

C++

Урок 10

Разбор ДЗ

Разбор дз

1 задача:

Задача: Написать функцию подсчета кол-ва отрицательных значений в одномерном массиве.

Разбор дз

1 задача:

```
int count(int arr[100], int razm) //сама ф-ия
{
    int cnt = 0;
    for (int i = 0; i < razm; i++)
    {
        if (arr[i] < 0)
            cnt++;
    }
    return cnt;
}
```

Реализация возвратной функции

Разбор дз

1 задача:

```
int count(int[100], int); //объявляем прототип

int main()
{
    int arr[100] = {-1, 2, -4, 12, -47, -5, 55, -7};
    int size = 8;
    int cnt = count(arr, size); //использование ф-ии
    cout << "Отриц эл-тов: " << cnt << endl;
}

int count(int arr[100], int razm) //сама ф-ия
{
    int cnt = 0;
    for (int i = 0; i < razm; i++)
    {
        if (arr[i] < 0)
            cnt++;
    }
    return cnt;
}
```

Весь код

Разбор дз

2 задача:

Задача: Написать функцию, которая ищет максимальный элемент в одномерном массиве.

Разбор дз

2 задача:

```
int max(int arr[100], int size)
{
    int ind_max = 0;
    for (int i = 0; i < size; i++)
    {
        if (arr[i] > arr[ind_max])
            ind_max = i;
    }
    return arr[ind_max];
}
```

Реализация возвратной функции

Разбор дз

2 задача:

```
int max(int[100], int);

int main()
{
    int arr[100] = {-1, 2, -4, 12, -47, -5, 55, -7};
    int size = 8;
    cout << "Максимальный элемент: " << max(arr, size) << endl;
}

int max(int arr[100], int size)
{
    int ind_max = 0;
    for (int i = 0; i < size; i++)
    {
        if (arr[i] > arr[ind_max])
            ind_max = i;
    }
    return arr[ind_max];
}
```

Весь код

Разбор дз

3 задача:

Задача: Написать функцию, которая суммирует элементы ниже полочной диагонали в квадратном (двумерном) массиве.

Разбор дз

3 задача:

```
int sum(int arr[100][100], int size)
{
    int sum = 0;
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if (i + j > size - 1)
                sum += arr[i][j];
        }
    }
    return sum;
}
```

Реализация возвратной функции

Разбор дз

3 задача:

```
int sum(int[100][100], int);

int main()
{
    int arr[100][100] = {};
    int n;
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (i + j > n - 1)
                arr[i][j] = 1;
        }
    }
    cout << sum(arr, n);
}

int sum(int arr[100][100], int size)
{
    int sum = 0;
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if (i + j > size - 1)
                sum += arr[i][j];
        }
    }
    return sum;
}
```

Весь код

Разбор дз

4 задача:

Задача: Найдите площадь круга по введенному радиусу. Напишите ф-ию поиска площади круга. На вход ф-ии подается вещественная переменная.

Разбор дз

4 задача:

```
#include <iostream>

using namespace std;
const float PI = 3.14; //объявили константу PI

double pl(double); //объявили прототип

int main()
{
    double r;
    cin >> r;
    cout << "Площадь круга равна " << pl(r);
}

double pl(double rad)
{
    return (PI * rad * rad);
}
```

Разбор дз

5 задача:

Задача: Найти корни квадратного уравнения.
Пользователь вводит коэффициенты a , b , c функция выводит результат решения уравнения.

A hand-drawn blue oval frame with a slightly irregular, sketchy border, centered on the page. The word 'Повторение' is written inside this frame in a bold, black, sans-serif font.

Повторение

Что такое функция?

**Как хранится функция в памяти
компьютера ?**

Назовите виды функций

Что такое прототипы функций?



Какую проблему они решают?

Что такое перегрузка функций?

**Как можно еще назвать
перегруженную функцию?**

Перегрузка

Перегрузка — это возможность использовать одну и ту же функцию для разных типов данных.

Полиморфизм - способность функции обрабатывать данные разных типов.

перегруженная функция = полиморфная функция

Задача

Задача: Пользователь вводит три числа (три стороны). Напишите ф-ию нахождения площади. (учтите разные типы данных).

Рекурсия



Что такое рекурсия?



Рекурсия

Функция является **рекурсивной**, если оператор в теле функции вызывает функцию, содержащую данный оператор.

