

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г. И. Носова»  
Многопрофильный колледж



УТВЕРЖДАЮ  
Директор  
/ С.А. Махновский  
«24» февраля 2021 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ  
ОП.09 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

**«Профессиональный учебный цикл»  
программы подготовки специалистов среднего звена  
специальности 09.02.01 Компьютерные системы и комплексы  
(базовой подготовки)**

**Форма обучения**

**очная**

Магнитогорск, 2021

Рабочая программа учебной дисциплины «Основы алгоритмизации и программирования» разработана на основе: Федерального государственного образовательного стандарта по специальности среднего профессионального образования 09.02.01 Компьютерные системы и комплексы, утвержденного приказом Министерства образования и науки Российской Федерации от «28» июля 2014 г. №849.

**Организация-разработчик:** Многопрофильный колледж ФГБОУ ВО «Магнитогорский государственный технический университет им. Г. И. Носова»

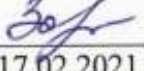
Разработчик:

преподаватель МпК ФГБОУ ВО «МГТУ им. Г.И. Носова»

  
/Людмила Александровна Фетисова

**ОДОБРЕНО**


Предметной -цикловой комиссией  
«Информатики и вычислительной  
техники»

Председатель  /И.Г.Зорина  
Протокол № 6 от 17.02.2021

Методической комиссией МпК

Протокол № 3 от 24.02.2021

Рецензент: доцент кафедры вычислительной  
Техники и программирования, к.т.н., доцент  
ФГБОУ ВО «МГТУ им. Г.И.Носова»

  
/А.Н. Калитаев

## СОДЕРЖАНИЕ

1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	4
2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	6
3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	11
4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	18
ПРИЛОЖЕНИЕ 1 .....	28
ПРИЛОЖЕНИЕ 2 .....	29
ПРИЛОЖЕНИЕ 3 .....	31
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ.....	33

# 1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «Основы алгоритмизации и программирования»

## 1.1 Область применения программы

Рабочая программа учебной дисциплины «Основы алгоритмизации и программирования» является частью программы подготовки специалистов среднего звена по специальности 09.02.01 Компьютерные системы и комплексы. Рабочая программа составлена для очной формы обучения.

## 1.2 Место дисциплины в структуре программы подготовки специалистов среднего звена

Учебная дисциплина «Основы алгоритмизации и программирования» относится к общепрофессиональному учебному циклу.

Освоению учебной дисциплины предшествует изучение учебной дисциплины ОП.05 Информационные технологии.

Дисциплина «Основы алгоритмизации и программирования» является предшествующей для изучения следующих профессиональных модулей: ПМ.02 Применение микропроцессорных систем, установка и настройка периферийного оборудования, ПМ.03 Техническое обслуживание и ремонт компьютерных систем и комплексов.

## 1.3 Цель и планируемые результаты освоения дисциплины:

Содержание дисциплины ориентировано на подготовку обучающихся к освоению профессиональных модулей программы подготовки специалистов среднего звена по специальности и овладению следующими общими и профессиональными компетенциями:

ПК 2.1. Создавать программы на языке ассемблера для микропроцессорных систем.

ПК 2.2. Производить тестирование, определение параметров и отладку микропроцессорных систем.

ПК 3.3. Принимать участие в отладке и технических испытаниях компьютерных систем и комплексов: инсталляции, конфигурировании программного обеспечения.

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и команде, обеспечивать ее сплочение, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

<i>Код ПК/ ОК</i>	<i>Умения</i>	<i>Знания</i>
ПК 2.1,	У1. формализовать поставленную задачу; У3. составлять и оформлять программы на языках	З1. общие принципы построения и использования языков программирования, их классификацию;

<p>ОК 1, ОК 4, ОК 8</p>	<p>программирования;</p> <p>У01.3. оценивать свои способности и возможности в профессиональной деятельности;</p> <p>У04.1. определять необходимые источники информации;</p> <p>У08.1. самостоятельно определять задачи профессионального и личностного развития; ставить себе образовательные цели под возникающие жизненные задачи;</p>	<p>33. процесс создания программ;</p> <p>35. общую характеристику языков ассемблера: назначение, принципы построения и использования;</p> <p>301.2. возможности применения профессиональных навыков в смежных областях;</p> <p>304.2. инструменты крупнейших цифровых экосистем для получения, обработки и анализа информации;</p> <p>308.3. основные образовательные Интернет-ресурсы, типы цифрового образовательного контента;</p>
<p>ПК 2.2,  ОК 2, ОК 5</p>	<p>У4. тестировать и отлаживать программы;</p> <p>У02.3. использовать цифровые средства и ресурсы для генерирования новых идей и решений;</p> <p>У05.2. использовать специализированное программное обеспечение;</p>	<p>32. современные интегрированные среды разработки программ;</p> <p>33. процесс создания программ;</p> <p>302.2. структуру плана для решения профессиональной задач;</p> <p>302.3. цифровые средства и ресурсы решения поставленных задач;</p> <p>305.2. специализированное программное обеспечение в профессиональной деятельности;</p>
<p>ПК 3.3,  ОК 3, ОК 6, ОК 7, ОК 9</p>	<p>У2. применять полученные знания к различным предметным областям;</p> <p>У03.2. принимать решения в нестандартной профессиональной ситуации и определять необходимые ресурсы;</p> <p>У06.1. работать в коллективе и команде;</p> <p>У07.2. выбирать оптимальные способы, приемы и методы решения профессиональных задач коллективом исполнителей;</p>	<p>34. стандарты языков программирования;</p> <p>303.2. алгоритмы принятия решения в профессиональных нестандартных ситуациях;</p> <p>306.1. основные принципы работы в коллективе;</p> <p>307.2. способы, приемы и методы решения профессиональных задач коллективом исполнителей;</p> <p>309.3. методы работы в профессиональной и смежных сферах;</p>

## 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1 Объем учебной дисциплины и виды учебной работы

<b>Вид учебной работы</b>	<b>Объем часов</b>
<b>Максимальная учебная нагрузка (всего)</b>	204
<b>Обязательная аудиторная учебная нагрузка (всего)</b>	136
в том числе:	
лекции, уроки	88
практические занятия	48
лабораторные занятия	<i>не предусмотрено</i>
курсовая работа (проект)	<i>не предусмотрено</i>
<b>Самостоятельная работа</b>	68
Форма промежуточной аттестации – <i>дифференцированный зачет</i>	

