

**МИНОБРНАУКИ РОССИИ**  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
**«Ухтинский государственный технический университет»**  
**(УГТУ)**  
**филиал Ухтинского государственного технического университета**  
**в г. Усинске**  
**(УФ УГТУ)**  
(среднего профессионального образования)

УТВЕРЖДАЮ  
И. о. директора филиала  
\_\_\_\_\_ О. В. Филиппова

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

**КОМПЛЕКС ОЦЕНОЧНЫХ СРЕДСТВ**  
**ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**  
**ОПЦ.06 Основы алгоритмизации и программирования**  
образовательной программы  
среднего профессионального образования  
По специальности 09.02.11 Разработка и управление программным  
обеспечением

г. Усинск  
2026

# 1. ПАСПОРТ КОМПЛЕКТА ОЦЕНОЧНЫХ СРЕДСТВ

## 1.1 Область применения

Комплект оценочных средств (далее – КОС) предназначен для контроля и оценки результатов прохождения учебной дисциплины ОПЦ.06 «Основы алгоритмизации и программирования», образовательной программы среднего профессионального образования (далее – ОП СПО) по специальности **09.02.11** Разработка и управление программным обеспечением.

## 1.2 Результаты освоения компетенции

В результате проведения промежуточной аттестации по дисциплине осуществляется комплексная оценка овладения следующими профессиональными и общими компетенциями:

Таблица 2.1

| Код           | Результат освоения компетенции                                                                                                                                                                                                      |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ПК 2.1</b> | Проектировать модули программного обеспечения.                                                                                                                                                                                      |
| <b>ПК 2.2</b> | Разрабатывать модули программного обеспечения.                                                                                                                                                                                      |
| <b>ПК 2.4</b> | Выполнять тестирование и отладку программного обеспечения                                                                                                                                                                           |
| <b>ОК 1</b>   | Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам                                                                                                                                   |
| <b>ОК 2</b>   | Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности                                                                         |
| <b>ОК 3</b>   | Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях |
| <b>ОК 4</b>   | Эффективно взаимодействовать и работать в коллективе и команде                                                                                                                                                                      |
| <b>ОК 5</b>   | Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста                                                                               |
| <b>ОК 9</b>   | Пользоваться профессиональной документацией на государственном и иностранном языках                                                                                                                                                 |

## 2. ФОРМЫ КОНТРОЛЯ И ОЦЕНКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В соответствии с учебным планом и рабочей программой предусматривается текущий и промежуточный контроль результатов освоения.

Предметом оценки служат умения и знания, предусмотренные требованиями ФГОС по дисциплине ОПЦ.06 «Основы алгоритмизации и программирования», направленные на формирование общих и профессиональных компетенций, а также личностных результатов в рамках программы воспитания.

## **2.1 Критерии и нормы оценки знаний обучающихся.**

**Оценка «5»** ставится в случае:

- знания, понимания, глубины усвоения обучающимися всего объёма программного материала;
- умения выделять главные положения в изученном материале, на основании фактов и примеров обобщать, делать выводы, устанавливать межпредметные и внутри предметные связи, творчески применять полученные знания в незнакомой ситуации;
- отсутствия ошибок и недочётов при воспроизведении изученного материала, при устных ответах устранения отдельных неточностей с помощью дополнительных вопросов учителя, соблюдения культуры письменной и устной речи, правил оформления письменных работ.

**Оценка «4»** ставится в случае:

- знания всего изученного программного материала;
- умения выделять главные положения в изученном материале, на основании фактов и примеров обобщать, делать выводы, устанавливать внутри предметные связи, применять полученные знания на практике;
- допущения незначительных (негрубых) ошибок, недочётов при воспроизведении изученного материала; соблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

**Оценка «3»** ставится в случае:

- знания и усвоения материала на уровне минимальных требований программы, затруднения при самостоятельном воспроизведении, возникновения необходимости незначительной помощи преподавателя;
- умения работать на уровне воспроизведения, затруднения при ответах на видоизменённые вопросы;
- наличия грубой ошибки, нескольких грубых ошибок при воспроизведении изученного материала; незначительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

**Оценка «2»** ставится в случае:

- знания и усвоения материала на уровне ниже минимальных требований программы; наличия отдельных представлений об изученном материале;
- отсутствия умения работать на уровне воспроизведения изученного материала, затруднения при ответе на стандартные вопросы;

- наличия нескольких грубых ошибок, большого числа негрубых при воспроизведении изученного материала, значительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

## **2.2 Критерии оценки за тестирование**

Оценка 5 «отлично» (85 % -100 %)

Оценка 4 «хорошо» (75 % - 84 %)

Оценка 3 «удовлетворительно» (61 % -74 %)

Оценка 2 «неудовлетворительно» (менее 60 %)

## 2.3 Формы и методы оценивания учебной дисциплины «Основы алгоритмизации и программирования»

| Раздел, тема                                                 | Формы и методы контроля |                                                    |                          |                                                    |                     |                                                    |
|--------------------------------------------------------------|-------------------------|----------------------------------------------------|--------------------------|----------------------------------------------------|---------------------|----------------------------------------------------|
|                                                              | Текущий контроль        |                                                    | Промежуточная аттестация |                                                    | Итоговая аттестация |                                                    |
|                                                              | Формы контроля          | Проверяемые Знания, умения, ОК, ЛР                 | Формы контроля           | Проверяемые знания, умения ОК, ЛР                  | Формы контроля      | Проверяемые ЗУН, ОК, ЛР                            |
| Тема 1. Основные понятия: алгоритм и языки программирования. | УО, ПР №1               | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09.                | ДФК, Т                   | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09.                | Эк                  | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09.                |
| Тема 2. Типы данных и арифметические операции.               | УО, ПР №2.              | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. | ДФК, Т                   | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. | Эк                  | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК2.4, ПК2.5.  |
| Тема 3. Операторы языка программирования                     | УО, ПР №3.              | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. | ДФК, Т                   | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. | Эк                  | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК2.4, ПК2.5.  |
| Тема 4. Массивы.                                             | УО, ПР №4.              | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. | ДФК, Т                   | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. | Эк                  | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. |
| Тема 5. Функции.                                             | УО, ПР №5.              | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. |                          | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. | Эк                  | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК2.4, ПК2.5.  |
| Тема 6. Ссылки.                                              | ПР № 6. СР.             | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. |                          |                                                    | Эк                  | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК2.4, ПК2.5.  |
| Тема 7. Функции. (рекурсия, перегрузка, шаблон)              | ПР № 7, СР.             | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. |                          |                                                    | Эк                  | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК2.4, ПК2.5.  |
| Тема 8. Строки.                                              | ПР № 8, СР.             | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК 2.4, ПК2.5. |                          |                                                    | Эк                  | ОК 01, ОК 0,2, ОК 04, ОК 05, ОК 09, ПК2.4, ПК2.5.  |

|                                                             |                |                                                             |  |  |    |                                                            |
|-------------------------------------------------------------|----------------|-------------------------------------------------------------|--|--|----|------------------------------------------------------------|
| Тема 9.<br>Организация<br>файлового потока<br>ввода/вывода. | ПР № 9, СР.    | ОК 01, ОК 0,2,<br>ОК 04, ОК 05,<br>ОК 09, ПК 2.4,<br>ПК2.5. |  |  | Эк | ОК 01, ОК<br>0,2, ОК 04,<br>ОК 05, ОК 09,<br>ПК2.4, ПК2.5. |
| Тема 10. Указатели.                                         | ПР № 10.<br>СР | ОК 01, ОК 0,2,<br>ОК 04, ОК 05,<br>ОК 09, ПК 2.4,<br>ПК2.5. |  |  | Эк | ОК 01, ОК<br>0,2, ОК 04,<br>ОК 05, ОК 09,<br>ПК2.4, ПК2.5. |

