

# Семинар #1: Основы C. Решения классных задач.

## Часть 2: Основы C

### Hello World!

Простейшая программа на языке C выглядит следующим образом:

```
#include <stdio.h>
int main() {
    printf("Hello world!");
}
```

Эта программа печатает на экран строку "Hello world!".

- `#include <stdio.h>` - включаем библиотеку `stdio` (standard input/output), которая содержит `printf`.
- `int main() { ... }` - основная функция программы, с неё начинается исполнение любой программы.
- `printf("Hello world!");` - печатаем на экран.

### Задание на основы printf

1. Скомпилируйте программу, используя `gcc` и запустите.
2. В строке функции `printf()` можно использовать некоторые специальные символы `\n`, `\t` и `\b`. Добавьте эти символы в строку функции `printf` (в произвольное место) и выясните, что они делают.
3. Напишите программу, которая будет выводить на экран:

```
First
    Second
        Third
```

Используйте 1 вызов функции `printf`. Для отступов используйте пробелы или знаки табуляции (`\t`).

```
#include <stdio.h>
int main() {
    printf("First\n\tSecond\n\t\tThird\n");
}
```

### Целочисленные переменные int:

В переменных `int` можно хранить целые числа от  $-2^{31}$  до  $2^{31} - 1$ . ( $2^{31}$  примерно равно двум миллиардам)

```
#include <stdio.h>
int main() {
    int a;
    int b = 5;
    a = 3;
    int res = a * b + (b / a);
}
```

```
    printf("Result = %i\n", res);  
}
```

- `int a` - Объявляем, что у нас есть переменная `a`, которая будет хранить целые числа.
- `int b = 5` - Объявляем, что есть переменная `b`, которая будет хранить целые числа и присваиваем ей 5.
- `a = 3` - Присваиваем переменной `a` число 3.
- `res = a * b + (b / a)` - Сохраняем в переменной `res` результат вычислений.
- `printf("Result = %i \n ", res)` Печатаем, за место спецификатора `%i` (сокращение от `int`) подставится значение переменной.

## Задание на целочисленные переменные:

1. Создайте переменные `a` и `b` и присвойте им значения `a = 26`, `a b = 7`. Затем:

- Напечатайте на экран число `a`. Вот так:

```
#include <stdio.h>  
int main() {  
    int a = 26, b = 7;  
    printf("%i\n");  
}
```

- Напечатайте на экран 2 числа `a` и `b`, разделённые пробелом. Вот так:

```
#include <stdio.h>  
int main() {  
    int a = 26, b = 7;  
    printf("%i %i\n", a, b);  
}
```

- Напечатайте на экран 2 числа `a` и `b` в следующем формате (26, 7).

```
#include <stdio.h>  
int main() {  
    int a = 26, b = 7;  
    printf("(%i, %i)\n", a, b);  
}
```

- Напечатайте на экран 2 числа `a` и `b` в следующем формате [26:7]).

```
#include <stdio.h>  
int main() {  
    int a = 26, b = 7;  
    printf("[%i:%i]\n", a, b);  
}
```

- Напечатайте на экран 2 числа **a** и **b** в следующем формате (**A = 26 and B = 7**).

```
#include <stdio.h>
int main() {
    int a = 26, b = 7;
    printf("A = %i and B = %i\n", a, b);
}
```

- Напечатайте на экран сумму чисел **a** и **b**.

```
#include <stdio.h>
int main() {
    int a = 26, b = 7;
    printf("%i\n", a + b);
}
```

- Напечатайте на экран произведение чисел **a** и **b**.

```
#include <stdio.h>
int main() {
    int a = 26, b = 7;
    printf("%i\n", a * b);
}
```

- Напечатайте на экран результат целочисленного деления **a** на **b**. Используйте оператор: **a / b**. В результате этой операции должно получиться целое число 3 (так как это операция деления нацело).

```
#include <stdio.h>
int main() {
    int a = 26, b = 7;
    printf("%i\n", a / b);
}
```

- Напечатайте на экран остаток деления **a** на **b**. Используйте оператор: **a % b**. В результате этой операции должно получиться целое число 5.

```
#include <stdio.h>
int main() {
    int a = 26, b = 7;
    printf("%i\n", a % b);
}
```