## 2.2 Тематический план и содержание учебной дисциплины «Основы алгоритмизации и программирования»

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем часов	Коды компетенций/осваиваемых элементов компетенций
1	2	3	4
<b>Раздел 1. Принципы машинной обработки данных</b>		<b>24</b>	<b>ОК/ПК</b>
<b>Тема 1.1. Основные понятия алгоритмизации. Основные алгоритмические конструкции.</b>	Содержание учебного материала	6	У2, У03.2 34, 303.2
	1. Порядок решения инженерной задачи с помощью ЭВМ. Математическая модель. Методы решения задач. Спецификация алгоритма		
	2. Понятие алгоритмизации, свойства алгоритма, общие принципы построения алгоритма Способы описания алгоритмов. Структурный подход к разработке алгоритмов. Пошаговый метод разработки алгоритма. Основы метода. Пример пошаговой разработки алгоритма и программы. Отладка и тестирование программы		
	3. Структуры алгоритмов. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические. Примеры блок-схем.	6	
	Практические занятия		
	Практические работы № 1-4 Построение блок схем основных алгоритмических конструкций.		
	Самостоятельная работа обучающихся		
<b>Тема 1.2. Методы программирования</b>	Содержание учебного материала	2	
1. Эволюция программирования. Машинно-ориентированные, высокого уровня языки программирования. Парадигма программирования. Методологии и технологии программирования.			
<b>Раздел 2. Структурное программирование</b>		<b>132</b>	<b>ОК/ПК</b>
<b>Тема 2.1. Основные элементы языка</b>	Содержание учебного материала	6	У2, У3, У4, У02.3, У05.2, У06.1, У07.2  32, 33, 34, 3.03.2, 3.02.2, 302.3, 305.2,
	1. Общие свойства: алфавит, синтаксис, семантика.		
	2. Структурная схема программы на алгоритмическом языке. Лексика языка. Переменные и константы. Типы данных. Выражения и операции.		

<b>Тема 2.2. Ввод и вывод данных</b>	Содержание учебного материала	6	3.06.1, 307.2, 309.3
	1. Консольный и потоковый ввод-вывод.		
	2. Арифметические операции и математические функции. Типичные ошибки программирования. Примеры программ.	6	
	Практические занятия		
	Практические работы №5-6 Операции ввода - вывода	8	
Самостоятельная работа обучающихся			
<b>Тема 2.3. Базовые конструкции языков программирования</b>	Содержание учебного материала	8	
	1. Оператор условия. Назначение и синтаксис. Сокращенные варианты записи.		
	2. Вложенные операторы условия. Составные логические выражения. Типичные ошибки программирования.		
	3. Оператор цикла. Назначение и синтаксис. Сокращенные варианты записи.		
	4. Вложенные операторы цикла. Типичные ошибки. Примеры решения задач.	10	
	Практические занятия		
	Практические работы №7-8 Оператор условия		
	Практическая работа №9 Оператор цикла с предусловием		
	Практическая работа №10 Оператор цикла с постусловием		
	Практическая работа №11 Оператор цикла с параметром		
Самостоятельная работа обучающихся	10		
<b>Тема 2.4. Массивы</b>	Содержание учебного материала	22	
	1. Одномерные (линейные) массивы		
	2. Многомерные массивы		
	3. Типизированные константы. Инициализация элементов массива. Типичные ошибки программирования. Решения задач.		
	4. Линейный поиск по условию. Поиск максимального и минимального элемента.		



	5. Сортировка методом "прямого выбора", методом "пузырька". Решение задач		
	Практические занятия	6	
	Практическая работа №12 Работа со строками		
	Практические работы №13-15 Алгоритмы поиска, сортировки и замены		
	Самостоятельная работа обучающихся	12	
<b>Тема 2.5. Функции</b>	Содержание учебного материала	16	
	1. Определение и объявление функций. Вызов. Формальные и фактические параметры. Глобальные переменные.		
	2. Рекурсивные функции. Типичные ошибки программирования.		
	3. Перегрузка функций. Шаблоны функций. Функция main(). Функции стандартной библиотеки. Многофайловые проекты.		
	Практические занятия	6	
	Практическая работа №16 Параметры функции		
	Практическая работа №17 Рекурсивные функции		
	Практическая работа №18 Многофайловые проекты		
	Самостоятельная работа обучающихся	12	
<b>Тема 2.6. Функции</b>	Содержание учебного материала	6	
	1. Инициализация указателей. Операции с указателями. Ссылки		
	Практические занятия	4	
	Практическая работа №19 Работа с указателями.		
	Практическая работа №20 Динамическое распределение памяти.		
<b>Раздел 3 Основы программирования на Ассемблере</b>		<b>48</b>	
<b>Тема 3.1. Основные понятия языка ассемблера</b>	Содержание учебного материала	8	У1, У3, У01.3, У04.1, У08.1
	1. Регистры процессора. Указатели команд. Флаги состояния. Сегментные регистры. Стек.		
	2. Структура программы. Основные понятия языка ассемблера		31, 33, 35, 301.2, 304.2, 308.3
<b>Тема 3.2. Директивы и операторы</b>	Содержание учебного материала	14	
	1. Директивы определения данных. Операнды команд. Пересылка и арифметические команды.		

<b>ассемблера</b>	2. Программы типа COM, EXE		
	3. Ввод и вывод. Графические видеорежимы. Работа с мышью. Последовательные и параллельный порты. Управление памятью.		
	4. Программирование на уровне порта ввода-вывода.		
	Практические занятия	10	
	Практическая работа №21 Работа и использование отладчика AFDP: Основные команды отладчика		
	Практическая работа №22 Работа и использование отладчика AFDP: Команды передачи данных		
	Практическая работа №23 Работа и использование отладчика AFDP: Арифметические команды		
	Практическая работа №24 Работа и использование отладчика AFDP: Логические операторы и команды сдвига		
	Практическая работа №25 Работа и использование отладчика AFDP: Команды передачи управления		
	Самостоятельная работа обучающихся	16	
<b>Всего (максимальная учебная нагрузка):</b>	<b>204</b>		

### 3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

#### 3.1 Материально-техническое обеспечение

Для реализации программы учебной дисциплины предусмотрены следующие специальные помещения и оснащение:

Тип и наименование специального помещения	Оснащение специального помещения
лаборатория «Программирования»	Мультимедийные средства хранения, передачи и представления информации. Учебно-методическая документация, дидактические средства Датчик цвета для микрокомпьютера NXT; Комплекты робототехнические "ПервоРобот NXT" Комплект робототехнический "LEGO" Контроллер 500995 ROBO TX Набор аккумуляторный Accu Set Наборы конструкторские 508778 ROBO TX
Помещение для самостоятельной работы обучающихся	Персональные компьютеры с пакетом MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета

