

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»**

КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

«ПРОГРАММИРОВАНИЕ И ОСНОВЫ АЛГОРИТМИЗАЦИИ»

Специальность

27.05.01 Специальные организационно-технические системы

Специализация

Информационные технологии и программное обеспечение в специальных
организационно-технических системах

Квалификация (степень) выпускника — инженер-системотехник

Форма обучения — очная, очно-заочная

1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций, закрепленных за дисциплиной.

Контроль знаний обучающихся проводится в форме промежуточной аттестации. Промежуточная аттестация проводится в форме зачета.

Форма проведения зачета и экзамена – тестирование, письменный опрос по теоретическим вопросам.

2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной:

Описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

Описание критериев и шкалы оценивания теоретического вопроса и задачи:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

На промежуточную аттестацию (зачет) выносится тест, два теоретических вопроса. Максимально студент может набрать 6 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено», «не зачтено».

Оценка «зачтено» выставляется студенту, который набрал в сумме не менее 4 баллов (выполнил одно задание на эталонном уровне, другое – не ниже порогового, либо оба задания выполнит на продвинутом уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «не зачтено» выставляется студенту, который набрал в сумме менее 4 баллов, либо имеет к моменту проведения промежуточной аттестации несданные практические, либо лабораторные работы.

Экзамен и курсовой проект оцениваются по принятой в ФГБОУ ВО «РГРТУ» четырехбалльной системе: «неудовлетворительно», «удовлетворительно», «хорошо» и «отлично»

На промежуточную аттестацию (экзамен) выносится тест, два теоретических вопроса и две задачи. Максимально студент может набрать 15 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

Оценка «отлично» выставляется студенту, который набрал в сумме 15 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «хорошо» выставляется студенту, который набрал в сумме от 10 до 14 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «удовлетворительно» выставляется студенту, который набрал в сумме от 5 до 9 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «неудовлетворительно» выставляется студенту, который набрал в сумме менее 5 баллов или не выполнил всех предусмотренных в течение семестра практических заданий.

Сдача курсового проекта (защита) производится в форме демонстрации разработанной программы и собеседования, направленных на выявление уровня знаний студента и понимания алгоритмов программы. При определении оценки по курсовому проекту учитываются показатели работы студента в течение семестра над курсовым проектом, материалы пояснительной записки и ответы студента в ходе собеседования.

Если курсовой проект сдаётся на проверку не в срок, за 1 неделю до начала зачётной недели, то оценка снижается на 1 балл. Если работа предоставляется на проверку только на зачётной неделе, то оценка снижается на 2 балла. Дополнительно оценка снижается на 1 балл в случае, если студент ни разу не представлял преподавателю в течение семестра промежуточные результаты выполнения этапов работы.

3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
Знакомство со средой программирования. Ввод и запуск простейшей программы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен
Линейные алгоритмы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Функции пользователя	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Разветвляющиеся алгоритмы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Оператор-переключатель	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Цикл с заголовком	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Цикл с предусловием	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Цикл с постусловием	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Ввод/вывод данных	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Итерационные циклы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Зачет
Вложенные циклы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Зачет, КП
Одномерные массивы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Зачет, КП
Матрицы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Зачет, КП
Указатели	ОПК 7.1, ОПК 7.2, ОПК 7.3	Зачет, КП
Подпрограммы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Зачет, КП
Массивы и подпрограммы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Зачет, КП
Динамические массивы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Строки	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Строки и подпрограммы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП

Структуры	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Массив структур	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Бинарные файлы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Текстовые файлы	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен, КП
Рекурсия	ОПК 7.1, ОПК 7.2, ОПК 7.3	Экзамен

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация в форме зачета

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ОПК-7	Способен аргументировано выбирать и обосновывать, а также разрабатывать схемотехнические, системотехнические и аппаратно-программные решения управления сложными техническими объектами и технологическими процессами и реализовывать их на практике

ОПК-7.1 Выбирает и обосновывает схемотехнические, системотехнические и аппаратно-программные решения управления сложными техническими объектами и технологическими процессами

Типовые тестовые вопросы

1. Алгоритм это – ...

графическая схема

любая последовательность действий

конечная последовательность действий

конечная последовательность действий, приводящая к однозначному решению задачи

2. К средствам записи алгоритма не относятся:

словесная форма записи

диаграмма Эйлера-Венна

схема алгоритма

структурограмма

псевдокод

3. Что означает символ ромб в схеме алгоритма ?

Процесс

Решение

Модификация

4. При изображении каких конструкций языка программирования используется символ ромб?

