

Препроцессор

Практическое занятие

#define

В массивах:

```
#include <stdio.h>
#define N 4
int main(void)
{
    int numbers[N] = {1, 2, 3, 4};
    for(int i = 0; i < N; ++i)
    {
        printf("%d ", numbers[i]);
    }
    return 0;
}
```

Вложенные макросы:

```
#define FALSE 0
#define TRUE !FALSE
int main(){
    if (y == FALSE){

    }
    if (z == TRUE){}

}
```

Макросы с параметрами:

```
#define PRINT_VALUE (number) printf("%d", value_##number);

int value_one = 10, value_two = 20;
PRINT_VALUE(one) // -> 10
PRINT_VALUE(two) // -> 20
```

Предопределенные макросы:

```
#include <stdio.h>

#define log_info(M, ...) printf("[INFO] (%s:%d) " M "\n", \
    __FILE__, __LINE__, ##__VA_ARGS__)

int main(int argc, char *argv[])
{
    log_info("Start of program");
    int var = 10;

    log_info("my variable is %d", var);

    log_info("End of program");
    return 0;
}
```

Перенос тела макроса:

```
#define SWAP(type,a, b) ({ \
    typeof(a) tmp = a; \
    a = b; \
    b = tmp; })
```

```
SWAP(int, num1, num2);
SWAP(float, num1, num2);
```

```
#include <stdio.h>
```

```
#define HELLO printf("Hello World! \n")
#define FOR for(int i=0; i<4; i++)
```

```
int main(void)
{
    FOR HELLO;
    return 0;
}
```

Условная компиляция

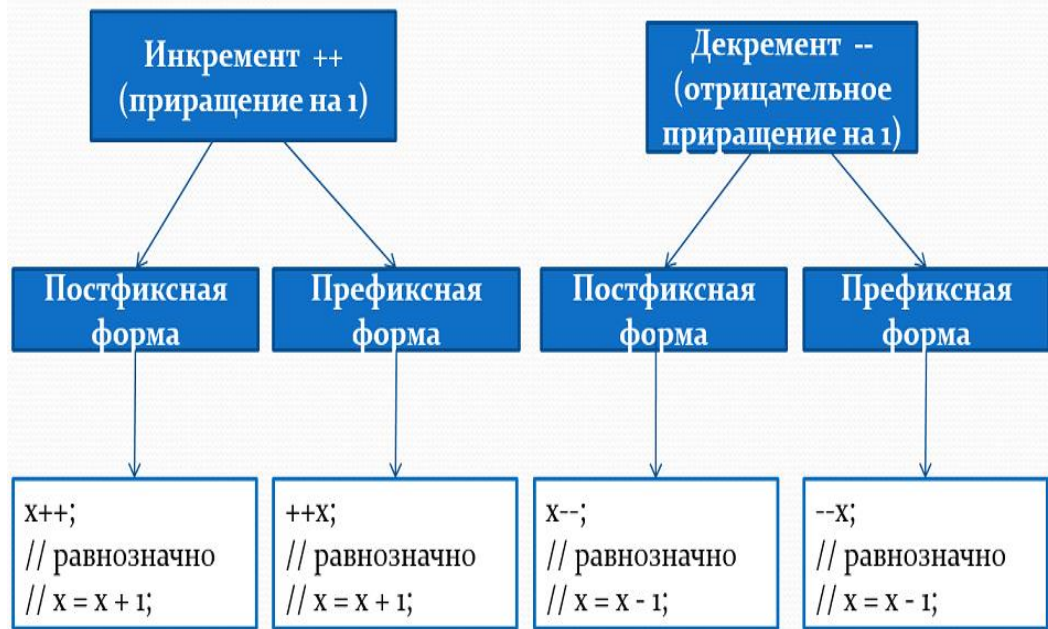
```
#ifndef имя_макроса  
    последовательность операторов  
#endif
```

```
#ifndef _MY_LIB  
#define _MY_LIB  
// код  
#endif
```

#pragma once

```
#include <stdio.h>  
  
#define TED 10  
  
int main(void)  
{  
    #ifdef TED  
        printf("Привет, Тед\n");  
    #else  
        printf("Привет, кто-нибудь\n");  
    #endif  
    #ifndef RALPH  
        printf("А RALPH не определен, т.ч. Ральфу не повезло.\n");  
    #endif  
  
    return 0;  
}
```

Использование инкремента/декремента



Можно ли применять операции инкремента и декремента к переменным символьного типа (char)?

```
char c;  
c = 'x';  
c++; // c = ?  
  
c = '6';  
--c; // c = ?
```

Можно ли применять операции инкремента (++) и декремента (—) к типу bool?

В Visual C++ только операцию инкремента (++) можно применять к типу bool.

Пример использования операции инкремента над переменной типа bool

```
bool b;  
b = false;  
b++; // b = true  
b++; // b = true
```

Можно ли применять операции инкремента и декремента к вещественным типам (float, double, long double)?

Продолжаем практиковаться

```
#include <iostream>
using namespace std; // для чего это здесь!

int main()
{

    int var = 1;
    int a = 0;

    // расставим сверху приоритеты,
    // найдем значение выражения
    a = ++var + 1 + ++var * 2;

    cout << "a = " << a << endl;

    return 0;
}
```

1. Даны натуральные числа от 35 до 87. Найти и напечатать те из них, которые при делении на 7 дают остаток 1, 2 или 5.

2. Найти сумму чисел от 100 до 200 кратных 17.

3. С помощью оператора **while** напишите программу определения суммы всех нечетных чисел в диапазоне от 1 до 99 включительно.

4. Найти произведение двузначных нечетных чисел кратных 13.

Домашняя работа

- Напечатать минимальное число, большее 200, которое нацело делится на 17.
- Известен факториал числа. Найти это число (факториал числа n равен $1 \cdot 2 \cdot \dots \cdot n$).
- Задача посложней. Нарисовать равнобедренный треугольник из символов `^`. Высоту выбирает пользователь. Например: высота = 5, на экране



- И пару заданий на макросы:
 - Напишите стандартный макрос `MIN` – поиск минимального значения из 2-х чисел. Для определенности пусть числа типа `int`.
 - Используя директиву `#define`, напишите именованную константу, возвращающую число секунд в году. Високосным годом – пренебречь.