#### 3.2 Учебно-методическое и информационное обеспечение реализации программы

##### Основные источники:

1. **Гуриков, С. Р.** Введение в программирование на языке Visual C# [Электронный ресурс] : учебное пособие / С. Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2019. — 447 с. - Режим доступа: <https://new.znaniium.com/read?id=338986>
2. **Колдаев, В. Д.** Основы алгоритмизации и программирования [Электронный ресурс] : учебное пособие / В. Д. Колдаев; Под ред. Л. Г. Гагариной. - Москва : ИД ФОРУМ: ИНФРА-М, 2019. - 416 с.: ил. - Режим доступа: <https://new.znaniium.com/read?id=329679>
3. **Немцова, Т. И.** Программирование на языке высокого уровня. Программирование на языке C++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 512 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - Режим доступа: <https://znaniium.com/catalog/product/1172261> ISBN 978-5-8199-0699-6

##### Дополнительные источники:

1. **Белов, В. В.** Алгоритмы и структуры данных [Электронный ресурс] : учебник / В. В. Белов, В. И. Чистякова. - Москва : КУРС, НИЦ ИНФРА-М, 2019. - 240 с. - Режим доступа: <https://new.znaniium.com/read?id=335510>
2. **Семакин, И.Г.** Основы алгоритмизации и программирования. Практикум [Электронный ресурс] : учебное пособие для студентов учреждений среднего профессионального образования / И. Г. Семакин, А.П. Шестаков. - 3-е изд., стер. - Москва : Издательский центр "Академия", 2019. - 144 с. - Режим доступа: <https://academia-moscow.ru/reader/?id=412957> - ISBN 978-5-4468-8366-0

##### Программное обеспечение и Интернет-ресурсы:

MS Windows 7 (подписка Imagine Premium)  
MS Office 2007  
7 Zip  
MS Visual Studio (подписка Imagine Premium)

##### Интернет-ресурсы

3. Федеральный центр информационно-образовательных ресурсов – ФЦИОР [Электронный ресурс]. – Режим доступа: [www.fcior.edu.ru](http://www.fcior.edu.ru), свободный. – Загл. с экрана. Яз. рус.

### 3.3 Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа является обязательной для каждого обучающегося. Самостоятельная работа может осуществляться индивидуально или группами в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений обучающихся.

Контроль результатов внеаудиторной самостоятельной работы осуществляется в пределах времени, отведенного на обязательные учебные занятия и внеаудиторную самостоятельную работу обучающихся по учебной дисциплине, проходит как в письменной, так и устной или смешанной форме, с представлением изделия или продукта самостоятельной деятельности.

В качестве форм и методов контроля внеаудиторной самостоятельной работы используются: проверка выполненной работы преподавателем, семинарские занятия, тестирование, самоотчеты, контрольные работы, защита творческих работ и др.

№	Наименование раздела/темы	Оценочные средства (задания) для самостоятельной внеаудиторной работы
1	<p>Раздел 1. Принципы машинной обработки данных. Тема 1.1.</p> <p>Основные понятия алгоритмизации и. Основные алгоритмические конструкции.</p>	<p>Текст задания</p> <p>Приоритет выполнения операций, сложные выражения.</p> <p>1. <math>Z = X^2 \cdot Y^2 + 3 \cdot X \cdot Y^2 - 5 \cdot X^2 \cdot Y + X^2 - 2 \cdot Y^2 + 4 \cdot X \cdot Y - X + Y</math>, <math>X = (2; -2)</math>, <math>Y = (4; -3)</math>.</p> <p>2. <math>B = A + 2</math>; <math>C = (A + 3)(A + 2)</math>; <math>D = (A + 4)(A + 3)</math>; <math>E = (A + 5)(A + 4)</math>; <math>A = (1; 2; -2; 3; 4)</math>.</p> <p>3. <math>Z = (X + 2) \frac{(X + 2)^2 + 3}{(X + 2)^4 + (X + 2)^2 + 3}</math>; <math>X = (0; 1; 2; -2; 4)</math>.</p> <p>4. <math>B = \sin A</math>; <math>C = \lg A</math>; <math>D = e^A</math>; <math>E =  A </math>; <math>S = (A + B)(A + B + C)(A + B + C + D)(A + B + C + D + E)</math>; <math>A = (8; -2; 4; -5)</math>.</p> <p>5. <math>B = A + 5</math>; <math>C = A - 2</math>; <math>D = B + C</math>; <math>E = A - C</math>; <math>P1 = \frac{A}{B}</math>; <math>P2 = \frac{A \cdot C}{B}</math>; <math>P3 = \frac{A \cdot C}{B \cdot D}</math>; <math>P4 = \frac{A \cdot C \cdot E}{B \cdot D}</math>; <math>A = (-15; -5; 0; 7; 14)</math>.</p> <p>Цель:</p> <ul style="list-style-type: none"> <li>– вычислить, упростив за счет использования скобочных форм и/или дополнительных переменных, значения по заданным формулам,</li> <li>– для контроля правильности результатов выполнить вычисления по формулам без использования скобочных форм и дополнительных переменных,</li> <li>– проверить результаты на комбинациях заданных значений.</li> </ul> <p>Критерии оценки:</p> <p>«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.</p> <p>– «Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.</p> <p>– «Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном</p>

		<p>сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.</p> <p>–«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>
2	<p>Раздел 2. Структурное программирование. Тема 2.2. Ввод и вывод данных  Тема 2.3. Базовые конструкции языков программирования  Тема 2.4. Массивы</p>	<p>Текст задания</p> <p>–В массиве <math>V(10)</math>, заданном начальными значениями, содержатся разные числа от 0 до 9 в произвольном порядке. Требуется поместить в массив <math>D</math> зашифрованную произвольную последовательность <math>S</math> длины <math>L \leq 30</math> из целых чисел от 0 до 9. Шифрование выполнить по следующему правилу: . Затем по данным из <math>D</math> расшифровать <math>k</math>-тую цифру и поместить в <math>R</math>.</p> <p>–Найти и сохранить в массиве <math>N</math> коэффициенты <math>p_0, p_1, p_2, p_3, p_4, p_5</math> разложения целого числа <math>K</math> (<math>0 &lt; K &lt; 10^6</math>) по степеням числа 10.</p> <p>–Выполнить циклический сдвиг элементов массива <math>X(N)</math>, <math>N \leq 20</math>, на <math>K</math> позиций, в результате которого последние <math>K</math> элементов займут место в начале массива, а остальные будут сдвинуты на <math>K</math> позиций в сторону увеличения индекса. Использовать дополнительный массив <math>D</math>.</p> <p>–На заданном отрезке, с заданным шагом изменения аргумента вычислить и поместить в массив <math>X(20)</math> значения аргумента функции <math>e^{-x} \sin(3x) - 0,2</math>, предшествующие изменению знака функции, и подсчитать их количество. Вычисления проводить либо до достижения границы интервала, либо до заполнения массива.</p> <p>–В массив <math>X(N)</math>, <math>N \leq 20</math>, упорядоченный по возрастанию значений элементов, добавить новое число так, чтобы не нарушить упорядоченность.</p> <p>Цель: Решение задач с использованием одномерных и двумерных массивов. Критерии оценки: «Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. –«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. –«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки. –«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>