Условный оператор

Оператор выбора

Оператор ввода

Оператор вывода

Циклы

5. Верно ли утверждение, что нисходящим проектированием или проектированием «сверху вниз» называется разработка алгоритма с разбиения сложной исходной задачи на отдельные более простые подзадачи, решение которых может быть представлено в общей структуре алгоритма функционально независимыми блоками.

да

нет

6. Что называется разветвляющимися вычислительными процессами?

многоальтернативные ветвления

вычислительные процессы, в которых в зависимости от тех или иных условий должны выполняться различные этапы вычислений

алгоритмы с деревьями поиска

7. Какую структуру данных выбрать в языке C++ для хранения информации об оборудовании (порядковый номер, название, вес, технические характеристики...)

массив

структура

массив структур

строка

8. Можно ли решить задачу поиска корней уравнения с помощью программирования?

да, с помощью итерационных циклов

нет

да, с помощью рекурсии

9. Может ли при реализации вложенных циклов внешний цикл закрываться раньше, чем внутренний цикл?

нет

да

10. Что означает переменная для компьютера?

буква алфавита

различные числа

область памяти

11. Алгоритм называют циклическим:

если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий

если ход его выполнения зависит от истинности тех или иных условий

если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий

12. Определите, какой тип алгоритмической структуры необходимо применить, если последовательность команд выполняется или не выполняется в зависимости от условия:

цикл

ветвление

линейный

13. Алгоритмом можно считать:

описание процесса решения квадратного уравнения

расписание уроков в школе
технический паспорт автомобиля
список класса в журнале

14. Как называется свойство алгоритма, которое означает, что данный алгоритм применим к решению целого класса задач?

понятность
определённость
результативность
массовость

Типовые вопросы открытого типа:

1. Обоснуйте выбор среды программирования для реализации программных решений на языке C++ (Dev C++, VS Code, Code Blocks, ...).
2. Что такое алгоритмизация (процесс разработки и описания алгоритма решения какой-либо задачи).
3. Что такое схема алгоритма (графический способ его представления с элементами словесного описания)?
4. Обоснуйте выбор типа цикла для решения задачи табулирования функции на языке C++ (цикл с предусловием, цикл с постусловием, цикл с параметром).
5. Обоснуйте, в каком случае программное решение требует использование подпрограмм (функций) в языке C++?
6. Обоснуйте, в каком случае законченную часть алгоритма можно реализовать в виде функции, а в каком случае в виде процедуры (функции с возвращаемым значением типа void).
7. Что такое итерационные циклы? Приведите примеры решения задач численных методов, реализуемых через итерационные циклы.
8. В чем отличия, преимущества и недостатки в последовательном и произвольном доступе к файлам?

ОПК-7.2 Практически реализует схемотехнические, системотехнические и аппаратно-программные решения управления сложными техническими объектами и технологическими процессами

Типовые тестовые вопросы

1. Как правильно подключать стандартную библиотеку в C++?

```
#include "iostream.h"  
#include <iostream.h>  
#include <iostream>  
#include 'iostream.h'
```

2. Где правильно инициализирована переменная?

```
int a  
int a;  
int a=5;  
int a=5
```

3. Что покажет код ниже?

```
char *s = "Fine";  
*s = 'N';  
cout << s << endl;
```

Nine

Fine

Ошибку

Chat Type

4. Где правильно указана переменная?

```
int num = "1";
```

```
char sym = 'a';
```

```
done = true;
```

```
var str = "Hi";
```

5. Какой из вариантов не является верным?

```
auto number = 5;
```

```
auto number;
```

```
auto int p = 5;
```

```
auto int p = '5';
```

6. Что будет выведено на экране?

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    int code = 'g';
```

```
    char letter = 103;
```

```
    std::cout << letter << " + " << code << "\n";
```

```
    return 0;
```

```
}
```

Ошибка компиляции

```
103 + g
```

```
g + 103
```

```
206
```

7. Какой из вариантов не является верным?

```
bool a = 1;
```

```
bool b = 0;
```

```
bool c = 'g';
```

```
bool d = 3.4;
```

Все верны

Все неверны

8. Что выведет данный фрагмент кода?

```
int main()
```

```
{
```

```
    const int x = 22;
```

```
    x++;
```

```
    std::cout << x;
```

```
}
```

Ошибка компиляции

```
22
```

```
23
```

```
Null
```

9. Как указать комментарий?