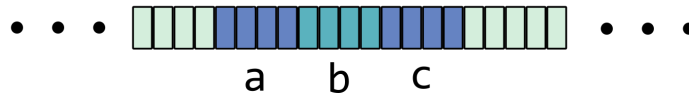
2. Пусть `a = 2147483647` (максимальное возможное значение для `int`). Напечатайте значение `a + 1` и `2a`.

```
#include <stdio.h>
int main() {
    int a = 2147483647;
    printf("%i\n%i\n", a + 1, 2 * a);
}
```

## Адрес и размер переменной:

- 1 бит - минимальная единица измерения памяти. В 1 бите может храниться либо 0 либо 1.
- Вся память делится на ячейки, размером в 8 бит = 1 байт.
- Все эти ячейки занумерованы, номер ячейки называется адресом.
- Все переменные содержатся в памяти. Адрес переменной - это адрес первого байта переменной.
- Чтобы найти адрес переменной, нужно перед ней поставить `&`, например, `&a`
- Чтобы найти размер переменной в байтах: `sizeof(a)`
- Например, переменная типа `int` имеет размер 4 байта = 32 бита. Значит в ней может храниться максимум  $2^{32}$  значений.

`int a, b, c;`



## Задание:

1. Напечатать размер переменной типа `int` в байтах. Для этого используйте оператор `sizeof`:

```
#include <stdio.h>
int main() {
    int a = 26;
    printf("%i\n", sizeof(a));
}
```

2. Напечатать адреса переменных типа `int`. Для этого используйте оператор `&`. Адреса памяти обычно хранятся не в переменных типа `int`, а в больших по размеру переменных. Поэтому для их печати нужно использовать не `%i`, а `%lli` или `%p`:

```
#include <stdio.h>
int main() {
    int a = 26, b = 7;
    printf("%lli %lli\n", &a, &b);
}
```

Убедитесь, что переменные `a` и `b` лежат в памяти вплотную друг к другу.

## Считывание переменных - scanf:

Считывание переменных из терминала осуществляется с помощью функции `scanf` из библиотеки `stdio`. В отличие от `printf`, в `scanf` нужно передавать не саму переменную, а её адрес. Это естественно, так как `scanf` должен записать считываемое значение в соответствующие ячейки памяти.

Пример программы, которая считывает переменные `a` и `b` и печатает на экран их произведение:

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%i", &a); // <-- не забудьте тут амперсанд &
    scanf("%i", &b); // <-- не забудьте тут амперсанд &
    printf("Result = %i\n", a * b);
}
```

*Примечание:* При задании формата в `scanf` не нужно ставить пробелы и символы переноса строки. Т.е. не нужно писать так:

```
scanf("%i\n", &a); // будет ожидать ввода ещё одного числа
```

### Задание на считывание:

1. Написать программу, которая считывает целое число и печатает на экран квадрат этого числа.

```
#include <stdio.h>
int main() {
    int a;
    scanf("%i", &a);
    printf("%i\n", a * a);
}
```

2. Считать 2 целых числа и напечатать результат целочисленного деления первого на второе.

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%i%i", &a, &b);
    printf("%i\n", a / b);
}
```

3. Считать 2 целых числа и напечатать остаток деления первого на второе.

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%i%i", &a, &b);
    printf("%i\n", a % b);
}
```

4. Считать целое число и напечатать его последнюю цифру. Используйте оператор остатка.

```
#include <stdio.h>
int main() {
    int a;
    scanf("%i", &a);
    printf("%i\n", a % 10);
}
```

5. На вход подаётся прошедшее время в формате hh:mm, например, 05:14. Нужно напечатать, общее количество минут (314). Создайте 2 переменные `hours` и `minutes` и считать значения этих переменных с помощью `scanf`. Вот так:

```
#include <stdio.h>
int main() {
    int hours, minutes;
    scanf("%i:%i", &hours, &minutes);
    printf("%i\n", 60 * hours + minutes);
}
```

## Операторы инкремента:

Для удобства в языке C введены следующие операторы:

`+=`    `-=`    `*=`    `/=`    `++`    `--`    и другие

Например оператор присваивания сложения `+=` увеличивает левый аргумент на величину правого. Оператор `++` увеличивает значение аргумента на 1.