### Кодификатор оценочных средств

| <b>Функциональный признак<br/>оценочного средства<br/>(тип контрольного задания)</b> | <b>Код оценочного средства</b> |
|--------------------------------------------------------------------------------------|--------------------------------|
| Устный опрос                                                                         | УО                             |
| Практическая работа                                                                  | ПР №                           |
| Самостоятельная работа                                                               | СР                             |
| Тестирование                                                                         | Т                              |
| Другая форма контроля                                                                | ДФК                            |
| Экзамен                                                                              | Эк                             |

### 3. Контрольно-оценочные материалы для промежуточной и итоговой аттестации по учебной дисциплине «Основы алгоритмизации и программирования»

#### 3.1 Типовые задания для проведения текущего контроля

##### 3.1.1 Перечень практических работ.

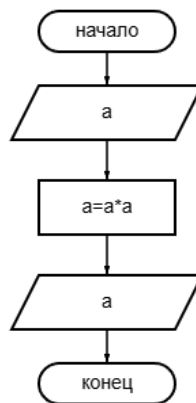
##### 3 семестр

##### Практическая работа № 1.

1. Создать блок – схему используя Microsoft Visio или онлайн: <https://programforyou.ru/block-diagram-redactor> для решения следующих задач:

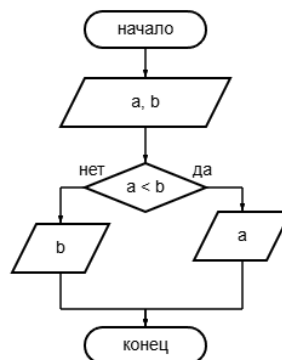
**Задача № 1.** Вычислить квадрат, введенного пользователем числа. Линейная структура.

**Решение:**



**Задача № 2.** Найти минимальное число из двух введенных чисел. Разветвляющаяся структура.

**Решение:**



##### Практическая работа № 2.

**Задача № 1.** Знакомство со средой программирования Visual Studio. Изучение структуры программы C++. Ввод, вывод данных, используя операторы **cin** и **cout**. Объявление и инициализация переменных типа **int**. Разработать программу по блок-схеме ПР № 1 задача № 1 для переменных целочисленного типа (язык программирования C++).

**Решение:**

**На основе блок-схемы 1 задачи темы № 1.**

|                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> #include &lt;iostream&gt; using namespace std; int main() {     setlocale(LC_ALL, "Russian");     int a, a2; // объявление переменной     cin &gt;&gt; a; //ввод значения     a2 = a * a;     cout &lt;&lt; "Квадрат числа " &lt;&lt; a &lt;&lt; " = " &lt;&lt; a2; } </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     setlocale(LC_ALL, "Russian");     int a, a2; // объявление переменной     cin &gt;&gt; a; //ввод значения     a2 = pow(a,2);     cout &lt;&lt; "Квадрат числа " &lt;&lt; a &lt;&lt; " = " &lt;&lt; a2; } </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Задача № 2.** Объявление и инициализация переменных типа **char**. Преобразование символа в значение десятичного кода. Тип данных **float, double**. Разработать программу нахождения суммы двух чисел с плавающей точкой (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    float a, b, sum; // объявление переменной
    cin >> a; //ввод значения
    cin >> b;
    sum = a+b;
    cout << "Сумма двух чисел = " << sum;
}

```

### Практическая работа № 3.

#### 1. Составление программ разветвляющейся структуры. (if, switch, тернарный оператор).

**Задача № 1.** Напишите программу, которая выводит сообщение «Yes!», если целое число является четным, и «NO!», если число не четное, используя оператор if (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    int a;
    cin >> a; //ввод значения
    if (a % 2 == 0) {
        cout << "Yes!";
    }
    else
        cout << "No";
}

```

**Задача № 2.** Разработать программу, используя блок – схему задачи 2 ПР № 1, используя тернарный оператор (язык программирования C++).

**Решение:**

**На основе блок-схемы задачи 3 темы № 1.**

```

#include <iostream>
using namespace std;

int main()
{

```

```

    int a, b;
    cin >> a >> b;
    a < b ? cout << a : cout << b;
}

```

## 2. Составление программ циклической структуры. (do ...while, while)

**Задача № 1.** Используя цикл **do ...while**, напишите программу. На ввод поступает число, а затем последовательность чисел, включающее это число. Программа должна остановиться при встрече введенного числа в последовательности. Выведите на печать порядковый номер данного числа (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;
int main()
{
    int n, i=0, m;
    cin >> n;
    do
    {
        cin >> m;
        i++;
        if (n == m)
            break;
    } while (n != m);
    cout << i;
}

```

**Задача № 2.** Используя цикл **while**, напишите программу. На ввод поступает целое число *n*. Программа должна вывести все четные числа до числа *n* включительно через пробел (язык программирования C++).

**Решение:**

```

#include<iostream>
#include <cmath>
using namespace std;

int main() {
    setlocale(LC_ALL, "Russian");
    int n;
    cout << "Введите число:";
    cin >> n;
    int i = 2;
    while (i <= n) {
        cout << i << " ";
        i += 2;
    }
}

```

## 3. Составление программ циклической структуры. for.

**Задача № 1.** На вход программе подаются натуральное число *n*, а затем *n* целых чисел, каждое на отдельной строке. Напишите программу, которая подсчитывает сумму введенных чисел (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;

int main()
{
    int n, n1, sm = 0;
    cin >> n;
    for (int i = 0; i < n; i++)
    {

```

```

        cin >> n1;
        sm += n1;
    }
    cout << sm;
    return 0;
}

```

**Задача № 2.** Используя цикл **for**, напишите программу. На ввод подается целое число  $n$ . Составьте программу, которая печатает все числа, кратные введенному числу не больше 50, начиная с самого числа. Вывод последовательность чисел предоставьте через пробел (язык программирования C++).