```
# здесь комментарий
/* здесь комментарий
/* здесь комментарий */
// здесь комментарий
/ здесь комментарий
/* здесь комментарий */
/// здесь комментарий
```

10. Что будет выведено на экран?

```
int a = 8;
int b = --a;
std::cout << a << b << "n";
```

77
87
78
88

11. Что будет выведено на экран?

```
int a = 8;
int = 7;
int c = a + 5* ++b;
std::cout << c;
```

91
104
48
43

Типовые вопросы открытого типа:

1. Что такое линейный алгоритм?
2. Какие операторы языка C++ реализуют алгоритмы с ветвлениями (if, switch, тернарный оператор, операторы безусловного перехода...)?
2. Какие бывают виды циклов в C++?
3. Какие есть примитивные типы данных в C++ (числовые, символьные, логические...)?
4. Какие есть составные (структурные) типы данных в C++ (массивы, строки, структуры, файлы,...)?
5. Почему лучше использовать динамические, а не статические массивы?
6. Какие два способа работы со строками есть в C++ (char* или классические C-строки, класс и тип cstring)?
7. В каком случае лучше реализовать бинарный файл, а в каком строковый?
8. Приведите пример задачи, которую можно решить через итерационную формулу, а можно с помощью рекурсии.
9. Каковы общие принципы организации вложенных циклов?
10. Для чего нужны указатели в C++?

ОПК-7.3 Имеет представление о современных методах и программных средствах схмотехнического, системотехнического проектирования, применяемые программно-аппаратные решения

Типовые тестовые вопросы

1. Как расшифровывается аббревиатура IDE?

интегрированная среда разработки

индивидуальный определяющий номер

параллельный интерфейс подключения накопителей

2. Что из приведенного является средой разработки на языке C++?

NetBeans

IntelliJ IDEA

Dev C++

Eclipse

Pascal ABC.NET

Visual Studio

VS Code

Code Blocks

3. Что такое компилятор

это компьютерная программа, которая переводит компьютерный код, написанный на одном языке программирования (исходный язык), на другой язык (целевой язык)

это компьютерная программа, которая преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд

4. Что такое транслятор

это компьютерная программа, которая переводит компьютерный код, написанный на одном языке программирования (исходный язык), на другой язык (целевой язык)

это компьютерная программа, которая преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд

5. Какие из сред разработки являются платными

Visual Studio Professional

Visual Studio Enterprise

Visual Studio Community

Clion

NetBeans

Dev C++

6. В каких операционных системах можно работать в среде Xcode

macOS

Windows

iOS

Linux

7. Верно ли утверждение, что главное различие между языками C и C++ состоит в том, что C - процедурный язык программирования, а C++ является объекто-ориентированным языком.

да

нет

8. Чем GUI отличается от CLI?

CLI подразумевает визуальный интерфейс для работы с программой - кнопки, переключатели и так далее. GUI подразумевает текстовый способ взаимодействия с программой - введение команд руками.

GUI подразумевает визуальный интерфейс для работы с программой - кнопки, переключатели и так далее. CLI подразумевает текстовый способ взаимодействия с программой - введение команд руками.

GUI используется в Windows и MacOS, CLI используется в Linux.

CLI используется в Windows, GUI используется в Linux и MacOS.

9. Что такое API?

Это система вывода текста в командную оболочку.

Это ядро операционной системы.

Это программный интерфейс, с его помощью программы общаются друг с другом.

Это интерфейс подключенного к компьютеру принтера.

10. Что такое лог программы?

Файл с инструкцией по ее запуску.

Файл с информацией о ходе ее работы.

Файл с информацией о текущей версии программы.

Файл с информацией об авторе и лицензионных правах на программу.

11. Чем отличается IDE и простой текстовый редактор?

IDE - это подвид текстовых редакторов, способных еще и работать с графикой.

IDE - это программа, включающая в себя не только текстовый редактор, но и другие необходимые для разработки программы.

IDE - это один из видов текстовых редакторов, как Блокнот или Word.

IDE - это графический редактор, а не текстовый.

12. Что такое кодировка текста?

Это информация о тексте - размер, шрифт и прочее.

Это указатель на таблицу символов, в которой хранится соответствие двоичного кода каждому символу.

Это язык символа - английский, японский, русский и т.д.

13. Формальные параметры делятся на:

локальные, глобальные

константы, локальные

переменные, глобальные

значения, переменные, константы

