

**МИНОБРНАУКИ РОССИИ**  
**Ярославский государственный университет им. П.Г. Демидова**

Кафедра математического моделирования

УТВЕРЖДАЮ

Декан математического факультета



Нестеров П.Н.

21 мая 2024 г.

**Рабочая программа дисциплины**  
**Структуры и алгоритмы компьютерной обработки данных**

Направление подготовки (специальности)  
02.03.01 Математика и компьютерные науки

Направленность (профиль)  
«Программирование, алгоритмы и анализ данных»

Форма обучения очная

Программа рассмотрена  
на заседании кафедры  
от 12 апреля 2024 г., протокол № 8

Программа одобрена НМК  
математического факультета  
протокол № 9 от 3 мая 2024 г.

## 1. Цели освоения дисциплины

Дисциплина "Структуры и алгоритмы компьютерной обработки данных" обеспечивает приобретение знаний и умений в соответствии с государственным образовательным стандартом, содействует фундаментализации образования, формированию культуры аналитических вычислений в рамках цикла аналитических дисциплин. Целью преподавания дисциплины является ознакомление слушателей с идеями и методами современных информационных технологиях, применяемых при функционировании алгоритмов построения операционных систем.

## 2. Место дисциплины в структуре образовательной программы

Данная дисциплина относится к обязательной части образовательной программы и входит в модуль «Программирование I». Она основывается на знаниях, полученных слушателями при изучении дисциплины «Основы программирования». Знания и навыки, полученные при изучении дисциплины «Структуры и алгоритмы компьютерной обработки данных», используются обучаемыми при дальнейшем изучении прикладных математических дисциплин и дисциплин компьютерного цикла.

## 3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
<b>Профессиональные компетенции</b>		
<b>УК-1</b> Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	<b>И-УК-1.1</b> Знает принципы сбора, отбора и обобщения информации	<b>Уметь:</b> <ul style="list-style-type: none"><li>- правильно выбирать язык программирования для решения конкретной задачи</li><li>- правильно выбирать структуры для представления данных</li><li>- правильно выбирать языковые конструкции для реализации алгоритма</li></ul>
<b>УК-2</b> Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	<b>И-УК-2.2</b> Умеет определять круг задач в рамках избранных видов профессиональной деятельности, планировать собственную деятельность исходя из имеющихся ресурсов; соотносить главное и второстепенное, решать поставленные задачи в рамках избранных видов профессиональной деятельности	<b>Знать:</b> <ul style="list-style-type: none"><li>- о категориях языков и областях их применения.</li><li>- о критериях оценки языков,</li><li>- об описании синтаксиса и семантики языка,</li><li>- о месте компилятора в программном обеспечении</li></ul> <b>Владеть навыками:</b> <ul style="list-style-type: none"><li>- определения языка программирования для решения поставленной задачи;</li><li>- выбора структур для представления данных;</li><li>- оптимизации полученного кода</li></ul>

<b>Общепрофессиональные компетенции</b>		
<b>ОПК-4</b> Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	<b>И-ОПК-4.1</b> Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности	<b>Знать:</b> - основные алгоритмы обработки данных; <b>Уметь:</b> - определять их трудоемкость; <b>Владеть:</b> - опытом применения различных алгоритмов.
<b>ОПК-5</b> Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	<b>И-ОПК-5.3</b> Реализует принципы работы современных информационных технологий для решения задач профессиональной деятельности	<b>Знать:</b> типы данных, используемые в языках программирования (имена, связывание, проверка типов и области видимости), основные конструкции языков программирования
<b>ОПК-6</b> Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	<b>И-ОПК-6.1</b> Разрабатывает алгоритмы решения практических задач в области профессиональной деятельности <b>И-ОПК-6.3</b> Применяет алгоритмы и компьютерные программы для решения практических задач в области профессиональной деятельности	<b>Владеть навыками:</b> Применения методов современных информационных технологий, применяемых при функционировании алгоритмов построения операционных систем.