```
#include <stdio.h>
int main() {
    int a = 100;
    a = a + 5; // увеличиваем a на 5
    a += 5;    // увеличиваем a на 5 ( то же самое )

    a++;       // увеличиваем a на 1
    ++a;       // увеличиваем a на 1
}
```

Чему будет равно значение переменной `a` после выполнение данного кода?

## Логические операторы:

==	равно		
!=	не равно		
>	больше	&&	логическое И
>=	больше и равно		логическое ИЛИ
<	меньше	!	логическое НЕ
<=	меньше и равно		

Пример программы, которая считывает возраст человека и печатает **Yes**, если возраст больше или равен 18:

```
#include <stdio.h>
int main() {
    int age;
    scanf("%i", &age);
    if (age >= 18) {
        printf("Yes\n");
    }
}
```

Пример программы, которая считывает число **n** и печатает **Yes**, если число двузначно и **No** иначе:

```
#include <stdio.h>
int main() {
    int n;
    scanf("%i", &n);
    if (n >= 10 && n < 100) {
        printf("Yes\n");
    }
    else {
        printf("No\n");
    }
}
```

Пример программы, которая принимает на вход число и печатает **Positive**, если число положительное, **Negative**, если число отрицательное и **Zero**, если число равно нулю:

```
#include <stdio.h>
int main() {
    int n;
    scanf("%i", &n);
    if (n > 0) {
        printf("Positive\n");
    }
    else if (n == 0){
        printf("Zero\n");
    }
    else {
        printf("Negative\n");
    }
}
```

## Задание на логические операторы:

1. Написать программу, которая считывает число и печатает **Yes**, если число равно 42 и **No** иначе. Обратите внимание, что для сравнения чисел нужно использовать оператор `==` ("два равна").

```
#include <stdio.h>
int main() {
    int a;
    scanf("%i", &a);
    if (a == 42)
        printf("Yes\n");
    else
        printf("No\n");
}
```

2. Написать программу, которая принимает на вход число и печатает **Yes**, если число принадлежит множеству  $(-\infty, -12] \cup (97, +\infty)$  и **No** иначе.

```
#include <stdio.h>
int main() {
    int a;
    scanf("%i", &a);
    if (a <= -12 || a > 97)
        printf("Yes\n");
    else
        printf("No\n");
}
```

3. Написать программу, которая принимает на вход число и печатает **Even**, если число четное и **Odd**, если число нечетное. Подсказка: число чётное, если остаток от деления на 2 равен 0.

```
#include <stdio.h>
int main() {
    int a;
    scanf("%i", &a);
    if (a % 2)
        printf("Odd\n");
    else
        printf("Even\n");
}
```



4. Написать программу, которая принимает на вход два числа и печатает **First**, если первое число больше второго, **Second**, если второе больше первого и **Equal**, если числа равны.

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%i%i", &a, &b);
    if (a > b)
        printf("First\n");
    else if (a < b)
        printf("Second\n");
    else
        printf("Equal\n");
}
```

5. Написать программу, которая принимает на вход два числа и печатает большее из этих двух чисел.

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%i%i", &a, &b);
    if (a > b)
        printf("%i\n", a);
    else
        printf("%i\n", b);
}
```

6. Написать программу, которая принимает на вход три числа и печатает **Unique**, если все числа различны и **Not Unique**, если хотя бы 2 числа равны.

```
#include <stdio.h>
int main() {
    int a, b, c;
    scanf("%i%i%i", &a, &b, &c);
    if (a != b && b != c && c != a)
        printf("Unique\n");
    else
        printf("Not Unique\n");
}
```

## Цикл while:

### Пример:

Пример программы, которая печатает числа от 0 до 9, разделённые пробелом:

```
#include <stdio.h>
int main() {
    int i = 0;
    while (i < 10) {
        printf("%i ", i);
        i += 1;
    }
}
```

### Задачи:

Измените программу выше так чтобы:

1. Программа печатала числа от 0 до 20

```
#include <stdio.h>
int main() {
    int i = 0;
    while (i <= 20) {
        printf("%i ", i);
        i += 1;
    }
}
```

2. Программа печатала числа от 5 до 15

```
#include <stdio.h>
int main() {
    int i = 5;
    while (i <= 15) {
        printf("%i ", i);
        i += 1;
    }
}
```

3. Программа печатала числа от 5 до 15, разделённые не пробелом, а запятой:

5,6,7,8,9,10,11,12,13,14,15,

```
#include <stdio.h>
int main() {
    int i = 5;
    while (i <= 15) {
        printf("%i,", i);
        i += 1;
    }
}
```

4. Программа печатала числа, разделённые символом +

