

## Вопросу к экзамену по курсу Алгоритмизация и программирование

### Файл содержит Вопросы, примеры билетов и образцы решения задач

- |   |  |
|---|--|
| Значения и правила<br>запомнить!  | 1. Целочисленные типы данных на C++: размер в байтах, пределы допустимых значений  |
|   | 2. Вещественные типы данных на C++: размер в байтах, пределы допустимых значений   |
|   | 3. Логический и символьный типы данных на C++: размер в байтах, пределы допустимых значений  |
|   | 4. Унарные операции на C++. Их приоритеты. Отличие префиксных и постфиксных форм.  |
| Операции, операторы знать для того, чтобы<br>правильно программировать и грамотно производить<br>анализ кодов   | 5. Особенности арифметических операций на C++. Совмещенный оператор присваивания.  |
|   | 6. Основные арифметические функции на C++. Особенности операции деления на C++.  |
|   | 7. Операции сравнения и логические функции   |
|   | 8. Понятие <i>поток</i> на C++. Консольные потоки ввода/вывода. Управляющие символы в операторе консольного вывода   |
|   | 9. Логический оператор и логическая функция.   |
|   | 10. Оператор-переключатель switch: его структура, для каких случаев его можно использовать.  |
|   | 11. Битовые операции: алгоритм их работы и приоритет выполнения. Приёмы программирования задач с применением битовых операций (задачи)   |
|   | 12. Структура оператора for, разновидности его синтаксиса (задачи).  |
|   | 13. Формула для определения количества повторений в счётном операторе цикла.   |
|   | 14. Условные операторы цикла с пред- и пост- условием (задачи)   |
|   | 15. Задачи по табулированию функций на заданном интервале (задачи)   |
|   | 16. Оператор <i>printf</i> и шаблоны для печати (анализ кодов)   |
| Алгоритмические приёмы и правила необходимо<br>демонстрировать при программировании и анализе<br>кодов программ | 17. Вычисление суммы бесконечного ряда. <b>Рекуррентные</b> соотношения, способы программирования знакопеременных рядов.   |
|   | 18. Функции по работе с псевдослучайными числами. Макрос Rand_Max, функция rand(). Диапазоны значений, в которых генерируются случайные числа. Формулы для вычисления псевдослучайных целых и вещественных чисел. (Задачи) |
|   | 19. Общая форма определения функции. Формальные и фактические параметры. Необязательные аргументы функции (Задачи)   |
|   | 20. Взаимоотношение имени переменной и её адреса в ОП  |
|   | 21. Указатели: определение, свойства, основные операции. (Анализ кодов)  |
|   | 22. Ссылки: определение, свойства, основные операции. (Анализ кодов)   |
|   | 23. Примеры операции разыменования. Понятие разыменования переменной, как функции работают с разыменованной переменной. (Анализ кодов)   |
|   | 24. Различие в передаче параметров по значению и по ссылке. (Анализ кодов)   |
|   | 25. Разработка функции вычисления суммы конечного ряда по заданному аргументу и числу членов ряда. (Задачи)  |

26. Построение функции для построения таблицы заданной функции на заданном интервале изменения аргумента с заданным шагом. (Задачи)
27. Способы передачи статического массива в функцию (Задачи)
28. Разработка функций с применением графиков функций. (Задачи)
29. Для чего необходимы динамические массивы? Как описать одномерный динамический массив? Способы описания двумерного динамического массива.
30. Способы передачи динамических массивов в функции. (Задачи)
31. Символьные данные и строки. Строковые функции. Методы решения задач на обработку символьных массивов и строк. (Задачи)
32. Типы кодировок, используемых при работе с консолью и при операциях с символами в ОП и процессоре. Программы перекодировки символов. (задачи)
33. Программы вставка и удаления символа в строку/из строки.
34. Структуры: шаблон структуры, в каких случаях удобно пользоваться структурой. Какие операции определены со структурами? (Задачи)
35. Понятие полиморфизма. Перегрузки и шаблоны функций. (Задачи на применение перегруженных функций и анализ кодов программ, содержащих перегруженные функции.)
36. Описание класса и его методов (Анализ кодов)
37. Область видимости переменных: автоматические переменные. (Анализ кодов)
38. Статические переменные (Анализ кодов)
39. Глобальные переменные. Правила их описания. Операторы разрешения области видимости (Анализ кодов)