#### 4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет **2** зачетных единиц, **72** акад. часов.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)					Формы текущего контроля успеваемости  Форма промежуточной аттестации (по семестрам)	
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания		самостоятельная работа

1	Основные концепции языков программирования	4	2	2			4	
2	Архитектура IBM PC. Формат представления базовых данных	4	2	2		1	4	Лабораторная работа №1
3	Команды и директивы Ассемблера	4	2	2		1	4	Лабораторная работа №2
4	Ассемблер и языки высокого уровня	4	3	3			4	Лабораторная работа №3
5	Математический сопроцессор	4	3	3		1	4	Контрольная работа №1
6	Макросредства языка Ассемблера	4	2	2		1	4	Контрольная работа №2
7	Основы программирования Windows-приложений на Ассемблере	4	2	2			4	
						0,3	7,7	Зачет
	Всего		16	16		4	0,3	35,7

### Содержание разделов дисциплины:

#### Раздел 1. Основные концепции языков программирования

Парадигмы ЯП. Императивные языки. Языки функционального программирования. Декларативные языки. Объектно-ориентированные языки. Критерии оценки ЯП. Понятность. Надежность. Гибкость. Простота. Естественность. Мобильность. Стоимость. Объекты данных в ЯП. Имена. Константы. Переменные. Механизмы типизации. Статические и динамические типы данных. Слабая типизация. Строгая типизация. Производные типы. Эквивалентность типов. Наследование атрибутов. Подтипы. Анонимные типы и подтипы. Время жизни переменных. Статические, динамические, явные динамические и неявные динамические переменные. Область видимости переменных. Среда ссылок. Статическая область видимости. Динамическая область видимости. Типы данных. Элементарные типы данных: числовые типы, логические типы, символьный тип, указатели. Символьные строки. Перечислимые типы. Ограниченные типы. Векторы и массивы. Записи. Объединения. Множества. Списки. Выражения и операторы присваивания. Арифметические выражения. Логические выражения. Операторы присваивания. Структуры управления на уровне операторов. Составные операторы. Условные операторы. Циклы. Подпрограммы. Определение подпрограмм. Формальные и фактические параметры. Процедуры и функции. Методы передачи параметров. Сопрограммы.

#### Раздел 2. Архитектура IBM PC. Формат представления базовых данных

Семейство IBM PC, основные блоки. Регистры, организация оперативной памяти. Архитектура IBM PC. Регистры общего назначения, сегментные регистры, регистр указателя команд, регистр указателя стека, индексные регистры. Модели памяти, сегментная организация памяти. Формирование исполнительного адреса. Формат представления базовых данных (символы, целые числа, вещественные числа). Целые числа без знака, со знаком. Представление вещественных чисел в двоичном нормализованном виде. Машинные форматы вещественных чисел. Общая структура программы на Ассемблере. Директивы инициализации и описания данных.

#### Раздел 3. Команды и директивы Ассемблера.

Директивы SEGMENT, ASSUME, MODEL. Разница между директивами и командами Ассемблера. Описание сегментов, стандартные модели памяти. Описание процедур, внешних ссылок. Команды целочисленной арифметики. Команды пересылки и обмена информацией. Арифметические команды. Логические команды. Логическое умножение, сложение, сложение по модулю, логическое отрицание. Команды сдвига. Арифметический сдвиг, логический

сдвиг, циклический сдвиг. Команды передачи управления. Команды безусловной передачи управления, команды управления циклами. Основные принципы организации и обработки массивов. Одномерные и двумерные массивы. Команды управления состоянием микропроцессора.

#### **Раздел 4. Ассемблер и языки высокого уровня.**

Соглашения по интерфейсу. Передача параметров, возврат из процедуры. Основы организации ввода вывода информации. Функции работы с экраном. Ввод с клавиатуры. Прерывания. Встроенный ассемблер. Отладка. Ассемблер встроенный в программу. Категории ошибок. Процесс отладки. Обзор интегрированных отладчиков.

#### **Раздел 5. Математический сопроцессор.**

Типы данных. Регистры. Обычные данные, особые числа. Регистры данных, регистры состояния сопроцессора, регистры управления, регистры тегов. Система команд. Команды пересылки данных, загрузки констант, арифметические команды, команды сравнения.

#### **Раздел 6. Макросредства языка Ассемблера**

Основные понятия. Макроопределение и макрокоманда. Макрокоманда, макроопределение, макрорасширения. Директивы макроассемблера. Директивы управления листингом, Local, Rept, Irb, Irbp. Директивы условного ассемблирования. Создание библиотеки макросов.

#### **Раздел 7. Основы программирования Windows-приложений на Ассемблере.**

Современные Windows-платформы. Типы данных. Соглашения об именах. Оконные и консольные приложения.

### **5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине**

В процессе обучения используются следующие образовательные технологии:

**Академическая лекция с элементами лекции-беседы** – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Элементы лекции-беседы обеспечивают контакт преподавателя с аудиторией, что позволяет привлекать внимание студентов к наиболее важным темам дисциплины, активно вовлекать их в учебный процесс, контролировать темп изложения учебного материала в зависимости от уровня его восприятия.

**Практическое занятие** – занятие, посвященное освоению конкретных умений и навыков по закреплению полученных на лекции знаний.