```
#include <stdio.h>
int main() {
    int i = 5;
    while (i <= 15) {
        printf("%i+", i);
        i += 1;
    }
}
```

5. Программа печатала числа, разделённые символом переноса строки \n. (каждое число в новой строке)

```
#include <stdio.h>
int main() {
    int i = 5;
    while (i <= 15) {
        printf("%i\n", i);
        i += 1;
    }
}
```

6. Программа печатала квадраты этих чисел

```
#include <stdio.h>
int main() {
    int i = 5;
    while (i <= 15) {
        printf("%i ", i * i);
        i += 1;
    }
}
```

7. Программа печатала только чётные числа от 0 до 100

```
#include <stdio.h>
int main() {
    int i = 0;
    while (i <= 100) {
        printf("%i ", i);
        i += 2;
    }
}
```

8. Программа печатала только числа, делящиеся на 7 (от 0 до 100)

```
#include <stdio.h>
int main() {
    int i = 0;
    while (i <= 100) {
        printf("%i ", i);
        i += 7;
    }
}
```

9. Программа должна считывать число **n** и печатать все числа от 0 до **n** через пробел.

```
#include <stdio.h>
int main() {
    int n;
    scanf("%i", &n);
    int i = 0;
    while (i <= n) {
        printf("%i ", i);
        i += 1;
    }
}
```

10. Программа должна считывать число **n** и печатать все квадраты чисел от 0 до **n** через перенос строки.

```
#include <stdio.h>
int main() {
    int n;
    scanf("%i", &n);
    int i = 0;
    while (i <= n) {
        printf("%i\n", i * i);
        i += 1;
    }
}
```

11. Программа должна считывать число **n** и для каждого числа из диапазона от 0 до **n** программа должна печатать **Foo**, если число делится на 3 и **Bar**, если число делится на 5.

```
#include <stdio.h>
int main() {
    int n;
    scanf("%i", &n);
    int i = 0;
    while (i <= n) {
        printf("%i: ", i);
        if (i % 3 == 0)
            printf("Foo");
        if (i % 5 == 0)
            printf("Bar");
        printf("\n");
        i += 1;
    }
}
```

12. Программа должна считывать числа **a**, **b**, **c**, и печатать все числа, делящиеся на **c** на отрезке от **a** до **b** через пробел.

```
#include <stdio.h>
int main() {
    int a, b, c;
    scanf("%i%i%i", &a, &b, &c);
    int i = a - 1 + c - (a - 1) % c;
    while (i <= b) {
        printf("%i ", i);
        i += c;
    }
}
```

### Пример:

Пример программы, которая вычисляет сумму чисел от 1 до **n**.

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    int sum = 0;
    int i = 1;
    while (i <= n) {
        sum += i;
        i += 1;
    }
    printf("%d\n", sum);
}
```

### Задачи:

Измените программу выше так чтобы:

1. Программа находила произведение всех чисел от 1 до n.

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    int fact = 1;
    int i = 1;
    while (i <= n) {
        fact *= i;
        i += 1;
    }
    printf("%d\n", fact);
}
```

2. Программа находила сумму всех нечётных чисел от 1 до n.

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    int sum = 0;
    int i = 1;
    while (i <= n) {
        sum += i;
        i += 2;
    }
    printf("%d\n", sum);
}
```

3. Программа находила сумму квадратов всех чисел от 1 до n.

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    int sum = 0;
    int i = 1;
    while (i <= n) {
        sum += i * i;
        i += 1;
    }
    printf("%d\n", sum);
}
```

4. Программа вычисляла следующее выражение:

$$1^2 - 2^2 + 3^2 - 4^2 + 5^2 - 6^2 + \dots + (-1)^{n+1}n^2$$

```

#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    int sum = 0;
    int i = 1;
    while (i <= n) {
        if (i % 2)
            sum += i * i;
        else
            sum -= i * i;
        i += 1;
    }
    printf("%d\n", sum);
}

```

### Пример:

Пример программы, которая считывает числа последовательно и печатает квадраты этих чисел. Если попадётся отрицательное число, то программа закончится.

```

#include <stdio.h>
int main() {
    while (1) {
        int a;
        scanf("%i", &a);
        if (a < 0) {
            break;
        }
        printf("%i\n", a * a);
    }
}

```

### Задачи:

Измените программу выше так чтобы:

1. Программа выводила кубы чисел, пока не встретит отрицательное число.

