

РАЗДЕЛ 1. БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ КОНСТРУКЦИИ

1. Скомпилируется ли это? Если да, то что будет выведено, иначе — объясните, почему не скомпилируется.

```
int a, b;  
a = 4;  
b = ++a++;  
cout << b;
```

2. Что будет выводиться в поток вывода?

```
for(unsigned short int j = 0; j > -1; --j) {  
    cout << j << '\n';  
}
```

3. Вам даны неизветные фрагменты кода A , B . Перепишите цикл $do \{A\} while\{B\}$; через цикл for .

4. `char c = 251;` Скомпилируется ли этот код? Если да, то чему будет равно значение переменной c ?

5. Скомпилируется ли код? Если да, то напишите, что будет выведено, иначе — объясните, почему не скомпилируется.

```
int a = 128;
a &= 96;
cout << "a:" << a << '\n';
int b = 97;
b ^= a;
cout << "b:" << b << '\n';
```

6. Скомпилируется ли код? Конечен ли цикл? Если да, то чему равно значение *a* после выполнения кода?

```
int a = 5;
while(a--) {
    cout << a << '□';
}
```

РАЗДЕЛ 2. СТРУКТУРЫ

7. Что может пойти не так?

```
struct CR {
    int tasks_numb;
    int graves;
}

int main() {
    CR base_1* = new CR;
    base_1.graves = 35;
    base_1.tasks = base_1->graves;
    base_1->--tasks = 5;
    return 0;
}
```

8. Опишите структуру *CrDatabase*, хранящую динамический массив переменных *CRs* типа *CR* (из прошлого задания) и суммарное количество *тяжелых заданий* — переменную *F* целочисленного типа. Объявите переменную типа *CrDatabase* и опишите с помощью нее базу данных кр *my_favourite* с числом контрольных 2 и $F = 70$.

9. Пусть дано две переменные типа *CR*: *1_year* и *2_year*. Напишите код, который проверит, у какой из них процент *graves* от *tasks_numb* больше?

РАЗДЕЛ 3. ФУНКЦИИ

10. Что может пойти не так?

```
int main() {  
    cout << "GL, HF";  
}
```

11. Когда параметры передаются по значению, функция имеет доступ к переменным вызывающей функции? Если да, то к каким?

12. Что такое указатель на функцию? Приведите пример кода, покажите, как он работает.

13. Что может пойти не так?

```
int& f(int c) {  
    return min(c, 3);  
}  
  
int main() {  
    int b = 4;  
    f(b) = 5;  
    int c = 2;  
    f(c) = 1;  
    cout << c;  
}
```

РАЗДЕЛ 4. УКАЗАТЕЛИ и МАССИВЫ

14. Что может пойти не так?

```
int main() {  
    int a = 3;  
    int* p = &a;  
    void* v = p;  
    char c = *v;  
    cout << &c;  
}
```

15. `int* a = new int[5];` В чем различие между $a[3]$ и $a + 3$?

РАЗДЕЛ 5. АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

16. Напишите структуру очередь на двух стеках. Считайте, что у вас реализован полностью стек, реализуйте основные методы очереди.

17. Напишите поиск в двусвязном списке по элементу. Список хранит из содержательного только одну целочисленную переменную. Напишите функцию вставки в двусвязный список, принимающую указатель на элемент, впереди которого надо вставить элемент.

18. Докажите, что *MakeHeap* работает за линейно.