**Решение:**

|                                                                                                                                                                                                     |                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n;     cin &gt;&gt; n;     for (int i = n; i &lt; 50; i += n)     {         cout &lt;&lt; i &lt;&lt; "\t";     } } </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n;     cin &gt;&gt; n;     for (int i = 1; i*n &lt; 50; i++)     {         cout &lt;&lt; i*n &lt;&lt; "\t";     } } </pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### Практическая работа № 4.

**Задача № 1.** Напишите программу, которая считывает с клавиатуры значения элементов массива и выводит их на экран (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "");
    int const n = 5;
    float mas[n];
    cout << "Введите элементы массива:" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "mas[" << i+1 << "]=";
        cin >> mas[i];
    }
    cout << "Все элементы массива:" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "mas[" << i+1 << "]= " << mas[i] << endl;
    }
}

```

**Задача № 2.** Напишите программу, которая выводит среднее значение элементов массива размером 5 после ввода значений элементов массива (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "");
    int const n = 5;
    float mas[n];
    float sum=0;
    cout << "Введите элементы массива:" << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "mas[" << i+1 << "]=";
        cin >> mas[i];
    }
}

```

```

    }
    cout << "Все элементы массива:" << endl;
    for (int i = 0; i < n; i++)
    {
        sum += mas[i];
    }
    cout << "Среднее значение массива:" << sum / n << endl;
}

```

## Практическая работа № 5.

**Задача № 1.** Напишите функцию `lin_ur(a, b)`, которая возвращает корень линейного уравнения:  $ax + b = 0$ ; (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;

double lin_ur(double a, double b)
{
    return (-b) / a;
}

int main()
{
    double a, b;
    cin >> a >> b;
    cout << lin_ur(a, b);
}

```

**Задача № 2.** Напишите функцию `gip(a, b)`, принимающую в качестве аргумента значения катетов прямоугольного треугольника и возвращающую значение гипотенузы (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;

double gip(double a, double b)
{
    return sqrt(pow(a, 2) + pow(b, 2));
}

int main()
{
    double a, b, c;
    cin >> a >> b;
    c = gip(a, b);
    cout << c;
}

```

## Практическая работа № 6.

**Задача № 1.** Разработайте функцию обмена значений двух переменных выполнив передачу параметров по ссылке (язык программирования C++).

**Решение:**

```

void swap(int& a, int& b)
{
    int temp = a;
    a = b;
    b = temp;
}

int main()
{

```

```

int x , y;
cin >> x >> y;
swap(x, y);
cout << "x = " << x;
cout << "y = " << y;
}

```

### Практическая работа № 7.

**Задача № 1.** Дана последовательность чисел, завершающаяся числом 0. Найдите сумму всех этих чисел, не используя цикл (рекурсия функции) (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;

double sum(double s) {
    double a;
    cin >> a;
    if (a == 0) { return s; }
    else { s = s + a; return sum(s); }
}

int main() {
    int n;
    double s = 0;
    cout << sum(s);
    return 0;
}

```

### Практическая работа № 8.

**Задача № 1.** По данной строке определите, является ли она палиндромом (то есть, читается одинаково как слева-направо, так и справа-налево) (язык программирования C++).

**Решение:**

```

#include <iostream>
#include <string>
using namespace std;

int main() {
    string s, s1 = "";
    char c;
    getline(cin, s);
    for (auto c : s) {
        s1 = c + s1;
    }
    if (s == s1) { cout << "yes"; }
    else { cout << "no"; }
    return 0;
}

```

### Практическая работа № 9.

**Задача № 1.** Дан текстовый файл f.txt. Переписать в файл g.txt все его строки в перевернутом виде (язык программирования C++).

**Решение:**

```

#include <fstream>
#include <iostream>

```

```

#include <string>
using namespace std;
int main()
{
    ifstream in("f.txt");
    ofstream out("g.txt");

    string s;

    // пока не прочитан маркер файла
    while (in.peek() != EOF)
    {
        // читает очередную строку из файла f.txt
        getline(in, s);
        for (int i = 0; i < (int)s.length() / 2;
             i++) // зеркально отображает строку
        {
            char a = s[i];
            s[i] = s[s.length() - 1 - i];
            s[s.length() - 1 - i] = a;
        }
        // записываем измененную строку в файл g.txt
        out << s << endl;
    }
    in.close();
    out.close();
    return 0;
}

```

**Задача № 2.** Дан файл f.txt, компонентами которого являются целые числа. В файл g.txt переписать все неотрицательные компоненты файла f.txt, кратные трем (язык программирования C++).

**Решение:**

```

#include <fstream>
#include <iomanip>
using namespace std;
int main()
{
    ifstream in("f.txt");
    ofstream out("g.txt");
    int i;
    // пока не прочитан маркер конца файла, читаем из потока
    // очередной компонент
    while (in >> i)
    {
        // если компонент отвечает заданным требованиям
        if ((i >= 0) && (i % 3 == 0))
            out << setw(5) << i; // то помещаем его в выходной поток
    }
    in.close();
    out.close();
    return 0;
}

```

**Задача № 1.** Найти сумму отрицательных элементов динамического массива (язык программирования C++).

**Решение:**

```
#include <iostream>
// #include <locale>
using namespace std;