Вызов самой функции внутри нее.

Задача

Задача: По приезду в лилипутию жителей земли встречает конвертер валют. 1 рубль = 1.33 лилипутским дублонам.

Переведите рубли в дублоны и выведите эту сумму, если пользователь вводит отрицательно число, попросите ввести снова.

Рекурсия

Листинг:

```
#include <iostream>

using namespace std;

void run(int); //объявление прототипа

int main()
{
    int value;
    cout << "Введите изначальное кол-во денег: ";
    cin >> value;
    run(value);
}

void run(int num)
{
    if (num < 0)
    {
        cout << "Введите положительное число для подсчета: " << endl;
        int n;
        cin >> n;
        run(n); //вызов самой ф-ии
    }
    else
    {
        cout << "В лилипутии будет: " << num * 1.33 << " дублонов" << endl;
    }
}
```

Рекурсия

Задача:

Выведите факториал до введенного числа.

Рекурсия

Листинг:

```
#include <iostream>
using namespace std;

unsigned int factorial(unsigned int); // прототип рекурсивной функции

int main(int argc, char *argv[])
{
    int n; // локальная переменная для передачи введенного числа с клавиатуры
    cout << "Enter n!: ";
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        cout << i << "!" << "=" << factorial(i) << endl; // вызов рекурсивной функции
    }
    return 0;
}

unsigned int factorial(unsigned int f) // рекурсивная функция для нахождения n!
{
    unsigned long int result;
    if (f == 1 || f == 0) // 1!=1 и 0!=1
        return 1;
    result = f * factorial(f - 1); // вызов самой себя
    return result;
}
```

A hand-drawn blue oval frame with a double-line border, centered on the page. The word 'Указатели' is written inside this frame in a bold, black, sans-serif font.

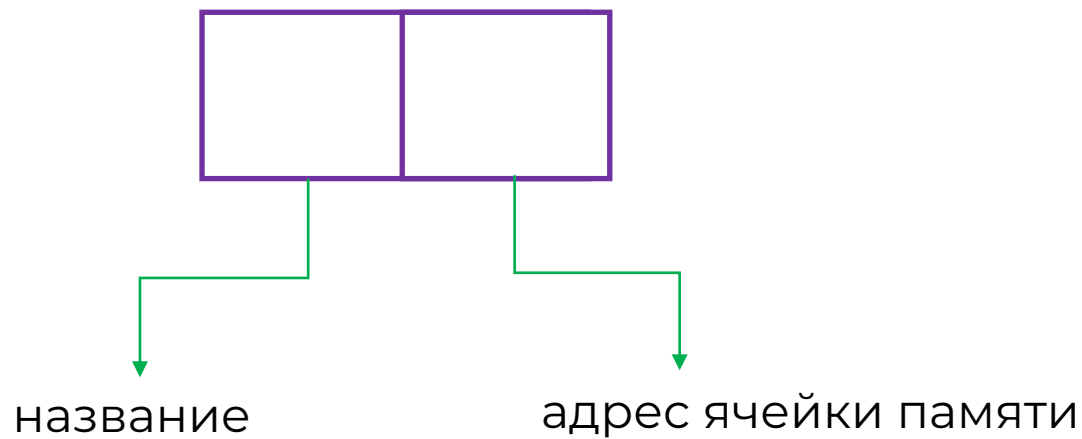
Указатели



Что такое указатель?

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти.



**В какой СС информация
хранится в компьютере?**

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Какой размер указателя в 32 битных системах?

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Какой размер указателя в 32 битных системах? **4 байт**

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Какой размер указателя в 64 битных системах?

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

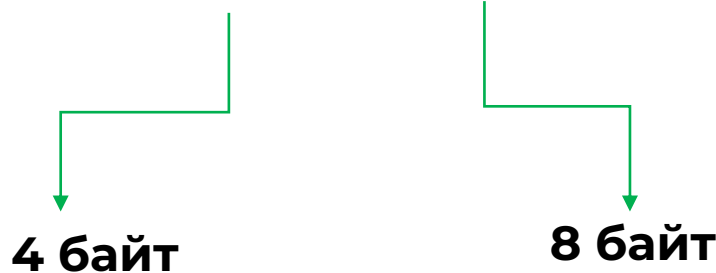
Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Какой размер указателя в 64 битных системах? **8 байт**

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)



ВСЕГДА!