3	<p>Раздел 2. Структурное программирование Тема 2.5. Функции Тема 2.6. Указатели</p>	<p>Текст задания Решение задач со строками.</p> <ul style="list-style-type: none"> <li>–Функция находит минимальный элемент массива и возвращает указатель на него. С использованием этой функции реализовать сортировку выбором.</li> <li>–Шейкер-сортировка с использованием указателей на правую и левую границы отсортированного массива и сравнения указателей.</li> <li>–Функция находит в строке пары одинаковых фрагментов и возвращает указатель на первый. С помощью функции найти все пары одинаковых фрагментов.</li> <li>–Функция находит в строке пары инвертированных фрагментов (например "123арг" и "гра321") и возвращает указатель на первый. С помощью функции найти все пары.</li> <li>–Функция производит двоичный поиск места размещения нового элемента в упорядоченном массиве и возвращает указатель на место включения нового элемента. С помощью функции реализовать сортировку вставками.</li> </ul> <p>Цель: Задание реализовать в виде функции, использующей для работы со строкой только указатели и операции над ним вида *<b>p++</b>, <b>p++</b>, <b>p[i]</b> и т.д.. Если функция возвращает строку или ее фрагмент, то это также необходимо сделать через указатель.</p> <p>Критерии оценки: «Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.</p> <p>–«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.</p> <p>–«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.</p> <p>–«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p> <p>Текст задания Практическое применение динамического программирования. Цель:</p> <ul style="list-style-type: none"> <li>– Построение матрицы с исходными данными.</li> <li>– Нахождение минимума по строкам.</li> <li>– Редукция строк.</li> <li>– Нахождение минимума по столбцам.</li> <li>– Редукция столбцов.</li> <li>– Вычисление оценок нулевых клеток.</li> </ul>
---	---	--

- Редукция матрицы.
- Если полный путь еще не найден, переходим к пункту 2, если найден к пункту 9.
- Вычисление итоговой длины пути и построение маршрута.

Для решения задачи коммивояжера методом ветвей и границ необходимо выполнить следующий **алгоритм** (последовательность действий):

#### 1. Построение матрицы с исходными данными

Сначала необходимо длины дорог соединяющих города представить в виде следующей таблицы:

Город	1	2	3	4
1	М	5	11	9
2	10	М	8	7
3	7	14	М	8
4	12	6	15	М

В нашем примере у нас 4 города и в таблице указано расстояние от каждого города к 3-м другим, в зависимости от направления движения (т.к. некоторые ж/д пути могут быть с односторонним движением и т.д.).

Расстояние от города к этому же городу обозначено буквой М. Также используется знак бесконечности. Это сделано для того, чтобы данный отрезок путь был условно принят за бесконечно длинный. Тогда не будет смысла выбрать движение от 1-ого города к 1-му, от 2-ого ко 2-му, и т.п. в качестве отрезка маршрута.

Критерии оценки:

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

–«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

–«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

–«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

4	<p>Раздел 3 Основы программирова ния на Ассемблере. Тема 3.2. Директивы и операторы ассемблера</p>	<p>Тема: «Директивы языка Ассемблера» Директива. DATA несколько более сложна. Как можно понять, директива. DATA отмечает начало сегмента данных. В этом сегменте следует размещать ваши переменные памяти. Например: .Data TopBoundary DW 100 Counter DW? ErrorMessage DB 0dh,0dh, '***Ошибка***',0dh,0ah,'\$' Это довольно просто. Вся "сложность" директивы. DATA заключается в том, что до того, как вы будете обращаться к ячейкам памяти в сегменте, определенном с помощью директивы. DATA, нужно явно загружать сегментный регистр DS идентификатором @data. Так как сегментный регистр можно загрузить из регистра общего назначения или ячейки памяти, но в него нельзя загрузить константу, регистр DS обычно загружается с помощью последовательности из двух инструкций:  mov ax,@data mov ds, ax  (Вместо регистра AX можно использовать любой общий регистр). Данная последовательность инструкций устанавливает DS таким образом, чтобы он указывал на сегмент данных, который начинается по директиве. DATA. Директивы определения данных позволяют определить переменные в памяти различного размера: DB - 1 байт DW - 2 байта = 1 слово DD - 4 байта = 1 двойное слово DQ - 8 байт = 1 "четвертное" слово DT - 10 байт .COM (англ. command) — расширение файла, использовалось в некоторых операционных системах в различных целях. В системах DOS и в 8-битной CP/M COM-файл — простой тип исполняемого файла, размер которого не может превышать 65280 байт (на 256 байт меньше размера 16-битного сегмента — 216–256 байт). COM-файлы для DOS можно выполнять также на эмуляторах, например, в среде Windows. Кроме COM-файлов DOS поддерживает файлы в формате EXE. Тип файла определяется при запуске автоматически (в формате EXE в начале файла имеется специальная сигнатура), независимо от расширения. Для запуска. COM-программы MS-DOS выделяет сегмент памяти, устанавливает на него все сегментные регистры, в первых 256 байтах строит PSP, содержимое COM-файла без изменений загружается следом за ним и запускается с первого байта (то есть с адреса 256), предварительно установив указатель стека в конец сегмента. Модель памяти, используемую COM-программами, когда код программы, все ее данные, PSP и стек расположены в одном</p>
---	--	--



	<p>сегменте, компиляторы высокоуровневых языков называют TINY (англ. tiny — крохотная).</p> <p>СОМ-программы обычно являются небольшими приложениями, системными утилитами или небольшими резидентными программами.</p> <p><b>Цель работы:</b>  ознакомление со структурой программы на ассемблере; дать понятие о директивах определения данных; директивы DATA, CODE, Mode.</p> <p><b>Критерии оценки:</b>  «Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.  –«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.  –«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.  –«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>
--	---