int main()
{
    setlocale(LC_CTYPE, "rus");
    int n;
    int* a; // указатель на первый элемент
    cout << "Введите количество элементов массива n: ";
    cin >> n;
    while (n <= 0)
    {
        cout << "Введите положительное число n: ";
        cin >> n;
    }
    // Выделение памяти для массива размера n
    a = new int[n];
    while (a == NULL)
    {
        cout << "Не возможно выделить такое количество памяти" << endl;
        cout << "Попробуйте ввести меньшее количество: ";
        cin >> n;
    }
    cout << "Введите элементы массива:" << endl;
    for (int i = 0; i < n; i++)
        cin >> a[i];
    int sum = 0;
    for (int i = 0; i < n; i++)
    {
        if (a[i] < 0)
            sum += a[i];
    }
    if (sum == 0)
        cout << "Нет отрицательных элементов в массиве" << endl;
    cout << "sum = " << sum;
    // освобождение памяти, занимаемой массивом
    delete[] a;
    return 0;
}
```

### 3.1.2. Примерные вопросы для устного опроса обучающихся по учебной дисциплине

1. Опишите основную структура программы языка программирования C++?
2. Как подключить библиотеку потока ввода/вывода языка программирования C++?
3. Как подключить пространство имен std языка программирования C++?
4. Как объявить и инициализировать переменную целочисленного типа (C++)?
5. Как объявить и инициализировать переменную вещественного типа (C++)?
6. Как объявить и инициализировать переменную символьного типа (C++)?
7. Что такое условный оператор?
8. Что такое цикл с предусловием? Какие виды циклов с предусловием в C++ можно использовать?
9. Что такое цикл с постусловием? Какие виды циклов с постусловием в C++ можно использовать?
10. Что такое одномерный массив? Как выполнить ввод/вывод одномерного массива размером 5?
11. Что такое двумерный массив? Как выполнить ввод/вывод двумерного массива размером 3×6?

### 3.1.3 Перечень самостоятельных работ.

**Задача № 1.** На ввод поступает положительное целое число (с консоли). Программа должна сформировать новое число, цифры в котором идут в обратном порядке. Для решения данной задачи используйте цикл while (**язык программирования C++**).

**Задача № 2.** Дан массив чисел. Посчитайте, сколько в нем пар элементов, равных друг другу. Считается, что любые два элемента, равные друг другу образуют одну пару, которую необходимо посчитать. Размер массива равна 9. Элементы массива вводит пользователь (**язык программирования C++**).

### **3.2. Типовые задания для проведения промежуточной аттестации**

#### **Другая форма контроля**

#### **3 семестр**

**Часть 1. Только один верный ответ.**

**1. Алгоритм - это:**

- a) правила выполнения определенных действий;
- b) ориентированный граф, указывающий порядок выполнения некоторого набора команд;
- c) описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов;**
- d) набор команд для компьютера;
- e) протокол вычислительной сети;

**2. Алгоритм называется линейным, если:**

- a) составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- b) ход его выполнения зависит от истинности тех или иных условий;
- c) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;**
- d) включает в себя вспомогательный алгоритм;

**3. Алгоритм называется циклическим, если:**

- a) составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;**
- b) ход его выполнения зависит от истинности тех или иных условий;
- c) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
- d) включает в себя вспомогательный алгоритм;

**4. Алгоритм включает в себя ветвление, если:**

- a) составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
- b) ход его выполнения зависит от истинности тех или иных условий**
- c) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
- d) включает в себя вспомогательный алгоритм

**5. Свойством алгоритма является:**

- a) результативность**
- b) цикличность
- c) возможность изменения последовательности выполнения команд

**6. Свойство алгоритма, заключающиеся в том, что алгоритм поддается расчленению на элементарные шаги, которые могут быть исполнены при помощи системы команд исполнителя, называется**

- a) дискретность**
- b) детерминированность
- c) конечность
- d) массовость

7. Свойство алгоритма, заключающиеся в том, что один и тот же алгоритм можно использовать с разными исходными данными, называется
- a) дискретность
  - b) детерминированность
  - c) **массовость**
  - d) конечность
8. Алгоритм, записанный на «понятном» компьютеру языке программирования, называется
- a) **программой**
  - b) текстовкой
  - c) листингом
9. Графическое задание алгоритма (блок-схема) - это:
- a) **Способ графического представления алгоритма, в котором шаги изображаются в виде блоков различной формы, соединенных между собой стрелками;**
  - b) Система обозначения правил для единообразной и точной записи алгоритмов их исполнения
  - c) Представление алгоритма в форме таблиц и расчетных формул
  - d) Схематическое изображение в произвольной форме
10. Какой из документов является алгоритмом?
- a) Бухгалтерский отчет
  - b) **Инструкция по приготовлению пищи**
  - c) Список книг в школьной библиотеке
11. Какая фигура в блок-схеме предназначен для вывода данных?
- a) прямоугольник
  - b) **параллелограмм**
  - c) ромб
  - d) трапеция
12. Какой тип данных относится к символьному:
- a) int
  - b) double
  - c) **char**
  - d) float
13. Какой цикл является с постусловием:
- a) for
  - b) **do ... while**
  - c) while
14. Что выведет на экран данный оператор в языке программирования C++, если  $a=12$ ,  $b=5$ :
- ```
if (a > b) cout << a;  
else cout << b;
```
- a) **12**
  - b) 5
  - c) **Ничего**
15. При выполнении операции в языке программирования C++, какой будет результат:  $a = 735$ ;  $a=a\%10$ ;
- a)  **$a=7$ ;**

- b) a=3;
- c) a=5;

16. При выполнении операции в языке программирования C++, какой будет результат: a=12; b=4; a=a%b:

- a) a=12, b=4;
- b) a=0, b=4;
- c) a=3, b=4;

17. Каким способом возможно получить первую цифру целого трехзначного числа a в языке программирования C++:

- a) a%10
- b) a/10
- c) a/100

18. Определите, что находит данная программа (язык программирования C++):

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    cin >> a;
    if (a % 2 == 0) cout << "Yes!";
    else cout << "No!";
}
```

- a) Минимальное значение;
- b) Четные числа;
- c) Числа, в котором присутствует цифра 2;

19. Определите в какой строке присутствует ошибка (язык программирования C++):

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a;
6      cin >> a
7      cout << a++;
8  }
9
```

- a) 5
- b) 6
- c) 7

20. Определите какой будет результат программы (язык программирования C++):