фактические, глобальные

Типовые вопросы открытого типа:

1. Какие возможности представляет современная среда программирования (текстовый редактор, отладка, компилятор и др...)?

2. Зачем нужен компилятор и как организована работа с ним в современной среде разработки?

3. Причины и типы ошибок (синтаксические ошибки и ошибки на этапе исполнения программы).

3. Что такое справка и контекстная справка?

4. Как работает автоматическая подстановка кода (Ctrl+Пробел) в современных средах программирования.

5. Правила именования идентификаторов в современных средах программирования (стиль Паскаль, стиль Camel и др...).

6. Какие есть разновидности алгоритмов сортировки данных (сортировка методом пузырька, быстра сортировка и др...)
7. Как организуется проект в C++ (модульное программирование)?
8. Что такое заголовочный файл и для чего он нужен?
9. Современные подходы к проектированию программ (декомпозиция и абстракция)
10. Что такое модель жизненного цикла программы (система процедур, правил и инструментальных средств, используемых для разработки программной системы)?

Типовые теоретические вопросы для зачета и экзамена по дисциплине

Вопросы по промежуточной аттестации 1 семестр

Знакомство со средой.

Запуск программы. Создание файла с программой, его сохранение. Набор текста простейшей программы на языке программирования. Компиляция и запуск этой программы. Окно вывода результатов.

Линейные алгоритмы.

Основные символы языка программирования. Понятие типа данных. Константы. Переменные и идентификаторы. Арифметические выражения. Стандартные функции. Оператор присваивания. Структура программы. Стил записи программ. Основные блоки схемы алгоритма.

Программирование с использованием функций пользователя.

Описание функции. Обращение к функции. Оформление схемы алгоритма для программы с использованием функции пользователя.

Условный оператор.

Синтаксис условного оператора. Логический (булев) тип. Логические операции. Составной оператор. Схема алгоритма условного варианта с обеими ветвями и с пустой ветвью «иначе».

Оператор варианта.

Синтаксис оператора варианта. Типы селектора. Метки. Альтернативная конструкция. Схема алгоритма оператора варианта.

Цикл с параметром.

Синтаксис цикла с параметром. Тип параметра цикла. Перечисляемый тип. Шаг изменения параметра цикла. Схема алгоритма цикла с параметром. Табулирование функции с помощью цикла с параметром. Расчет конечной суммы с помощью цикла с параметром. Расчет конечного произведения с помощью цикла с параметром.

Цикл с предусловием.

Синтаксис цикла с предусловием. Логическое выражение – условие продолжение цикла. Схема алгоритма цикла с предусловием. Табулирование функции с помощью цикла с предусловием.

Цикл с постусловием.

Синтаксис цикла с постусловием. Логическое выражение – условие окончания цикла. Схема алгоритма цикла с постусловием. Табулирование функции с помощью цикла с постусловием.

Организация ввода-вывода данных.

Ввод исходных данных. Стандартные функции ввода и вывода данных. Ввод числовых данных вещественных и целых типов. Ввод символьных и строковых данных. Вывод результатов. Управление выводом. Представление результатов решения.

Вопросы по промежуточной аттестации 2 семестр

Итерационные циклы.

Понятие итерационного цикла. Вычисление суммы бесконечного ряда. Типы задач с вычислением суммы бесконечного ряда. Вывод формулы коэффициента приращения последующего члена ряда от предыдущего и начальных значений. Решение задачи вычисления суммы бесконечного ряда.

Вложенные циклы.

Понятие вложенных циклов. Табулирование функции от нескольких переменных. Вложенные суммы и произведения. Формула определения числа повторений внутреннего цикла. Пример схем алгоритмов с вложенными циклами различных типов.

Одномерные массивы.

Понятие массива. Понятие одномерного массива. Описание массива. Тип индексов. Тип элементов. Ввод-вывод одномерного массива. Поиск минимума и максимума в массиве. Сортировка массив.

Двумерные массивы (матрицы).

Способы описания матриц. Ввод-вывод матриц. Сортировка матриц. Обработка строк и столбцов матриц. Квадратные матрицы. Работа с главной и побочной диагональю матриц.

Вопросы по промежуточной аттестации 3 семестр

Указатели и подпрограммы.

Описание указателей, операция адресации, применение указателей для массивов. Описание подпрограмм. Вызов подпрограмм. Формальные и фактические параметры подпрограмм. Блочный принцип организации программ. Использование подпрограмм при работе с одномерными массивами и матрицами. Использование указателей при работе с массивами. Динамические массивы.