## 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе текущего контроля и промежуточной аттестации.

### 4.1 Текущий контроль

№	Контролируемые разделы (темы) учебной дисциплины	Контролируемые результаты (умения, знания)	Наименование оценочного средства
1	Тема 1.1. Основные понятия алгоритмизации. Основные алгоритмические конструкции.	У2, 34	<p>Тестовые задания</p> <p><b>1.</b> Алгоритм – это</p> <ol style="list-style-type: none"> <li>1. понятное и точное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи</li> <li>2. некоторые истинные высказывания, которые должны быть направлены на достижение поставленной цели</li> <li>3. отражение предметного мира с помощью знаков и сигналов, предназначенное для конкретного исполнителя</li> <li>4. инструкция по технике безопасности</li> </ol> <p><b>2.</b> Свойствами алгоритма являются</p> <ol style="list-style-type: none"> <li>1. новизна, понятность, массовость, дискретность, результативность</li> <li>2. дискретность, результативность, детерминированность, массовость, понятность</li> <li>3. массовость, понятность, условность, четкость, однозначность</li> <li>4. четкость, однозначность, массовость, дискретность, результативность</li> </ol> <p><b>3.</b> Расчлененность алгоритма на отдельные шаги, возможность выполнения которых исполнителем не вызывает сомнений, отражена в свойстве алгоритма, которое называется</p> <ol style="list-style-type: none"> <li>1. дискретностью</li> <li>2. однозначностью</li> <li>3. результативностью</li> <li>4. понятностью</li> </ol> <p><b>4.</b> Основное свойство алгоритма, характерное только для решения задач на ЭВМ, это</p> <ol style="list-style-type: none"> <li>1. дискретность</li> <li>2. понятность</li> <li>3. массовость</li> <li>4. точность</li> </ol> <p><b>5.</b> Свойство алгоритма – дискретность – обозначает</p> <ol style="list-style-type: none"> <li>1. разбиение алгоритма на конечное число простых шагов</li> <li>2. что команды должны следовать последовательно друг за другом</li> <li>3. что каждая команда должны быть описана в расчете на конкретного исполнителя</li> <li>4. нет верного ответа</li> </ol> <p><b>6.</b> Графическое задание алгоритма – это</p> <ol style="list-style-type: none"> <li>1. представление алгоритма в форме таблиц и расчетных формул</li> </ol>
2	Тема 2.2. Ввод и вывод данных	У2, У3, У4, 32, 33, 34	
3	Тема 2.3. Базовые конструкции языков программирования	У2, У3, У4, 32, 33, 34	
4	Тема 2.4. Массивы	У2, У3, У4, 32, 33, 34	
5	Тема 2.5. Функции	У2, У3, У4, 32, 33, 34	
6	Тема 3.2. Директивы и операторы ассемблера	У1, У3, 31, 33, 35	

			<ol style="list-style-type: none"> <li>2. система обозначения правил для единообразной и точной записи алгоритмов и их исполнения</li> <li>3. способ представления алгоритма с помощью геометрических фигур</li> <li>4. схематичное изображение в произвольной форме</li> </ol> <p><b>7.</b> Выбор метода решения должен предшествовать этапу</p> <ol style="list-style-type: none"> <li>1. тестирования и отладки</li> <li>2. разработки алгоритма</li> <li>3. построения математической модели</li> <li>4. анализа и уточнения результатов</li> </ol> <p><b>8.</b> Запись алгоритма в виде последовательности команд компьютера называется</p> <ol style="list-style-type: none"> <li>1. вербальной</li> <li>2. графической</li> <li>3. программной</li> <li>4. словесной</li> </ol> <p><b>9.</b> На этапе тестирования и отладки происходит</p> <ol style="list-style-type: none"> <li>1. получение результата</li> <li>2. перевод алгоритма на алгоритмический язык</li> <li>3. представление задачи в виде последовательности математических формул</li> <li>4. обнаружение и исправление синтаксических ошибок, и доведение программы до рабочего состояния</li> </ol> <p><b>10.</b>Компьютер может отследить</p> <ol style="list-style-type: none"> <li>1. синтаксические ошибки</li> <li>2. логические ошибки</li> <li>3. фактические ошибки в формулах</li> </ol> <p>любые ошибки</p> <p><b>11.</b>Завершимость алгоритма за конечное число шагов отражено в свойстве, которое называется</p> <ol style="list-style-type: none"> <li>1. однозначностью</li> <li>2. результативностью</li> <li>3. понятностью</li> <li>4. дискретностью</li> </ol> <p><b>12.</b> Прикладные программы</p> <ol style="list-style-type: none"> <li>1. программы, предназначенные для решения конкретных задач</li> <li>2. управляют работой аппаратных средств и обеспечивают услугами нас и наши прикладные комплексы</li> <li>3. игры, драйверы, трансляторы и т.д.</li> <li>4. программы, которые хранятся на различного типа дискетах</li> </ol> <p><b>13.</b>Программа</p> <ol style="list-style-type: none"> <li>1. это набор инструкций на машинном языке, который хранится в виде файла на магнитном диске и по вашей команде загружается в компьютер для выполнения</li> <li>2. это игры, предназначенные для использования на ЭВМ</li> <li>3. это набор инструкций, предназначенный для запуска компьютера</li> </ol>
--	--	--	---

4. это набор инструкций, предназначенных для работы компьютера

**Практические работы**

Практическая работа № 1-4 Построение блок схем основных алгоритмических конструкций

**Задача**

1. Какие операции определены над переменными булевого типа?
2. Записать на языке C++ следующие выражения:

а.  $\frac{x^2 + 3x - y}{a \sin x + e^y}$  б.  $\frac{ab^{-2}}{2c}$

3. Организовать ввод и вывод данных заданных типов, снабдив распечатки соответствующими заголовками  
x='\*'                      y='/'                      z=0,75

J=0.0                       $\pi$

Дать протокол программы.

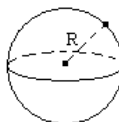
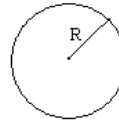
Практическая работа № 5-6 Операции ввода - вывода

**Задача**

1. Формулировка задачи

Составить алгоритм вычисления длины окружности, площади круга, площади сферы и объема шара по заданному радиусу окружности.

2. Математическая постановка задачи



Для расчета перечисленных характеристик воспользуемся формулами:

длина окружности –  $L = 2\pi R$ ;

площадь круга –  $S_{кр} = \pi R^2$ ;

площадь сферы –  $S_{сф} = 4\pi R^2$ ;

объем шара –  $V = \frac{4}{3}\pi R^3$ ,

где  $\pi$  – число Пи, математическая константа, которая выражает отношение длины окружности к её диаметру

$\pi \approx 3,141592653589793238462643...$ ,

R – радиус окружности.

3. Выбор переменных программы

Из приведенного выше решения определяем следующие переменные:

исходные данные – радиус окружности (R);

справочные данные – число  $\pi$  (Pi);

результат – длина окружности (L), площадь круга ( $S_{кр}$ ), площадь сферы ( $S_{сф}$ ) и объем шара (V).