**Консультации** – вид учебных занятий, являющийся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты при освоении материала дисциплины, преподаватель отвечает на вопросы студентов, которые возникают у них в процессе самостоятельной работы.

### **6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине**

В процессе осуществления образовательного процесса по дисциплине используются: для формирования материалов для текущего контроля успеваемости и проведения промежуточной аттестации, для формирования методических материалов по дисциплине:

- программы Microsoft Office;
- издательская система LaTeX;
- Adobe Acrobat Reader.

## **7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)**

В процессе осуществления образовательного процесса по дисциплине используются:

- Автоматизированная библиотечно-информационная система «БУКИ-NEXT»

[http://www.lib.uniyar.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php)

- Электронная библиотечная система «Лань» <https://e.lanbook.com>

- Электронная библиотечная система «Юрайт» <https://urait.ru>

- Электронная библиотечная система «Консультант студента»  
<https://www.studentlibrary.ru>

## **8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины**

### **а) основная литература**

1. Кауфман В.Ш. Языки программирования. Концепции и принципы. - Москва: ДМК Пресс, 2010.

<https://www.studentlibrary.ru/ru/doc/ISBN9785940746225-SCN0000/000.html>

2. Вирт Н. Построение компиляторов - Москва: ДМК Пресс, 2016.  
<https://www.studentlibrary.ru/ru/doc/ISBN9785970602195-SCN0000/000.html>

3. Ахо Альфред В. Теория синтаксического анализа, перевода и компиляции: В 2-х томах.. Т.1, Синтаксический анализ. / А.Ахо, Дж.Ульман. Пер.с англ. - М.: Мир, 1978. - 612с.

4. Юров В. И. ASSEMBLER: учебник для вузов. / В. И. Юров - СПб.: Питер, 2002. - 622 с.: ил.

### **б) дополнительная литература**

1. Лавров С.С. Основные понятия и конструкции языков программирования. М.: Финансы и статистика, 1982.

2. Хантер Р. Проектирование и конструирование компиляторов. / Р.Хантер; Пер.с англ - М.: Финансы и статистика, 1984. - 232с.

3. Буч Г. Объектно-ориентированное проектирование: с примерами применения. / Г. Буч; пер. с англ. А. А. Иванова; под ред. А. Н. Артамошкина - М.: Конкорд, 1992. - 519 с.: ил.

4. Секаев, В. Г. Основы программирования на Ассемблере : учеб. пособие / Секаев В. Г. - Новосибирск : Изд-во НГТУ, 2010. - 100 с. - ISBN 978-5-7782-1473-6. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN9785778214736.html>

5. Опалева Э. А., Самойленко В. П. Языки Программирования и методы трансляции. – СПб.,: БВХ-Петербург, 2005.

6. Пратт Т. Языки программирования. Разработка и реализация. М.: Мир, 1979.

7. Себеста Роберт У. Основные концепции языков программирования. / Роберт У.Себеста; Пер. с англ - 5-е изд. - М.: Вильямс, 2001. - 668 с.

8. Грис Д. Конструирование компиляторов для цифровых вычислительных машин - М.: Мир, 1975.

## **9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа;
- учебные аудитории для проведения практических занятий (семинаров);
- учебные аудитории для проведения групповых и индивидуальных консультаций;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

**Авторы:**

Доцент кафедры КБиММОИ, к.ф.-м.н.

Н. П. Федотова

Зав. кафедрой мат. моделирования, д.ф.-м.н., доц.

И. С. Кащенко

# Приложение №1 к рабочей программе дисциплины «Структуры и алгоритмы компьютерной обработки данных»

## Фонд оценочных средств для проведения текущей и промежуточной аттестации студентов по дисциплине

### 1. Типовые контрольные задания или иные материалы, используемые в процессе текущей аттестации

#### Лабораторная работа 1.

#### Внутреннее представление целочисленных данных в IBM PC.

##### Цель работы.

Выполнить перевод заданных преподавателем чисел из десятичной в двоичную систему счисления. Дать их внутреннее (машинное) представление в соответствии с диапазоном в знаковых и беззнаковых форматах типов **ShortInt (signed char)**, **Byte (unsigned char)**, **Integer (int)**, **Word (unsigned int)**, **LongInt (int)**. Машинное представление данных должно быть дано в двоичной и шестнадцатеричной системах счисления.

##### Порядок работы.

1. Вычислить для своего варианта целые числа.
2. Перевести их из 10-тичной в 2-ичную (или 16-ричную) систему счисления.
3. Получить их внутреннее представление.
4. Написать программу описания этих чисел на языке Ассемблера и получить листинг.
5. Проверить правильность своих выкладок.