```
#include <iostream>
using namespace std;
int main()
{
    int a = 3;
    for (int i = 0; i < a; i++) {
        cout << i << " ";
    }
    return 0;
}
```

- a) 0 1 2
- b) 0 1 2 3

с) 1 2 3

**Часть 2. Выберите соответствие.**

**21. Расставьте числа в таком порядке, чтобы короткая запись соответствовала арифметической операции (язык программирования C++):**

|    |                   |    |                    |
|----|-------------------|----|--------------------|
| 1. | <code>a+=b</code> | a) | <code>a=a/b</code> |
| 2. | <code>a*=b</code> | b) | <code>a=a*b</code> |
| 3. | <code>a/=b</code> | c) | <code>a=a+b</code> |

**Ответ:**

1 – c

2 – b

3 – a

**22. Укажите в каком порядке выглядит структура программы вывода информации на экран (язык программирования C++):**

|    |                                           |
|----|-------------------------------------------|
| 1. | <code>int main(){</code>                  |
| 2. | <code>using namespace std;</code>         |
| 3. | <code>return 0}</code>                    |
| 4. | <code>#include &lt;iostream&gt;</code>    |
| 5. | <code>cout &lt;&lt; "Hello word!";</code> |

**Ответ:**

42153

**23. Расставьте числа в таком порядке, чтобы операторы соответствовали их значению (язык программирования C++):**

|    |                   |    |                   |
|----|-------------------|----|-------------------|
| 1. | <code>cout</code> | a) | Оператор ввода    |
| 2. | <code>cin</code>  | b) | Оператор вывода   |
| 3. | <code>if</code>   | c) | Условный оператор |

**Ответ:**

1 – b

2 – a

3 – c

## **3.2 Организация и проведение итоговой аттестации**

### **3.2.1 Условия проведения итоговой аттестации. Подготовка к проведению итоговой аттестации.**

Итоговая аттестация проводится в период экзаменационной сессии, установленной календарным графиком учебного процесса рабочего учебного плана. С формами проведения промежуточной аттестации обучающиеся знакомятся в течение двух месяцев с начала обучения.

Для подготовки к итоговой аттестации студентом (не позднее чем за 20 дней до проведения экзамена в соответствии с календарным графиком учебного процесса) выдается перечень тем к практическим заданиям, составленный исходя из требований ФГОС и рабочей программы предмета к уровню умений и знаний. Количество вопросов и практических задач в перечне для подготовки к промежуточной аттестации превышает количество вопросов и практических задач, необходимых для составления контрольно-измерительных материалов.

На основе разработанного и объявленного обучающимся перечня вопросов и практических задач, рекомендуемых для подготовки к итоговой аттестации, составлены задания, содержание которых до обучающихся не доводится. Вопросы и практические задачи носят равноценный характер. Формулировки вопросов четкие, краткие, понятные, исключают двойное толкование.

### **3.2.2 Проведение итоговой аттестации**

Студенты для сдачи итоговой аттестации распределяются по времени. На выполнение задания студенту отводится не более двух академических часов. Оценка, полученная на итоговой аттестации, заносится преподавателем в зачетную книжку студента (кроме неудовлетворительной) и экзаменационную ведомость (в том числе и неудовлетворительные). Экзаменационная оценка по предмету за данный семестр является определяющей, независимо от полученных в семестре оценок текущего контроля по предмету

**ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ  
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ  
4 семестр**

**Теоретические вопросы:**

1. Алгоритмы. Правила записи схем алгоритмов. Определить значение переменной *s* после выполнения следующих операторов: `int s=0; int n=4; for (int i=2; i< n; i++) s+=10/i;`  
**Решение:**  
`s = 8`
2. Структура программы на языке C++. Чему равны значения переменных *a* и *b* после выполнения последовательности действий: `int a=15 /(16 % 7); int b=34 / a*5 – 29 % 5*2;`  
**Решение:**  
`a = 7`  
`b = 12`
3. Арифметические операции, приоритет операций. Выражения. Примеры. Чему равны значения переменных *a* и *b* после выполнения последовательности действий: `int a=4*5 / 3 % 2; int b= 4*5 /(3 % 2)`  
**Решение:**  
`a = 0`  
`b = 20`
4. Типы данных (`integer`, `double`, `string`, массивы) – назначение, описание. Чему равны значения переменных *a* и *b* после выполнения последовательности действий: `int a=15 /(16 % 7); int b=34 / a*5 – 29 % 5*2;`  
**Решение:**  
`a = 7`  
`b = 12`
5. Строковый тип данных. Основные функции работы со строками. Определите значение переменной *p* после выполнения следующего фрагмента программы: `int p, m= 13; int n= 21; n= 2*m – n; if (m<=n ) p= m + n; else p= 4 – m*n;`  
**Решение:**  
`p = -61`
6. Операторы ввода/вывода. Форматы вывода данных. Чему равны значения переменных *a* и *b* после выполнения последовательности действий: `int a= 4*5 / 3 % 2; int b= 4*5 / (3 % 2);`  
**Решение:**  
`a = 0`  
`b = 20`
7. Арифметические функции. Пример. Чему равны значения переменных *a* и *b* после выполнения последовательности действий: `int a= 15 / (16 % 7); int b=34 % a*5 – 29 % 5*2;`

**Решение:**

a = 7  
b = 12

8. Основные символы языка C++. Примеры записи комментариев. Чему равны значения переменных a и b после выполнения последовательности действий: `int a= 4*5 / 3 % 2;`  
`int b= 4*5 /(3 %2);`

**Решение:**

a = 0  
b = 20

9. Алгоритм нахождения количества и суммы натуральных чисел. Определить значение переменной s после выполнения следующих операторов: `int s=0; int n=5; for (int i=2; i<= n; i++) s+= 100 / i;`

**Решение:**

s = 128

10. Переменные (назначение, описание в программе). Имена переменных (правила задания, примеры). Чему равны значения переменных a и b после выполнения последовательности действий: `int a= 15 /(16 % 7); int b=34 / a*5 – 29 % 5*2;`

**Решение:**

a = 7  
b = 12

11. Алгоритм обмена местами элементов массива. Чему равны значения переменных a и b после выполнения последовательности действий: `int a= 4*5 / 3 % 2; int b= 4*5 /(3 % 2);`

**Решение:**

a = 0  
b = 20

12. Операторы switch, if, тернарный оператор (назначение и отличие). Определить значение переменной p после выполнения следующего фрагмента программы: `int m = 13; int n= 21; n= 2*m – n; if (m<=n) p= m + n; else p= 4–m*n;`

**Решение:**

p = -61

13. Условный оператор if (if..., if...else). Вложенные условные операторы. Определить значение переменной p после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m – n; if (m<=n ) p= m + n; else p= 4 – m*n;`

**Решение:**

p = -61

14. Операторы цикла for, while и do... while, блок-схемы. Какими будут значения переменных s после выполнения следующих операторов: `int s=2; int i= 1; while (i<10) {s+= i; i+= 1;}`

**Решение:**

s = 47

15. Ввод/вывод одномерных массивов. Пример. Определить значение переменной s после выполнения следующих операторов: `int s=0; int n=5; for (int i=2 ; i< n; i++) s+= 100 / i;`

**Решение:**

s = 128

16. Алгоритм сортировки массива. Какими будут значения переменных s и i после выполнения следующих операторов: `int s=20; int i= 0; while (i<=10) { s+= 5; i+= 1;}`

**Решение:**

s = 75

17. Ввод/вывод двумерных массивов. Какими будут значения переменных a и b после выполнения следующих операторов: `int a= 1; int b= 2; while (b<20) { a+= 2; b+=2;}`

**Решение:**

a = 19

b = 20

18. Алгоритм нахождения максимального элемента в массиве (одномерный массив).

19. Назначение функции. Описание и вызов функции. Чему равны значения переменных a и b после выполнения последовательности действий: `int a= 4*5 /3 % 2; b=4*5/ (3 %2)`

**Решение:**

a = 0

b = 20

20. Подпрограммы (структура подпрограммы, схема алгоритма). Чему равны значения переменных a и b после выполнения последовательности действий: `int a= 4*5 / 3 % 2; int b= 4*5 /(3 %2);`

**Решение:**

a = 0

b = 20

21. Файлы (общие сведения, характеристики, описание файловых переменных). Основные процедуры работы с файлами.

22. Алгоритм сортировки массива: метод линейного выбора.

23. Алгоритм сортировки массива: метод пузырька.

24. Алгоритм нахождения минимального элемента в массиве (одномерный массив).

25. Алгоритм нахождения порядкового номера максимального элемента в массиве (одномерный массив).

26. Алгоритм нахождения порядкового номера минимального элемента в массиве (одномерный массив).
27. Алгоритм нахождения максимального элемента в массиве (двумерный массив).
28. Алгоритм нахождения минимального элемента в массиве (двумерный массив).
29. \*Компиляция и отладка программы: Использование точек останова. Просмотр и изменение значений элементов данных.
30. \*Компиляция и отладка программы: Типы сообщений компилятора. Использование встроенного отладчика. Трассировка программы.

## Практические задания:

1. Разработать и произвести отладку программы для решения квадратного уравнения. (язык программирования C++)

Решение:

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    int a, b, c, D;
    float x1, x2;
    cin >> a >> b >> c;
    D = pow(b, 2) - 4 * a * c;
    if (D < 0)
    {
        cout << "Нет корней" << endl;
    }
    else if (D == 0)
    {
        x1 = -b / 2 * a;
        cout << x1 << endl;
    }
    else
    {
        float Dpw = pow(D, 0.5);
        x1 = (-b + Dpw) / 2 * a;
        x2 = (-b - Dpw) / 2 * a;
        cout << x1 << endl;
        cout << x2 << endl;
    }
}
```