4. Блок-схема алгоритма

5. Код программы.

Практические работы 7-11 Оператор условия. Операторы цикла

			<p><b>Задача</b></p> <p>1. Формулировка задачи</p> <p>Составить алгоритм расчета факториала <math>N!</math> с использованием различных видов цикла.</p> <p>2. Математическая постановка задачи</p> <p>Факториал числа <math>N</math> (обозначается <math>N!</math>) – произведение всех натуральных чисел до <math>N</math> включительно:</p> $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot N$ <p>По определению полагают <math>0! = 1</math>. Факториал определен только для целых неотрицательных чисел.</p> <p>3. Выбор переменных программы</p> <p>Из приведенного выше решения определяем следующие переменные:</p> <p>исходные данные – значение <math>N</math>; результат – значение <i>factorial</i>.</p> <p>Выбор переменных: аргумент и значение функции могут принимать целые неотрицательные значения.</p> <p>4. Блок-схема алгоритма.</p> <p>5. Код программы.</p> <p><b>Задача</b></p> <p>1. Формулировка задачи</p> <p>Составить алгоритм вычисления суммы ряда:</p> $y = x - \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{(-1)^{n-1} x^n}{n}$ <p>с точностью <math>e = 0.0001</math> с использованием цикла с постусловием и предусловием.</p> <p>2. Математическая постановка задачи</p> <p>Ряд представляет собой сумму слагаемых, в котором перед каждым слагаемым чередуется знак (перед первым слагаемым знак «+», перед вторым – «-», перед третьим – «+», перед четвертым – «-», и т.д.). Поэтому для его учета в алгоритм накопления суммы вводим переменную, отвечающую за знак – <math>z</math>.</p> <p>При накоплении суммы, начальное значение суммы обнуляем, т.е. <math>S=0</math>. Первое слагаемое можно представить в виде <math>\frac{x^1}{1}</math>, т.е. начальное значение степени <math>x</math> и знаменатель равны 1. Шаг изменения степени и знаменателя совпадает и равен 1. Точность вычисления определяется значением слагаемого.</p> <p>3. Выбор переменных программы</p> <p>Из приведенного выше описания определяем следующие переменные:</p> <p>исходные данные – значение <math>x</math>; начальное значение суммы <math>S</math> равно 0 (<math>S=0</math>); начальное значение степени <math>x</math> и</p>
--	--	--	---

			<p>знаменатель равны 1 (<math>i=1</math>);  знак перед первым слагаемым – «+»  (<math>z=1</math>);  результат – значение <math>S</math>.</p> <p>4. Блок-схема алгоритма.  5. Код программы.</p> <p><b>Практическая работа №12-15</b>  <i>Заполнение массива с помощью генератора случайных чисел</i></p> <p>Для генерации чисел в диапазоне <math>[A, B]</math> можно использовать следующее выражение: <math>random(B-A+1)+A</math>.</p> <p>Например, выражение <math>A[i]=random(100)</math> генерирует числа от 0 до 99, при этом 100 не входит в диапазон.</p> <p>Записать выражение для генерации чисел в диапазоне <math>[-15; 38]</math>:  <math>A[i]=random(38-(-15)+1)+(-15)</math>, т.е.  <math>A[i]=random(54)-15</math>.</p> <p><b>Задача</b></p> <p>1. Формулировка задачи  Заполнить массив <math>n</math> целыми случайными числами в диапазоне <math>[-10; 15]</math>. Составить алгоритм вычисления суммы четных чисел.</p> <p>2. Математическая постановка задачи  При накоплении суммы, начальное значение суммы обнуляем, т.е. <math>Sum=0</math>.  Выражение для генерации чисел в диапазоне <math>[-10; 15]</math>:  <math>A[i]=random(15-(-10)+1)+(-10)</math>, т.е. запишем в следующем виде <math>A[i]=random(26)-10</math>.  Для четного значения элемента выполняется условие <math>A[i] \bmod 2=0</math>.</p> <p>3. Выбор переменных программы  Из приведенного выше описания определяем следующие переменные:  исходные данные – количество элементов массива <math>n</math>;  начальное значение суммы равно 0 (<math>Sum=0</math>);  результат – значение накопленной суммы <math>Sum</math>.</p> <p>4. Блок-схема алгоритма.  5. Код программы.</p> <p><b>Задача</b></p> <p>1. Формулировка задачи  Заполнить массив <math>n</math> целыми случайными числами в диапазоне <math>[-5; 10]</math>. Составить алгоритм вычисления произведения положительных элементов.</p> <p>2. Математическая постановка задачи</p>
--	--	--	---

			<p>При накоплении произведения, начальное значение произведения равно 1, т.е. <math>pr=1</math>.</p> <p>Выражение для генерации чисел в диапазоне <math>[-5; 10]</math>:  <math>A[i]=random(10-(-5)+1)+(-5)</math>, т.е. запишем в следующем виде <math>A[i]=random(16)-5</math>.</p> <p>Для положительного значения элемента выполняется условие <math>A[i] &gt; 0</math>.</p> <p>3. Выбор переменных программы</p> <p>Из приведенного выше описания определяем следующие переменные:</p> <p>исходные данные – количество элементов массива <math>n</math>;  начальное значение произведения равно 1 (<math>pr=1</math>);  результат – значение произведения <math>pr</math>.</p> <p>4. Блок-схема алгоритма.</p> <p>5. Код программы.</p> <p><b>Задача</b></p> <p>1. Формулировка задачи:  Составить алгоритм и написать программу:  Вычислить сумму и число положительных элементов матрицы <math>A[N, N]</math>, находящихся над главной диагональю.</p> <p>2. Выбор переменных программы</p> <p>3. Блок-схема алгоритма.</p> <p>4. Код программы</p> <p>Практическая работа №16-20 Параметры функции</p> <p><b>Задача</b></p> <p>1. Формулировка задачи:  Написать функцию, выводящую в порядке возрастания элементы одномерного массива. В главной программе вызвать функцию для двух разных массивов.</p> <p>2. Выбор переменных программы</p> <p>3. Блок-схема алгоритма.</p> <p>4. Код программы</p> <p><b>Задача</b></p> <p>1. Формулировка задачи:  Написать функцию вычисления произведения прямоугольной матрицы <math>A</math> размера <math>k \times m</math> на прямоугольную матрицу <math>B</math> размера <math>m \times n</math>. В главной программе обратиться к этой функции.</p> <p>2. Выбор переменных программы</p> <p>3. Блок-схема алгоритма.</p> <p>4. Код программы</p> <p><b>Задача</b></p> <p>1. Формулировка задачи:  Ввести с клавиатуры целое число <math>N</math>. Используя рекурсию, распечатать сначала последовательность,</p>
--	--	--	--

