

# ПОВТОРЕНИЕ

Какой метод вставляет в список **a** элемент **x** на позицию **i**?

`a.insert(i, x)`

Что делает метод `a.remove(x)`?

Убирает элемент **x** из списка **a**

# ПОВТОРЕНИЕ

Какой метод удаляет из списка `a` элемент на позиции `i` и возвращает его?

`a.pop(i)`

Какой элемент удаляется и возвращается при `a.pop()`?

Последний элемент списка



# ПОВТОРЕНИЕ

Какой метод возвращает индекс первого элемента со значением `x` из списка `a`?

`a.index(x)`

Как работает вызов метода `a.index(x, s, e)`?

Возвращает индекс первого значения `x` в интервале `[s:e]`

# ПОВТОРЕНИЕ

Что произойдет при вызове метода `index()`, если значения нет в списке?

Ошибка `ValueError`

Какой метод возвращает количество элементов со значением `x` из списка `a`?

`a.count(x)`



# ПОВТОРЕНИЕ

Какой метод сортирует список **a** в порядке возрастания?

`a.sort()`

Как отсортировать список **a** в порядке убывания?

`a.sort(reverse=True)`

# ЦЕЛЬ ЗАНЯТИЯ

- Вспомним, как работают **вложенные циклы**
- Подготовимся к работе с **двумерными списками**



# ВЛОЖЕННЫЕ ЦИКЛЫ

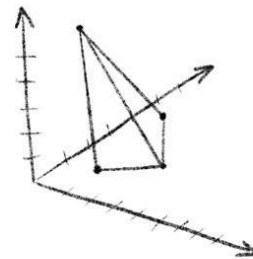
1D



2D



3D



# ЗАДАЧА

Вывести на экран **n** символов **\$**

```
n = int(input())
```

# вариант 1

```
for i in range(n):
```

```
    print("$", end="")
```

# вариант 2

```
print("$" * n)
```



# ЗАДАЧА

Вывести на экран **m** строк по **n** символов **\$**

```
m, n = map(int, input().split())
```

# вариант 1

```
for i in range(m):  
    print("$" * n)
```

# вариант 2

```
print((" $" * n + "\n") * m)
```

# ЗАДАЧА

Вывести на экран **m** строк по **n** символов **\$**

```
m, n = map(int, input().split())
```

**# вариант 3**

```
for i in range(m):
```

```
    for j in range(n):
```

```
        print("$", end="")
```

```
    print()
```



# ЗАДАЧА

Вывести на экран **m** строк по **n** символов **\$**

```
m, n = map(int, input().split())
```

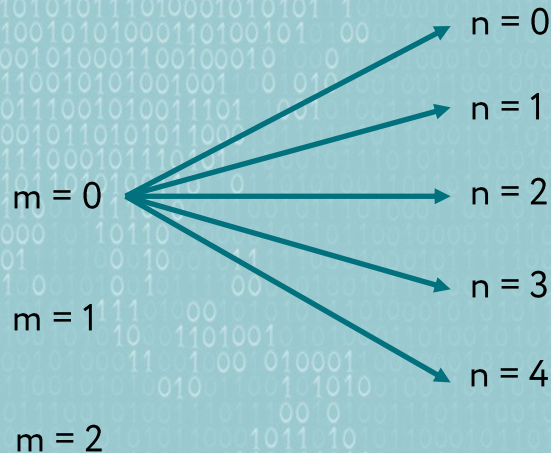
**# вариант 3**

```
for i in range(m):
```

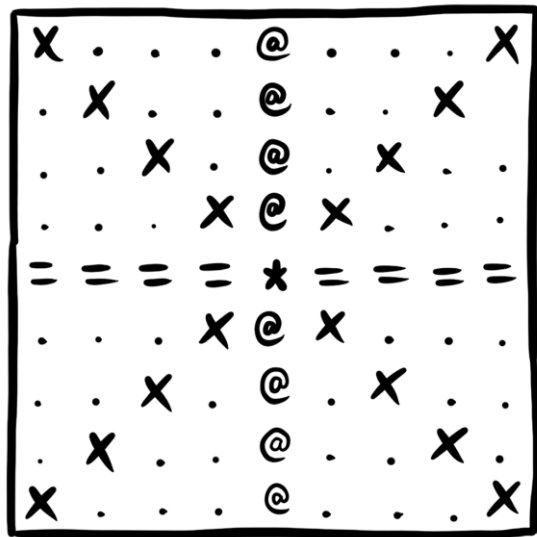
```
    for j in range(n):
```

```
        print("$", end="")
```

```
    print()
```



# РАЗБОР ЗАДАЧИ



Условие для \*

```
if x == n // 2 and y == n // 2:
```

Условие для =

```
if y == n // 2:
```

Условие для @

```
if x == n // 2:
```

Условие для x

```
if x == y or x = y - n - 1:
```

Условие для .

```
else:
```



# РАЗБОР ЗАДАЧИ

Основная программа:

```
n = int(input())  
for y in range(n):  
    for x in range(n):  
        print(pick_symbol(y, x, n), end="")  
    print()
```

# ВОПРОС

Что выведет код?

```
for i in range(5):  
    for j in range(7):  
        print(j, end=" ")  
    print()  
0 1 2 3 4 5 6  
0 1 2 3 4 5 6  
0 1 2 3 4 5 6  
0 1 2 3 4 5 6  
0 1 2 3 4 5 6
```



# ВОПРОС

Что выведет код?

```
for i in range(5):  
    for j in range(7):  
        print(i, end=" ")  
    print()  
    0 0 0 0 0 0 0  
    1 1 1 1 1 1 1  
    2 2 2 2 2 2 2  
    3 3 3 3 3 3 3  
    4 4 4 4 4 4 4
```

# ВОПРОС

Что выведет код?

```
c = 0
for i in range(5):
    for j in range(7):
        print(c, end=" ")
        c += 1
    print()
```

0 1 2 3 4 5 6  
7 8 9 10 11 12 13  
14 15 16 17 18 19 20  
21 22 23 24 25 26 27  
28 29 30 31 32 33 34



# ОСНОВНАЯ МЫСЛЬ

Для получения нужной последовательности чисел  
можно использовать...

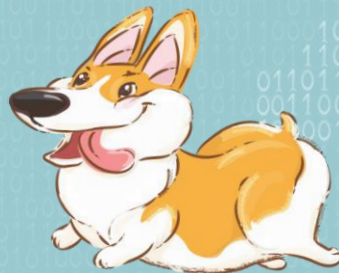
счетчик

цикла



вспомогательную

переменную



# ИТОГИ ЗАНЯТИЯ

- Вспомнили, как работать с **вложенными циклами**
- Подготовились к изучению **двумерных списков**