2. Создать и отладить приложение – конвертор перевода суммы денег из долларов в рубли (язык программирования C++).

Решение:

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    const float ru = 98.26;
    float dllr;

    cout << "Введите валюту: ";
    cin >> dllr;

    float rb = dllr * ru;
    cout << "Стоимость " << dllr << " $ в RU составит(" << rb << ").";
}
```

3. Разработать и произвести отладку программы для вычисления делителей натурального числа N. Вывести сами делители (язык программирования C++).

Решение:

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
```

```

    int n;

    cout << "Введите натуральное число: ";
    cin >> n;
    for (int i = 1; i <= n; i++) {
        if (n % i == 0) {
            cout << i << " ";
        }
    }
}

```

4. Разработать и произвести отладку программы, вычисляющей сумму 1-й и последней цифр натурального числа N. Вывести эти цифры и сумму (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    int n;

    cout << "Введите натуральное число: ";
    cin >> n;
    int nk = n;
    int k = 0;
    while (n != 0) {
        k++;
        n /= 10;
    }
    int n1 = nk / pow(10, (k - 1));
    int nn = nk % 10;
    cout << n1 << " " << nn;
    cout << nn + n1 << endl;
}

```

5. Разработать и произвести отладку программы, находящей все простые числа в заданном диапазоне (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    int a, b;
    int f = 0;
    cout << "Введите начало диапазона: ";
    cin >> a;
    cout << "Введите конец диапазона: ";
    cin >> b;
    for (int i = a; i <= b; i++) {
        for (int j = 2; j < b; j++) {
            if (i % j == 0 && i != j)
                f = 1;
        }
        if (f == 0) {
            cout << i << " ";
        }
        f = 0;
    }
}

```

```

    }
}

```

6. Разработать и произвести отладку программы, находящей все нечетные числа в заданном диапазоне и их количество (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    int a, b;
    int k = 0;
    cout << "Введите начало диапазона: ";
    cin >> a;
    cout << "Введите конец диапазона: ";
    cin >> b;
    for (int i = a; i <= b; i++) {
        if (i % 2 != 0) {
            cout << i << " ";
            k++;
        }
    }
    cout << endl;
    cout << "Всего нечетных чисел в диапазоне от " << a << " до " << b << "
составляет: " << k;
}

```

7. Разработать и произвести отладку программы, находящей все четные числа в заданном диапазоне и их количество (язык программирования C++).

**Решение:**

```

#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    int a, b;
    int k = 0;
    cout << "Введите начало диапазона: ";
    cin >> a;
    cout << "Введите конец диапазона: ";
    cin >> b;
    for (int i = a; i <= b; i++) {
        if (i % 2 == 0) {
            cout << i << " ";
            k++;
        }
    }
    cout << endl;
    cout << "Всего четных чисел в диапазоне от " << a << " до " << b << "
составляет: " << k;
}

```

8. Разработать и произвести отладку программы, которая выполняет замену отрицательных чисел на 0, положительных – на 1 и выводит список элементов массива. Размер одномерного массива 10. Ввод элементов массива производится через клавиатуру (язык программирования C++).

**Решение:**

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    int a[10];

    for (int i = 0; i < 10; i++) {
        cout << "Введите элемент массива a[" << i+1 << "] = ";
        cin >> a[i];
        if (a[i] > 0) {
            a[i] = 1;
        }
        else {
            a[i] = 0;
        }
    }

    for (int i = 0; i < 10; i++) {
        cout << "a[" << i+1 << "] = " << a[i] << endl;
    }
}
```

9. Разработать и произвести отладку программы, которая определяет минимальный и максимальный элементы массива. Размер одномерного массива 8. Ввод элементов массива производится через клавиатуру (язык программирования C++).

**Решение:**

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    int a[8];
    int mx = 0;
    int mn = INT64_MAX;

    for (int i = 0; i < 8; i++) {
        cout << "Введите элемент массива a[" << i+1 << "] = ";
        cin >> a[i];
        if (a[i] > mx) {
            mx = a[i];
        }
        if (a[i] < mn) {
            mn = a[i];
        }
    }

    cout << "Максимальный элемент массива: " << mx << endl;
    cout << "Минимальный элемент массива: " << mn << endl;
}
```

10. Разработать и произвести отладку программы, которая определяет сумму всех элементов и количество положительных элементов в массиве размером 7. Ввод элементов массива производится через клавиатуру (язык программирования C++).

**Решение:**

```
#include <iostream>
```

```
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    int a[5];
    int sm=0;
    int k=0;

    for (int i = 0; i < 5; i++) {
        cout << "Введите элемент массива a[" << i+1 << "] = ";
        cin >> a[i];
        if (a[i] > 0) {
            k++;
        }
        sm += a[i];
    }