			<p>состоящую из N букв 'A', а затем из N букв 'B'.</p> <ol style="list-style-type: none"> <li>Выбор переменных программы</li> <li>Блок-схема алгоритма.</li> <li>Код программы</li> </ol> <p><b>Контрольная работа</b></p> <p>Тема 2.3. Ввод и вывод данных</p> <ol style="list-style-type: none"> <li>Определить тип результата следующих выражений:       <ol style="list-style-type: none"> <li><math>I * I + J * J + 2 * K / 2</math></li> <li><math>x &lt; y</math></li> </ol>       если известно, что I, J, K - int; x, y - float?     </li> <li>Какие из приведенных ниже операторов присвоения являются правильными?       <ol style="list-style-type: none"> <li><math>x := y + \sin(\sin(z))</math></li> <li><math>A := (x &lt; y) \text{ or } B \text{ and } (I &gt; k)</math></li> <li><math>x := I + J - B</math></li> <li><math>I := I + K / J</math></li> </ol>       если известно, что I, J, K - int; x, y, z, A - float; D: bool?     </li> <li>Организовать ввод и вывод данных заданных типов, снабдив распечатки соответствующими заголовками        S=36    M=21425    Y=21.345        S1='*'    S2='V'        Дать протокол программы.     </li> </ol> <p>Тема 2.3. Базовые конструкции языков программирования</p> <ol style="list-style-type: none"> <li>Услуги телефонной сети оплачиваются по следующему правилу: разговоры до A минут в месяц — B руб., а разговоры сверх установленной нормы оплачиваются из расчета Сруб, за минуту. Написать программу, вычисляющую плату за пользование телефоном для введенного времени разговоров за месяц.</li> <li>Грузовой автомобиль выехал из одного города в другой со скоростью <math>v_1</math> км/ч. Через <math>t</math> ч в этом же направлении выехал легковой автомобиль со скоростью <math>v_2</math> км/ч. Составить программу, определяющую, догонит ли легковой автомобиль грузовой через <math>t_1</math> ч после своего выезда.</li> </ol> <p>Операторы цикла</p> <ol style="list-style-type: none"> <li>Даны числовой ряд и некоторое число <math>\varepsilon</math>. Найти сумму членов ряда, модуль которых больше или равен заданному <math>\varepsilon</math>. Общий член ряда имеет вид:       <math display="block">a_n = \frac{(-1)^{n-1}}{n^n}</math> </li> <li>Найти наименьший номер члена</li> </ol>
--	--	--	--



			<p>последовательности, для которого выполняется условие <math> a_n - a_{n-1}  &lt; \varepsilon</math>. Вывести этот номер и все элементы <math>a_i</math>, где <math>i=1,2,\dots,n</math>.</p> $a_n = 2 + \frac{1}{a_{n-1}}, \text{ где } a_1 = 2$ <p>Массивы</p> <ol style="list-style-type: none"> <li>1. В заданном одномерном массиве поменять местами соседние элементы, стоящие на четных местах, с элементами, стоящими на нечетных местах. Составить алгоритм и написать программу.</li> <li>2. Даны действительные числа <math>a_1, a_2, \dots, a_n</math>. Среди них есть положительные и отрицательные. Заменить нулями те числа, величина которых по модулю больше максимального числа: (<math>a_i &gt; \max\{a_1, a_2, \dots, a_n\}</math>)</li> </ol>
--	--	--	---

#### 4.2 Промежуточная аттестация

Промежуточная аттестация обучающихся осуществляется по завершении изучения дисциплины и позволяет определить качество и уровень ее освоения.

Форма промежуточной аттестации по дисциплине «Основы алгоритмизации и программирования» - дифференцированный зачет

Результаты обучения	Оценочные средства для промежуточной аттестации
<p>У1,У2,У3,У4, 31,32,33,34,35</p>	<p><b>Теоретические вопросы к диф. зачету по курсу «Основы алгоритмизации и программирования»</b></p> <ol style="list-style-type: none"> <li>1. Структура программы на языке C++.</li> <li>2. Основные типы переменных в С. Правила определения переменных и типов. Инициализация данных.</li> <li>3. Время жизни и область видимости программных объектов. Инициализация глобальных и локальных переменных</li> <li>4. Операторы языка Си. Оператор выражение, составной оператор, операторы условного перехода</li> <li>5. Оператор цикла <b>for</b></li> <li>6. Операторы цикла <b>while</b> и <b>do ... while</b></li> <li>7. Форматированный ввод-вывод.</li> <li>8. Указатели. Типизированные и нетипизированные.</li> <li>9. Работа с файлами. Файловая переменная, открытие, закрытие файла.</li> <li>10. Чтение и запись в файлы. Функция <b>eof()</b></li> <li>11. Определение и вызов функций. Фактические и формальные параметры.</li> <li>12. Определение и вызов функций. Передача массивов и указателей на функции.</li> <li>13. Бинарный поиск в упорядоченном массиве.</li> <li>14. Сортировка массива на примере одного из алгоритмов.</li> </ol>

- 15.Связанные списки: описание структуры, добавление и удаление элементов в односвязный линейный список.
- 16.Виды линейных списков: стек, очередь, дек.
- 17.Описание структуры на языке C++. Определение переменных структурного типа. Способы доступа к элементам структур.
- 18.Описание объединения на языке C++. Определение переменных типа «объединение». Способы доступа к элементам объединений.
- 19.Сущность методологии объектно-ориентированного программирования. Инкапсуляция. Наследование. Полиморфизм.
- 20.Определение класса на языке C++. Ограничение доступа к членам класса.
- 21.Дружественные функции класса.
- 22.Определение конструктора. Форматы вызова конструктора. Перегрузка конструкторов.
- 23.Наследование. Определение базового и производного классов.
- 24.Полиморфизм. Виртуальные функции.
- 25.Динамическое распределение памяти под объекты. Операторы new и delete.
- 26.Шаблоны классов.
- 27.Шаблоны функций.
- 28.Понятие потока в языке C++. Стандартные потоки.