Строки.

Символьная константа. Таблица кодов ASCII. Представление строки как массива символов. Особенности работы с символьными массивами через указатели. Стандартные функции работы со строками. Ввод-вывод строк. Использование функций пользователя при работе со строками.

Комбинированные типы данных (структуры).

Описание структуры. Вложенные структуры. Массивы структур. Ввод-вывод структур в виде таблицы.

Бинарные файлы.

Понятие бинарного файла. Определение файлового типа. Функции обработки файлов. Файлы последовательного доступа.

Текстовые файлы.

Понятие текстового файла. Функции чтения и записи данных в текстовые файлы.

Рекурсии.

Понятие рекурсии. Техника построения рекурсивных алгоритмов. Формы рекурсий: простая линейная рекурсия, параллельная рекурсия, взаимная рекурсия, рекурсии более высоких порядков. Рекурсия и итерация.

Типовые задачи к экзамену по дисциплине:

1. Протабулировать функцию от заданных значений аргумента

$$Y = \frac{x}{2} \sum_{n=1}^8 \frac{x^{n+1}}{n+2} + \frac{x+1}{3} \sum_{n=1}^7 \frac{x^{2n}}{n+3};$$

$x = 0,1(0,1)1.$

2. Протабулировать функцию от заданных значений аргумента

$$Y = \begin{cases} \sum_{n=0}^{50} \frac{1}{(2n+1)^2}, & \text{если } x < 0,5; \\ \sum_{n=0}^{10} \ln(x) \cdot \sin k(x-a), & \text{если } x \geq 0,5; \end{cases}$$

a — задается произвольно;

$$x = 0,4[0,2]1,8.$$

3. В одномерном массиве целых чисел подсчитать количество чисел, кратных 5.
4. В одномерном массиве целых чисел подсчитать количество чисел,
i. заканчивающихся на 3.
5. В произвольно заданном одномерном массиве поменять местами максимальное и минимальное числа.
6. В произвольно заданном одномерном массиве подсчитать количество отрицательных чисел и заменить их на среднее – арифметическое значение отрицательных чисел

7. В произвольно заданном одномерном массиве заменить отрицательные числа на их абсолютную величину.
8. В произвольно заданном одномерном массиве определить число с минимальной суммой цифр и вывести его адрес.
9. Составить подпрограмму определения минимума функции $y=f(x)$, заданной в дискретных точках отрезка $[a,b]$ с постоянным шагом h . Используя ее, найти минимумы функций $y_1=\sin^2x \cdot \cos x - \ln|x|$, $y_2=x^2-2x+8$.
10. Составить подпрограмму вычисления определенного интеграла по формуле прямоугольников
$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{i=0}^{n-1} f\left(a + i \frac{b-a}{n}\right)$$
 и, используя ее, вычислить
11. Составить подпрограмму вычисления k -й степени квадратной матрицы из $n \times n$ элементов, используя подпрограмму умножения двух матриц. Найти A^3 , B^2 , где A , B - произвольные квадратные матрицы.
12. Составить подпрограмму упорядочения элементов одномерного массива по убыванию их значений методом выбора максимального элемента, используя подпрограмму нахождения номера максимального элемента в последовательности чисел. Упорядочить по убыванию массивы X и Y .
13. Определить, является ли вводимая последовательность символов идентификатором.
14. Подсчитать количество сдвоенных символов "сс", "нн", "лл" во введенном тексте.
15. Разбить произвольный текст на строки определенной длины. При переносе слова предусмотреть вывод дефиса.
16. Дана символьная строка. Посчитать, сколько раз в ней встречается подстрока ABBA.
17. Найти во введенном тексте самое длинное и самое короткое слово.
18. Из заданной строки исключить все символы, входящие в нее более одного раза.
19. Организовать массив записей, содержащий информацию о результатах сдачи последней сессии вашими товарищами. Определить и дополнить данные средним баллом.
20. Организовать массив записей, содержащий информацию о месте жительства нескольких ваших товарищей. Предусмотреть возможность выдачи адреса по введенной фамилии.
21. Сформировать файл, содержащий фамилии нескольких студентов. Добавить к полученному файлу фамилии еще 2-3 студентов.
22. Записать в файл оценки (в баллах), полученные некоторым студентом на экзаменах в течение всех сессий. Добавить в начало файла оценки, полученные на вступительных экзаменах.
23. Записать в файл оценки (в баллах), полученные некоторым студентом на экзаменах в течение всех сессий, и определить средний балл.
24. Сформировать два файла. В один из них поместить фамилии пяти ваших знакомых, а в другой – номера их телефонов. Составить программу, которая по фамилии вашего знакомого определяет номер его телефона.