    cout << "Общая сумма элементов массива: " << sm << endl;
    cout << "Количество положительных элементов массива: " << k << endl;
}
```

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Алгоритмы. Правила записи схем алгоритмов.

Определить значение переменной  $s$  после выполнения следующих операторов:

`int s=0; int n=4; for (int i=2; i< n; i++) s+=10/i;`

2. Файлы (общие сведения, характеристики, описание файловых переменных). Основные процедуры работы с файлами.

3. Практическое задание:

Разработать и произвести отладку программы для решения квадратного уравнения.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Структура программы на языке C++. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a=15 / (16 % 7); int b=34 / a*5 - 29 % 5*2;`

2. Алгоритм сортировки массива: метод линейного выбора.

3. Практическое задание:

Создать и отладить приложение – конвертор перевода суммы денег из долларов в рубли.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 3

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Арифметические операции, приоритет операций. Выражения. Примеры. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a=4*5 / 3 % 2; int b= 4*5 / (3 % 2)`

2. Алгоритм сортировки массива: метод пузырька.

3. Практическое задание:

Разработать и произвести отладку программы для вычисления делителей натурального числа  $N$ . Вывести сами делители, их количество.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 4

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Типы данных (`integer`, `double`, `string`, массивы) – назначение, описание. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a=15 / (16 % 7); int b=34 / a*5 - 29 % 5*2;`

2. Алгоритм нахождения минимального элемента в массиве (одномерный массив)

3. Практическое задание:

Разработать и произвести отладку программы, вычисляющей сумму 1-й и последней цифр натурального числа  $N$ . Вывести эти цифры и сумму.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 5

#### ОПЦ.04. «Основы алгоритмизации и программирования»

- 1 Строковый тип данных. Основные функции работы со строками. Определите значение переменной  $p$  после выполнения следующего фрагмента программы:  $\text{int } m = 13; \text{int } n = 21; n = 2 * m - n; \text{if } (m \leq n) \text{ } p = m + n; \text{else } p = 4 - m * n;$
2. Алгоритм нахождения порядкового номера максимального элемента в массиве (одномерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, находящей все простые числа в заданном диапазоне.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 6

#### ОПЦ.04. «Основы алгоритмизации и программирования»

- 1 Операторы ввода/вывода. Форматы вывода данных. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий:  $\text{int } a = 4 * 5 / 3 \% 2; \text{int } b = 4 * 5 / (3 \% 2);$
2. Алгоритм нахождения порядкового номера минимального элемента в массиве (одномерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, находящей все нечетные числа в заданном диапазоне и их количество.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 7

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Арифметические функции. Пример. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий:  $\text{int } a = 15 / (16 \% 7); \text{int } b = 34 \% a * 5 - 29 \% 5 * 2;$
2. Алгоритм нахождения максимального элемента в массиве (двумерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, находящей все четные числа в заданном диапазоне и их количество.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 8

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Основные символы языка C++. Примеры записи комментариев. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий:  $\text{int } a = 4 * 5 / 3 \% 2; \text{int } b = 4 * 5 / (3 \% 2);$
2. Алгоритм нахождения минимального элемента в массиве (двумерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, которая выполняет замену отрицательных чисел на 0, положительных – на 1 и выводит список элементов массива. Размер одномерного массива 10. Ввод элементов массива производится через клавиатуру.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 9

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Алгоритм нахождения количества и суммы натуральных чисел. Определить значение переменной  $s$  после выполнения следующих операторов: `int s=0; int n=5; for (int i=2; i<= n; i++) s+= 100 / i;`
2. Компиляция и отладка программы: Использование точек останова. Просмотр и изменение значений элементов данных.
3. Практическое задание:  
Разработать и произвести отладку программы, которая определяет минимальный и максимальный элементы массива. Размер одномерного массива 8. Ввод элементов массива производится через клавиатуру.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 10

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Переменные (назначение, описание в программе). Имена переменных (правила задания, примеры). Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a= 15 /(16 % 7); int b=34 / a*5 – 29 % 5*2;`
2. Компиляция и отладка программы: Типы сообщений компилятора. Использование встроенного отладчика. Трассировка программы.
3. Практическое задание:  
Разработать и произвести отладку программы, которая определяет сумму всех элементов и количество положительных элементов.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 11

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Алгоритм обмена местами элементов массива. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a= 4*5 / 3 % 2; int b= 4*5 /(3 % 2);`
2. Файлы (общие сведения, характеристики, описание файловых переменных). Основные процедуры работы с файлами.
3. Практическое задание:  
Разработать и произвести отладку программы для решения квадратного уравнения.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 12

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Операторы `switch`, `If`, тернарный оператор (назначение и отличие). Определить значение переменной  $p$  после выполнения следующего фрагмента программы: `int m = 13; int n= 21; n= 2*m – n; if (m<=n) p= m + n; else p= 4–m*n;`
2. Алгоритм сортировки массива: метод линейного выбора.
3. Практическое задание:  
Создать и отладить приложение – конвертор перевода суммы денег из долларов в рубли.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 13

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Условный оператор if (if..., if...else). Вложенные условные операторы. Определить значение переменной p после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m – n; if (m<=n ) p= m + n; else p= 4 – m*n;`

2. Алгоритм сортировки массива: метод пузырька.

3. Практическое задание:

Разработать и произвести отладку программы для вычисления делителей натурального числа N. Вывести сами делители, их количество.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 14

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Операторы цикла for, while и do... while, блок-схемы. Какими будут значения переменных s после выполнения следующих операторов: `int s=2; int i= 1; while (i<10) {s+= i; i+= 1;}`

2. Алгоритм нахождения минимального элемента в массиве (одномерный массив).

3. Практическое задание:

Разработать и произвести отладку программы, вычисляющей сумму 1 -й и последней цифр натурального числа N. Вывести эти цифры и сумму

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 15

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Ввод/вывод одномерных массивов. Пример. Определить значение переменной s после выполнения следующих операторов: `int s=0; int n=5; for (int i=2 ; i< n; i++) { s+= 100 / i;`

`if (m<=n) p= m + n; else p= 4 – m*n`

2. Алгоритм нахождения порядкового номера максимального элемента в массиве (одномерный массив).

3. Практическое задание:

Разработать и произвести отладку программы, находящей все простые числа в заданном диапазоне.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 16

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Алгоритм сортировки массива. Какими будут значения переменных s и i после выполнения следующих операторов: `int s=20; int i= 0; while (i<=10) { s+= 5; i+= 1;}`

2. Алгоритм нахождения порядкового номера минимального элемента в массиве (одномерный массив).

3. Практическое задание:

Разработать и произвести отладку программы, находящей все нечетные числа в заданном диапазоне и их количество.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 17

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Ввод/вывод двумерных массивов. Какими будут значения переменных  $a$  и  $b$  после выполнения следующих операторов: `int a= 1; int b= 2; while (b<20) { a+= 2; b+=2;}`
2. Алгоритм нахождения максимального элемента в массиве (двумерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, находящей все четные числа в заданном диапазоне и их количество.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 18