## **Раздел 2. Структурное программирование**

### Теоретические вопросы

1. Когда целесообразно использовать консольные приложения?
2. Какой интерфейс пользователя имеют консольные приложения?
3. Что включает шаблон консольного приложения?
4. В чем назначение операторов программы?
5. Какую конструкцию программы называют блоком?
6. В чем назначение переменных программы?
7. Что такое именованная константа?
8. В каких случаях целесообразно объявлять переменные с начальными значениями?
9. Для чего предназначены директивы #include?
10. Какие виды ошибок бывают в программе?
11. Такие ошибки препятствуют созданию исполняемого файла?
12. Какими способами можно запустить консольное приложение?
13. Что понимают под отладкой программы?
14. Что такое тестирование?
15. Какие средства есть в Visual C++ для поиска смысловых ошибок в программе?
16. Что называют точкой безусловного останова?
17. Как приостановить на заданной строке программы её выполнение в отладочном режиме?
18. Какие бывают виды трассировки?
19. Какими способами можно наблюдать за значениями переменных в процессе отладки программы?
20. Как создать панель инструментов отладки или изменить набор её инструментов?
21. Для чего предназначены комментарии в программе?

	<p>22. Какие есть виды комментариев?</p> <p>23. Как можно быстро превратить часть строк программы в комментарий?</p> <p>24. Как можно создать отладочный вариант программы, используя директивы условной компиляции, чтобы после отладки можно было быстро восстановить рабочий вариант программы?</p>
--	--

**Критерии оценки зачета/дифференцированного зачета/экзамена/курсовой работы (проекта)**

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

–«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

–«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

–«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

## ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ, ВКЛЮЧАЯ АКТИВНЫЕ И ИНТЕРАКТИВНЫЕ МЕТОДЫ ОБУЧЕНИЯ

№ п/п	Название образовательной технологии (с указанием автора) / активные и интерактивные методы обучения	Цель использования образовательной технологии	Планируемый результат использования образовательной технологии	Описание порядка использования (алгоритм применения) технологии в практической профессиональной деятельности
1	Традиционная технология обучения (Я.А.Коменский и И.Ф.Герbart)	Организация усвоения учащимися знаний, умений.	На этапе объяснительного иллюстративного метода.	Формирование знаний, умений и воспроизведение усвоенного знания.
2	Информационно коммуникационная технология (М.В.Моисеева. Е.С.Полат. М.В.Бухаркина)	Обеспечение наглядности.	На протяжении урока: воспроизведение презентации.	Повышение интереса к изучаемой теме, овладение обучающимися первичными навыками работы по данной тематике, снижение уровня затруднения восприятия новой информации
3	Технология электронного обучения (Беляев М.И.)	Использование средств вычислительной техники для контроля знаний.	На заключительном этапе выдаётся домашнее задание с использованием электронного учебника	Контроль знаний, развитие навыков самоконтроля в интерактивном режиме.
4	Игровая технология (Ф. Шиллер)	Активизация познавательной деятельности, приобщение к коллективному взаимодействию для закрепления пройденного материала.	После изучения нового материала: проведение игры «Своя игра по базовым темам программирования на языке С#»	Закрепление пройденного материала, умение работать в коллективе, развитие интереса к дисциплине

### ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Разделы/темы	Темы практических/лабораторных занятий	Количество часов	Требования ФГОС СПО (уметь)
<b>Раздел 1. Принципы машинной обработки данных</b>			
Тема 1.1. Основные понятия алгоритмизации. Основные алгоритмические конструкции.	Практические работы № 1-4 Построение блок схем основных алгоритмических конструкций.	6	У1,У2
<b>Раздел 2. Структурное программирование</b>			
Тема 2.2. Ввод и вывод данных	Практические работы №5-6 Операции ввода - вывода	6	У1,У2, У3, У4
Тема 2.3. Базовые конструкции языков программирования	Практические работы №7-8 Оператор условия	10	У1,У2, У3, У4
	Практическая работа №9 Оператор цикла с предусловием		
	Практическая работа №10 Оператор цикла с постусловием		
	Практическая работа №11 Оператор цикла с параметром		
Тема 2.4. Массивы	Практическая работа №12 Поэлементные операции	6	У1,У2, У3, У4
	Практические работы №13-15 Алгоритмы поиска, сортировки и замены		
Тема 2.5. Функции	Практическая работа №16 Параметры функции	10	У1,У2, У3, У4
	Практическая работа №17 Рекурсивные функции		
	Практическая работа №18 Многофайловые проекты		
	Практическая работа №19 Работа с указателями.		
	Практическая работа №20 Динамическое распределение памяти.		
<b>Раздел 3 Основы программирования на Ассемблере</b>			
Тема 3.2. Директивы и операторы ассемблера	Практическая работа №21 Работа и использование отладчика AFDP: Основные команды отладчика	10	У1,У2, У3, У4
	Практическая работа №22 Работа и использование отладчика AFDP: Команды передачи данных		
	Практическая работа №23 Работа и использование отладчика		

	AFDP: Арифметические команды		
	Практическая работа №24 Работа и использование отладчика AFDP: Логические операторы и команды сдвига		
	Практическая работа №25 Работа и использование отладчика AFDP: Команды передачи управления		
<b>ИТОГО</b>		<b>48</b>	

## ОБРАЗОВАТЕЛЬНЫЙ МАРШРУТ

Контрольная точка	Раздел/тема	Формируемые компетенции (ОК, ПК, У, З)	Оценочные средства	
№1	Раздел 1. Принципы машинной обработки данных	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5 У1-У4, 31-34	Рубежная контрольная работа №1	1. Теоретические вопросы 2. Тестовые задания 3. Лабораторная работа
№2	Раздел 2. Структурное программирование Тема 2.3. Базовые конструкции языков программирования	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5 У1-У4, 31-34	Рубежная контрольная работа №2	1. Теоретические вопросы 2. Тестовые задания 3. Лабораторная работа
№3	Раздел 2. Структурное программирование Тема 2.4. Массивы	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5 У1-У4, 31-34	Рубежная контрольная работа №3	1. Лабораторная работа 2. Тестовые задания
№4	Раздел 2. Структурное программирование Тема 2.5. Функции	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5 У1-У4, 31-34	Рубежная контрольная работа №4	Лабораторная работа
№5	Раздел 3 Основы программирования на Ассемблере Тема 3.2. Директивы и операторы	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10	Рубежная контрольная работа №5	1. Тестовые задания 2. Лабораторная работа

	ассемблера	ПК 1.1- ПК 1.5 ПК 2.4, 2.5 У1-У5, 31-34		
№6	Допуск к зачету	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5 У1-У4, 31-34	Портфолио	1. Лабораторные работы
Промежуточная аттестация	дифференцированный зачет	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5 У1-У4, 31-34	Вопросы к диф. зачету	1 Теоретические вопросы по содержанию курса



## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ

№ п/п	Раздел рабочей программы	Краткое содержание изменения/дополнения	Дата, № протокола заседания ПЦК	Подпись председателя ПЦК