_test2\_\_\_\_\_\_\_\_")

A2 = [0, 0, 0, 0, 0, 10]

print(A2)

A2 = invertArray(A2, 6)

print(A2)

if A2 == [10, 0, 0, 0, 0, 0]:

print("#test2-ok")

else:

print("#test2-fail")

testInvertArray()

\_\_\_\_\_\_test1\_\_\_\_\_\_\_\_

[1, 2, 3, 4, 5]

[5, 4, 3, 2, 1]

#test1-ok

\_\_\_\_\_\_test2\_\_\_\_\_\_\_\_

[0, 0, 0, 0, 0, 10]

[10, 0, 0, 0, 0, 0]

#test2-ok

Press any key to continue . . .

Алгоритм «Циклический сдвиг в массиве»

0

1

2

3

4

tmp

Сдвиг влево, но движение вправо

0

1

2

3

4

tmp

Сдвиг вправо, но движение влево

м

A = [1, 2, 3, 4]

B = [5, 6, 7, 8]

lenA = len(A)

lenB = len(B)

print("влево")

print(A)

tmp = A[0]

for k in range(lenA-1):

A[k] = A[k + 1]

A[lenA - 1] = tmp

print(A)

print("вправо")

print(B)

tmp = B[lenB - 1]

for k in range(lenA -2, -1, -1):

B[k + 1] = B[k]

B[0] = tmp

print(B)

влево

[1, 2, 3, 4]

[2, 3, 4, 1]

вправо

[5, 6, 7, 8]

[8, 5, 6, 7]

Press any key to continue . . .

Алгоритм «Решето эратосфена»

print("Ввод размера массива")

N = int(input())

A = [True] \* N

A[0] = A[1] = False

for k in range(2, N):

if A[k]:

for m in range(2 \* k, N, k):

A[m] = False

for k in range(N):

print(k,'-',"простое" if A[k] else "составное")

Ввод размера массива

20

0 - составное

1 - составное

2 - простое

3 - простое

4 - составное

5 - простое

6 - составное

7 - простое

8 - составное

9 - составное

10 - составное

11 - простое

12 - составное

13 - простое

14 - составное

15 - составное

16 - составное

17 - простое

18 - составное

19 - простое

Press any key to continue . . .