```

#include <stdio.h>
int main() {
    while (1) {
        int a;
        scanf("%i", &a);
        if (a < 0) {
            break;
        }
        printf("%i\n", a * a * a);
    }
}

```

2. Программа печатала **Odd**, если число нечётное и **Even**, если число чётное, пока не встретит отрицательное число

```
#include <stdio.h>
int main() {
    while (1) {
        int a;
        scanf("%i", &a);
        if (a < 0) {
            break;
        }
        if (a % 2)
            printf("Odd\n");
        else
            printf("Even\n");
    }
}
```

3. Для каждого введённого числа **a** программа должна печатать последовательность чисел от 1 до **a** через пробел. Для этого вам нужно использовать ещё один цикл **while** внутри цикла **while**.

```
#include <stdio.h>
int main() {
    while (1) {
        int a;
        scanf("%i", &a);
        if (a < 0) {
            break;
        }
        int i = 1;
        while (i <= a) {
            printf("%i ", i);
            i++;
        }
        printf("\n");
    }
}
```



4. Для каждого введённого числа **a** программа должна печатать сумму последовательности чисел от 1 до **a**.

```
#include <stdio.h>
int main() {
    while (1) {
        int a;
        scanf("%i", &a);
        if (a < 0) {
            break;
        }
        int sum = 0;
        int i = 1;
        while (i <= a) {
            sum += i;
            i++;
        }
        printf("%i\n", sum);
    }
}
```

## Цикл for:

### Пример:

Пример программы с циклом **for**, которая печатает числа от 0 до 9.

```
#include <stdio.h>
int main() {
    for (int i = 0; i < 10; ++i) {
        printf("%i ", i);
    }
}
```

Для компиляции этой программы возможно потребуется указать опцию компилятора **-std=c11**. Вот так:

```
gcc -o prog -std=c11 <файл исходного кода>
```

### Задачи:

Измените программу выше так чтобы:

1. Программа печатала числа от 0 до 20

```
#include <stdio.h>
int main() {
    for (int i = 0; i < 21; ++i) {
        printf("%i ", i);
    }
}
```

2. Программа печатала числа от 5 до 15

```
#include <stdio.h>
int main() {
    for (int i = 5; i <= 15; ++i) {
        printf("%i ", i);
    }
}
```

3. Программа печатала числа от 5 до 15, разделённые не пробелом, а запятой:

5,6,7,8,9,10,11,12,13,14,15,

```
#include <stdio.h>
int main() {
    for (int i = 5; i <= 15; ++i) {
        printf("%i,", i);
    }
}
```

4. Программа печатала числа, разделённые символом переноса строки `\n`. (каждое число в новой строке)

```
#include <stdio.h>
int main() {
    for (int i = 5; i <= 15; ++i) {
        printf("%i\n", i);
    }
}
```

5. Программа печатала только числа, делящиеся на 7 (от 0 до 100)

```
#include <stdio.h>
int main() {
    for (int i = 0; i <= 100; i += 7) {
        printf("%i ", i);
    }
}
```

6. Программа считывала `n` и печатала все числа от 1 до `n` и их квадраты в следующем виде:

1 1  
2 4  
3 9  
4 16  
...

```

#include <stdio.h>
int main() {
    int n;
    scanf("%i", &n);
    for (int i = 1; i <= n; i += 1) {
        printf("%i %i\n", i, i * i);
    }
}

```

7. Программа считывала **n** и печатала **n** символов звёздочка \*. Например, если ввести 7, то программа должна напечатать \*\*\*\*\*.

```

#include <stdio.h>
int main() {
    int n;
    scanf("%i", &n);
    for (int i = 0; i < n; i++) {
        printf("*");
    }
    printf("\n");
}

```

### Пример:

Что напечатает данная программа?

```

#include <stdio.h>
int main() {
    int n;
    scanf("%i", &n);
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            printf("(%i %i) ", i, j);
        }
        printf("\n");
    }
}

```

### Задачи:

Измените программу выше так чтобы:

1. Программа печатала таблицу умножения.

```
#include <stdio.h>
int main() {
    for (int i = 1; i < 10; i++) {
        for (int j = 1; j < 10; j++) {
            printf("%2i ", i * j);
        }
        printf("\n");
    }
}
```

2. Программа считывала **n** и печатала квадрат из звёздочек размером **nхn**.  
*Это одна из домашних задач*