## Примерные варианты билетов

### Вариант 1

1. Особенности арифметических операций на C++. Совмещенный оператор присваивания. Чему будет равен результат вычисления:

```
int i, k = 2, m = 10;  
i = (m)/(m/k - 5);
```

2. Задача: Построить таблицу функции  $F(x) = \sum_{i=1}^N (-1)^i \frac{(2x)^{2i-1}}{(3i)!}$  для

$x \in [-3 \cdot 10^{-1}; +3 \cdot 10^{-1}]$  с шагом  $5 \cdot 10^{-2}$ .  $N$  – любое конечное целое число.

Использовать рекуррентные соотношения и не использовать функцию `pow` для получения степени (-1).

### Вариант 2

1. Различие в передаче параметров по значению и по ссылке: **Какой результат появится на консоли после выполнения программы**

```
int a, b, c, d;  
void f( int a, int& c, int& d)  
{  
    int b;  
    a = 5; c = 7; b = 9;
```

```

}
int main()
{
    a = 1; c = 1; b = 1;  f(b,a,c);  cout<<a <<c << b;
    return 0;
}

```

2. Задача: Зависимость сигнала от времени  $t$  определяется формулой  $U(t) = e^{\sin t + kt}$ . Регистрация сигнала  $U(t)$  реализуется на интервале  $t \in [1;10]$ ,  $\Delta t = 1$ . Создать массив из значений сигнала и найти момент, когда сигнал имеет максимальное значение. Значение параметра  $k = 1,3 \cdot 10^{-3}$

### Вариант 3

1. Определение методов класса. Какой результат будет после выполнения фрагмента программы. Ответ обосновать

```

class t
{ public: int sum;
  float f(int a, short m)
  {
    sum++;
    return sum * a - m;
  }
} v;
int main()
{
    v.sum = 5;
    cout << v.sum << " " << v.f(5, 2);
}

```

2. Задача: Зависимость давления в резервуаре от времени  $t$  определяется формулой:  $P(t) = |\sin(\sqrt{t}) + \lg T(t)|$ . Здесь  $T(t)$  – изменение температуры:  $T(t) = \cos \sqrt{t}$ . На интервале  $t \in [10.2; 20.5]$ ,  $\Delta t = 0.75$  снимаются показания барометра. Построить таблицу, демонстрирующую зависимость давления и температуры от времени. Найти момент, когда давление достигает максимального значения. Составить программу для решения поставленной задачи с разделением задачи на отдельные модули в виде функций

### Вариант 4

1. Общая форма определения функции: Какое определение функции является правильным?

(Ответ считается верным, если отмечены все правильные варианты ответов.)

Вариант 1

```

void f(int b,int c)
{return b+c;}

```

Вариант 2

```
int f(int b,int c)
{return 0;}
```

Вариант 3

```
int f(int b,int c)
{return;}
```

2. Задача: Создать с помощью генератора случайных чисел целочисленный массив A из  $N \in [10;20]$  элементов, значения которых определяются 8-ю старшими битами случайных чисел, вычисляемых в заданном пользователем диапазоне. Отсортировать массив методом пузырька и вывести исходный и отсортированный массивы на консоль?

### Образцы ответов на вопрос с анализом кода программы.

Задача 1: Проанализировать код программы и показать, что будет выведено на консоль, показав и объяснив промежуточные результаты

```
#include<iostream>
#include<cmath>
using namespace std;
int f(int x, int y);
int main()
{   int a[]={6,3,18}, b[]={10,7,13};
    int n=sizeof(a)/2;
    for (int i=0; i<n;i++)
    {
        int z= f(a[i],b[i]);
        cout<<"\n i = "<<i<<" z = "<<z<<endl;
    }
    system("pause");
    return 0;}
int f(int x, int y)
{int c;
  if(x>2 && x<5) c=1;
  else if(y==13) c=2;
  else c=x%y;
  return c;
}
```