В отчете по лабораторной работе должен быть представлен подробный протокол перевода всех заданных чисел из 10-тичной в 2-ичную и 16-ричную систему счисления.

##### Варианты.

Преподаватель задает два базовых числа  $\pm X$  и  $\pm Y$ . Студент должен прибавить и отнять от них № своего варианта.

Например, пусть  $X = \pm 6534$ ,  $Y = \pm 70$ ,  $N = 23$ . Тогда получается следующие восемь целых чисел для варианта № = 23, а именно:

- |                           |                       |
|---------------------------|-----------------------|
| 1. $6534 + 23 = 6557$ ;   | 5. $70 + 23 = 93$ ;   |
| 2. $6534 - 23 = 6511$ ;   | 6. $70 - 23 = 47$ ;   |
| 3. $-6534 + 23 = -6511$ ; | 7. $-70 + 23 = -47$ ; |
| 4. $-6534 - 23 = -6557$ ; | 8. $-70 - 23 = -93$ . |

##### Пример решения.

Протокол перевода чисел.

47	2				
46	23	2			
1	22	11	2		
	1	10	5	2	
		1	4	2	2
			1	2	1
				0	

47d → 00101111b

- 47d → 1) 00101111b – двоичный код числа |-47|

2) 11010000b – инверсия

3) + 1

-----

11010001b – дополнительный код

Остальные числа аналогично.

Машинное представление.

Dec	Byte		Word	
	Bin	Hex	Bin	Hex



47	00101111	2F	0000000000101111	002F
- 47	11010001	D1	1111111111010001	FFD1
93	01011101	5D	0000000001011101	005D
-93	10100011	A3	1111111110100011	FFA3
6511			0001000110011111	119F
-6511			1110111001100001	EE61
6557			0001000111001101	11CD
-6557			1110111000110011	EE33

## Лабораторная работа 2.

1. Дана строка, содержащая текст. Найти длину самого короткого и самого длинного слова.
2. Дана строка, содержащая текст, заканчивающийся точкой. Вывести на экран слова, содержащие не более трех различных букв.
3. Дана строка, содержащая текст. Определить сколько раз входит в нее группа букв **абс**.
4. Дана строка, содержащая текст. Определить, сколько различных символов встречается в ней. Вывести их на экран.
5. Дана строка, содержащая текст. Подсчитать самую длинную последовательность подряд идущих букв **а**.
6. Имеется строка, содержащая буквы латинского алфавита и цифры. Вывести на экран длину наибольшей последовательности цифр, идущих подряд.
7. Дана строка, содержащая текст. Найти в ней те слова, которые начинаются и заканчиваются одной и той же буквой.
8. Определить, сколько раз в строке встречается заданное слово.
9. В записке слова зашифрованы – каждое из них записано наоборот. Расшифровать сообщение.
10. Дана строка, содержащая текст. Упорядочить слова в строке по алфавиту.
11. Даны две строки А и В. Составьте программу, проверяющую, можно ли из букв, входящих в А, составить В(буквы можно использовать не более одного раза и можно переставлять). Например, А: ИНТЕГРАЛ; В: АГЕНТ – составить можно; В: ГРАФ – нельзя.
12. Строка содержит произвольный русский текст. Проверить, каких букв в нем больше: Гласных или согласных.
13. Из заданной строки выбрать те символы, которые встречаются в ней только один раз, в том порядке, в котором они встречаются в тексте.
14. Дана строка, содержащая текст на русском языке. Определить сколько раз встречается в ней самое длинное слово.
15. Дана строка, содержащая текст на русском языке. Составить в алфавитном порядке список всех слов, встречающихся в тексте.

## Лабораторная работа 3

**Трансляция различных языковых конструкций в некоторое промежуточное представление** (Обратная польская запись, Прямая польская запись, Трехадресная модель, Одноадресная модель).

### Цель работы.

Написать программу, которая читает из текстового файла некоторую языковую конструкцию (оператор присваивания, условный оператор, оператор цикла или простая программа) и выполняет необходимые преобразования исходных данных в промежуточную форму, указанную в задании. При вводе данных из текстового файла необходимо выполнить синтаксический контроль.

1. Заданное простое выражение, не содержащее вызовов функций и обращений к элементам массивов, преобразовать в обратную польскую запись. По полученной обратной польской записи построить трехадресную модель.

