

*Комментарий.* Во всех заданиях, где приведен некий код, подразумевается, что выше включены все необходимые для успешной компиляции `#include`, `using namespace`, etc. То есть можно считать, что каждому листингу предшествует:

```
#include "stdafx.h" //for MSVS only!
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <ctime>
#include <cstring>
#define _USE_MATH_DEFINES
#include <cmath>
#include <iostream>
#include <fstream>
```

```
using namespace std;
```

Если в листинге отсутствует `int main()`, подразумевается, что весь листинг содержится в стандартной функции `int main()`.

**Задание 1.** Сколько байт занимает значение типа `int`? А сколько — `(int*)`? Сколько `(int&)`?

**Задание 2.** Напишите на языке C++ с использованием `iostream` программу, которая будет считывать с клавиатуры фамилию, имя и год рождения пользователя и выводить приветствие вида "Здравствуйте, %username%! Вы родились в %year% году".

**Задание 3.** Что напечатает программа?

```
int acc = 2;
int foo(int i){
    return acc++;
}

int main(void)
{
    cout << foo(1) << foo(2) << foo(3);
    cout << (foo(1) << foo(2) << foo(3));
    return 0;
}
```

**Задание 4.** Напишите на C++ программу с использованием `printf`, которая будет выводить на экран таблицу синусов и косинусов для всех углов от  $0^\circ$  до  $90^\circ$  с шагом  $5^\circ$ .

**Задание 5.** Перед вами фрагмент кода на C++. В чем состоит возможная ошибка в этом коде?

```
int a, b;
scanf("%d, %d", &a, &b);
if(a = b){
    ++a;
    --b;
}
```

**Задание 6.** Перед вами фрагмент кода на C++. Какое значение он напечатает? В чем состоит возможная ошибка?

```
double d = 1/4;
printf("%f", d);
```

**Задание 7.** Перед вами программа на языке C++. Что она напечатает? Почему?

```
int main(void){
    printf("%s\n", (1<2<3<4) ? "true" : "false");
}
```

```

    printf("%s\n", (4>3>2>1) ? "true" : "false");
    return 0;
}

```

**Задание 8.** Перед вами функция на C++. Что она делает? Почему эта функция написана неоптимально? Перепишите ее оптимально.

```

void foo(char* sz){
    for(int i = 0; i < strlen(sz); ++i)
        if(sz[i]=='a')
            sz[i]='b';
}

```

**Задание 9.** Какую ошибку допустил программист при работе с памятью?

```

int *p = new int [10];
for(int i = 0; i < 10; ++i)
    p[i] = i*i;
delete p;

```

**Задание 10.** Какую ошибку допустил программист при работе с памятью?

```

int *p = new int [10];
for(int i = 0; i < 10; ++i)
    p[i] = i*i;
delete [] p;
delete p;

```

**Задание 11.** Что напечатает программа?

```

printf("WTF??!\n");

```

**Задание 12.** Что будет напечатано?

```

class A{
public:
    virtual int method1(void){
        printf("Method 1 in A\n");
    }
    void method2(void){
        printf("Method 2 in A\n");
    }
};

class B : public A{
public:
    virtual int method1(void){
        printf("Method 1 in B\n");
    }
    virtual void method2(void){
        printf("Method 2 in B\n");
    }
};

int main(void){
    A a;
    B b;
    A* pa = &b;
    A b2a = b;
    A& b2a1 = b
}

```

```

        a.method1();
        a.method2();
        b.method1();
        b.method2();
        pa->method1();
        pa->method2();
        b2a.method1();
        b2a.method2();
        b2a1.method1();
        b2a1.method2();
        return 0;
}

```

**Задание 13.** Что будет напечатано? Как можно иначе обратиться к переменной qua?

```

class StaticDemo{
public:
    static int qua;
    StaticDemo(){
        ++qua;
    }
    ~StaticDemo(){
        --qua;
    }
};

int StaticDemo::qua = 0;

int main(void){
    StaticDemo A, B, C;
    {
        StaticDemo E, F;
        printf("qua = %d\n", F.qua);
    }
    printf("qua = %d\n", A.qua);
}

```

**Задание 14.** Допишите данный класс, перегрузив операторы + и \* естественным образом для комплексных чисел.

```

class Complex{
public:
    double re, im;
    Complex(double re, double im = 0) : re(re), im(im){
        ;
    }
    double norm(void){
        return sqrt(re*re + im*im);
    }

    //You code here
}

```

**Задание 15.** Задание придумано К. Д. Усевичем. Вопрос, как обычно, что будет напечатано?

```

class Parent {
public:
    double x;
    Parent() { cout << "P"; }
    Parent(const Parent& P ) {}
    ~Parent() { cout << "DP"; }
}

```

```

};

class Child : public Parent {
public:
    Child() { cout << "C"; }
    ~Child() { cout << "DC"; }
};

void main() {
    Parent *x = new Child();
    Parent y = Child();

    delete x;
}

```

**Задание 16.** Какую ошибку допустил программист, реализуя класс-контейнер для строк?

```

class Container{
private:
    char *pchar;
public:
    Container(const char* sz = ""){
        pchar = new char[strlen(sz) + 1];
        strcpy(pchar, sz);
    }

    ~Container(){
        delete [] pchar;
    }

    const char* getString(void) const{
        return pchar;
    }
};

```