Решение:

1. Объявляется пользовательский тип int = беззнаковое короткое целое.
2. Главная программа:
  - a. Описываются и инициализируются два статических массива

индекс	0	1	2
A[]	6	3	18
B[]	10	7	13

b.  $n=4*3/4=3$

с. Цикл по i  
i=0. Z=f(6,10)=6. ←  
Функция f: 1. x=6, y=10; 2.  $x \notin (2;5)$ ,  $y \neq 13$ , тогда  $c=6\%10=6$   
Печать: **i = 0 z = 6**  
i=1. Z=f(3,7)=1 ←  
Функция f: 1. x=3, y=7; 2.  $x \in (2;5)$ , тогда c=1  
Печать: **i = 1 z = 1**  
i=2. Z=f(18,13)=2 ←  
Функция f: 1. x=18, y=13; 2.  $x \notin (2;5)$ ,  $y=13$ , тогда c=2  
Печать: **i = 2 z = 2**

- d. Завершение программы  
e. Консоль решения будет выглядеть следующим образом

```
i = 0 z = 6
i = 1 z = 1
i = 2 z = 2
```

**Задача 2: Проанализировать код программы и показать, что будет выведено на консоль, показав и объяснив промежуточные результаты**

Анализ кода с разграничением области видимости переменных

```
int count = 0; // подписываем count_g
void func()
{
    for (int count = 0; count<10; count++)
    {
        ++ ::count; // cout_f
    }
}
int main()
{
    cout << "count =" << ::count << endl;
    func();
    cout << "count =" << ::count << endl;
    cin.get();
    return 0;
}
```

Решение:

global	main	func
count_g=0		
	count=[count_g]=0	
func::for:		
		count_f=0<10?
count_g=0+1=1		count_f=1<10
count_g=1+1=2		...
...		count_f=9<10
count_g=9+1=10		count_f=10 = =10 выход из цикла
выход из функции func		
	count=[count_g]=10	

Консоль: **count = 0**  
**count = 10**

### Задача 3: Более сложный вариант: (самостоятельно)

```
int c; // внешняя область (контекст)
class X // внешнее объявление класса
{
public:
    static int c;
    class Y // внутренне объявление класса
    {
    public:
        static int c;
        static void f(int key)
        {
            ::c = key; // внешняя переменная
            X::c = key + 1; // переменная с класса X
            c = key + 2; // переменная с класса Y
        }
    };
};
```

```
int main()
{
    X::Y::f(3);
    cout << "X::c = " << X::c << endl;
    cout << "Y::c = " << X::Y::c << endl;
    cin.get();
    return 0;
}
```

```
c = 3
X::c = 4
Y::c = 5
```

Ответ:

```
int X::Y::c = 0;
cout << " c = " << c << endl;
```

### Пример анализа кода с применением символьных операций.

```
int main()
{
    setlocale(LC_ALL, "Russian");
    1. char fruit[] = "яблоко";
    2. char answer[80];
    3. do
    4. {
    5. cout << "Угадайте мой любимый фрукт! >> ";
    6. cin >> answer;
    7. convert(answer, answer);
    8. } while (strcmp(fruit, answer) != 0);
    9. cout << "Правильный ответ!\n";
```

#### Решение

- 1 main: Инициализация символьного массива словом «яблоко»
2. Создаётся символьный статический массив answer
3. Цикл do

5. печать: "Угадайте мой любимый фрукт!"

6. Ввод любого слова

7. Convert:

а) массив answer передаётся в функцию по указателю

б) в теле функции с помощью программы DosToWin буквы кириллицы перекодируются в кодировку windows

в) возврат в вызывающую функцию идёт на тот же массив answer, указатель на который даётся в качестве второго параметра

8. в операторе while перекодированный массив сравнивается со словом «яблоко» с помощью функции strcmp, которая, в случае совпадения answer с fruit выдаст ноль. Если совпадения нет, то возврат к шагу 5, если совпадение есть, то переход на шаг 9

9. main: печать "Правильный ответ!";