#### ОПЦ.04. «Основы алгоритмизации и программирования»

- 1 Алгоритм нахождения максимального (минимального) элемента в массиве (одномерный массив). Алгоритм нахождения порядкового номера максимального (минимального) элемента в массиве (одномерный массив).
2. Алгоритм нахождения минимального элемента в массиве (двумерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, которая выполняет замену отрицательных чисел на 0, положительных – на 1 и выводит список элементов массива. Размер одномерного массива 10. Ввод элементов массива производится через клавиатуру

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 19

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Назначение функции. Описание и вызов функции. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a= 4*5 / 3 % 2; b=4*5/ (3 %2)`
2. Компиляция и отладка программы: Использование точек останова. Просмотр и изменение значений элементов данных.
3. Практическое задание:  
Разработать и произвести отладку программы, которая определяет минимальный и максимальный элементы массива. Размер одномерного массива 8. Ввод элементов массива производится через клавиатуру.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 20

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Подпрограммы (структура подпрограммы, схема алгоритма). Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a= 4*5 / 3 % 2; int b= 4*5 /(3 %2);`
2. Компиляция и отладка программы: Типы сообщений компилятора. Использование встроенного отладчика. Трассировка программы.
3. Практическое задание: Разработать и произвести отладку программы, которая определяет сумму всех элементов и количество положительных элементов.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 21

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Алгоритмы. Правила записи схем алгоритмов. Определить значение переменной  $s$  после выполнения следующих операторов:  $\text{int } s=0; \text{ int } n=4; \text{ for } (\text{int } i=2; i < n; i++) s+=10/i;$
2. Файлы (общие сведения, характеристики, описание файловых переменных). Основные процедуры работы с файлами.  
Размещение компонентов на форме.
3. Практическое задание:  
Разработать и произвести отладку программы для решения квадратного уравнения.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 22

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Структура программы на языке C++. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий:  $\text{int } a=15 / (16 \% 7); \text{ int } b=34 / a*5 - 29 \% 5*2;$
2. Алгоритм сортировки массива: метод линейного выбора.
3. Практическое задание:  
Создать и отладить приложение – конвертор перевода суммы денег из долларов в рубли.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 23

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Арифметические операции, приоритет операций. Выражения. Примеры. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий:  $\text{int } a=4*5 / 3 \% 2; \text{ int } b=4*5 / (3 \% 2);$
2. Алгоритм сортировки массива: метод пузырька.
3. Практическое задание:  
Разработать и произвести отладку программы для вычисления делителей натурального числа  $N$ . Вывести сами делители, их количество.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 24

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Типы данных ( $\text{integer}$ ,  $\text{double}$ ,  $\text{string}$ , массивы) – назначение, описание. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий:  $\text{int } a=15 / (16 \% 7); \text{ int } b=34 / a*5 - 29 \% 5*2;$
2. Алгоритм нахождения минимального элемента в массиве (одномерный массив).
3. Практическое задание: Разработать и произвести отладку программы:  
Разработать и произвести отладку программы, вычисляющей сумму 1-й и последней цифр натурального числа  $N$ . Вывести эти цифры и сумму.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 25

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Строковый тип данных. Основные функции работы со строками. Определите значение переменной  $p$  после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m – n; if (m<=n ) p= m + n; else p= 4 – m*n;`
2. Алгоритм нахождения порядкового номера максимального элемента в массиве (одномерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, находящей все простые числа в заданном диапазоне.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 26

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Операторы ввода/вывода. Форматы вывода данных. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a= 4*5 / 3 % 2; int b= 4*5 / (3 % 2);`
2. Алгоритм нахождения порядкового номера минимального элемента в массиве (одномерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, находящей все нечетные числа в заданном диапазоне и их количество.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 27

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Арифметические функции. Пример. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a= 15 / (16 % 7); int b=34 % a*5 – 29 % 5*2;`
2. Алгоритм нахождения максимального элемента в массиве (двумерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, находящей все четные числа в заданном диапазоне и их количество.

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 28

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Основные символы языка C++. Примеры записи комментариев. Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a= 4*5 / 3 % 2; int b= 4*5 /(3 %2);`
2. Алгоритм нахождения минимального элемента в массиве (двумерный массив).
3. Практическое задание:  
Разработать и произвести отладку программы, которая выполняет замену отрицательных чисел на 0, положительных – на 1 и выводит список элементов массива. Размер одномерного массива 10. Ввод элементов массива производится через клавиатуру

Преподаватель: \_\_\_\_\_ / \_\_\_\_\_ /

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 29

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Алгоритм нахождения количества и суммы натуральных чисел. Определить значение переменной  $s$  после выполнения следующих операторов: `int s=0; int n=5; for (int i=2; i<= n; i++) s+= 100 / i;`
2. Компиляция и отладка программы: Использование точек останова. Просмотр и изменение значений элементов данных.
3. Практическое задание:  
Разработать и произвести отладку программы, которая определяет минимальный и максимальный элементы массива. Размер одномерного массива 8. Ввод элементов массива производится через клавиатуру.

Преподаватель:

\_\_\_\_\_/

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 30

#### ОПЦ.04. «Основы алгоритмизации и программирования»

1. Переменные (назначение, описание в программе). Имена переменных (правила задания, примеры). Чему равны значения переменных  $a$  и  $b$  после выполнения последовательности действий: `int a= 15 /(16 % 7); int b=34 / a*5 – 29 % 5*2;`
2. Компиляция и отладка программы: Типы сообщений компилятора. Использование встроенного отладчика. Трассировка программы.
3. Практическое задание: разработать и произвести отладку программы, которая определяет сумму всех элементов и количество положительных элементов.

Преподаватель:

\_\_\_\_\_/

**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Ухтинский государственный технический университет»  
(УГТУ)

филиал Ухтинского государственного технического университета  
в г. Усинске  
(УФ УГТУ)  
(среднего профессионального образования)

УТВЕРЖДАЮ  
И. о. директора филиала  
О. В. Филиппова  
« 09 » 2026 г.



**КОМПЛЕКС ОЦЕНОЧНЫХ СРЕДСТВ  
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**

**ОПЦ.06 Основы алгоритмизации и программирования**  
образовательной программы  
среднего профессионального образования

По специальности 09.02.11 Разработка и управление программным  
обеспечением

г. Усинск  
2026

**МИНОБРНАУКИ РОССИИ**  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
**«Ухтинский государственный технический университет»**  
**(УГТУ)**  
**филиал Ухтинского государственного технического университета**  
**в г. Усинске**  
**(УФ УГТУ)**  
(среднего профессионального образования)

УТВЕРЖДАЮ  
И. о. директора филиала  
**О.В. Филиппова**  
«20» 20 20 г.  
  
\_\_\_\_\_  
(подпись) (И. О. Фамилия)  
«  »    20    г.  
  
\_\_\_\_\_  
(подпись) (И. О. Фамилия)  
«  »    20    г.

## РАБОЧАЯ ПРОГРАММА

Дисциплина: Основы алгоритмизации и программирования  
Индекс: ОПЦ.06  
Специальность: 09.02.11 Разработка и управление ПО  
Форма обучения: очная  
Курс (ы): 2  
Семестр (ы): 3,4

г. Усинск  
2026

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта среднего общего образования, утвержденного приказом Минпросвещения России от 24.02.2025 № 138 "Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением".

Разработчик Капембва Марине Андраниковна преподаватель СПО УФ УГТУ.

Рассмотрено на заседании ученого совета УФ УГТУ (протокол от 26.05.2026 № 06).

СОГЛАСОВАНО

Заместитель директора УФ УГТУ  
по учебной работе



О.В. Филиппова