2. Построить график функции  $y = F(x)$  на отрезке  $[a, b]$ .  $F(x)$  вводится с клавиатуры и может содержать вызовы функций. Например,  $y = x^2 * \sin(x)$ ,  $y = 2x^2 + 5x - 4$  и т. д. Набор возможных функций, используемых в выражении ограничен.
3. Заданное простое выражение, содержащее вызовы функций преобразовать в обратную польскую запись. Например,  $y = f(I, J + g(x + y), K * L, x(I, J), 5) + 2x$ .
4. Построить обратную польскую запись для условного оператора (возможно вложение одного условного оператора в другой. Например, `if A then B else if C then K;`)
5. Заданное простое выражение, не содержащее вызовов функций и обращений к элементам массивов, преобразовать в обратную польскую запись. По полученной обратной польской записи построить одноадресную модель.
6. Заданное простое выражение, не содержащее вызовов функций и обращений к элементам массивов, преобразовать в прямую польскую запись.
7. Восстановить простое выражение по заданной его обратной польской записи.
8. Восстановить простое выражение по заданной его прямой польской записи.
9. Задана обратная польская запись некоторого простого выражения, не содержащего вхождений идентификаторов. Вычислить значение этого выражения.
10. Задана прямая польская запись некоторого простого выражения, не содержащего вхождений идентификаторов. Вычислить значение этого выражения.
11. Определить, эквивалентны ли две заданные логические формулы.
12. По заданной логической формуле построить эквивалентную логическую формулу, в которой знак отрицания встречается только перед переменными.  
Указания. Воспользоваться эквивалентными преобразованиями  

$$\text{NOT}(\text{переменная}) \rightarrow \text{переменная}$$

$$\text{NOT}(\text{формула1 OR формула2}) \rightarrow \text{NOT}(\text{формула1}) \text{ AND } \text{NOT}(\text{формула2})$$

$$\text{NOT}(\text{формула1 AND формула2}) \rightarrow \text{NOT}(\text{формула1}) \text{ OR } \text{NOT}(\text{формула2})$$
13. Пару скобок в логической формуле назовем *избыточной*, если после ее удаления формула останется эквивалентной исходной. Удалить все избыточные пары скобок в заданной формуле.
14. В заданной линейной программе удалить первый оператор каждой пары следующих друг за другом операторов в случае, если их левые части совпадают, а правая часть второго оператора не содержит вхождений переменной, совпадающей с его левой частью.
15. Вхождение оператора А линейной программы назовем избыточным, если среди предшествующих ему имеется еще одно вхождение этого оператора, причем левые части всех операторов, находящихся между двумя этими вхождениями, а также левая часть самого оператора не встречаются в правой части оператора А. Например, второе вхождение оператора  $x := y + z$  в линейной программе `«x:=y+z; u:=u-z; x:=y+z;»` избыточно. В заданной линейной программе удалить все избыточные.
16. Вхождение оператора А в линейную программу называется несущественным, если среди следующих после него операторов имеется еще одно вхождение оператора, скажем В, с той же переменной-левой частью, что и у А, причем эта переменная не встречается в правой части ни одного из операторов, находящихся между А и В. Например, Вхождение оператора  $x := u + v$  в линейную программу `«x:=u+v; u:=u*u; x:=u-u»` несущественное. Удалить вхождения всех несущественных операторов из заданной линейной программы.
17. По заданной линейной программе, в которой последний оператор содержит в левой части переменную RESULT, построить простое выражение, значение которого совпадает с тем значением, которое получает переменная RESULT после исполнения линейной программы.
18. Упростить заданную правдоподобную простую программу, заменяя всякий оператор `IF константа1 знак-отношения константа2 THEN оператор` на `оператор` в случае истинности условия или удаляя этот условный оператор в случае ложности условия.

19. Проверить правдоподобность заданной простой программы. Простая программа содержит раздел описания переменных, оператор присваивания, составной оператор, короткий условный оператор. Простая программа называется правдоподобной, если выполнены следующие условия:
  - а) каждая из переменных, встречающихся в последовательности операторов тела программы, описана, т.е. встречается в списке переменных программы, причем ровно один раз;
  - б) в программе нет больше ни одной переменной, совпадающей с названием программы;
  - в) Каждому использованию переменной текстуально предшествует хотя бы один оператор присваивания этой переменной или оператор ввода этой переменной.
20. Выполнить заданную линейную программу для заданных начальных значений всех встречающихся в ней переменных. Определить значения этих переменных после выполнения линейной программы.
21. Из-за недосмотра программиста был утерян раздел описания переменных правдоподобной простой программы (смотри задание 19). Восстановить его.
22. Реализовать упрощение логических формул относительно правил преобразования, заданных следующих схемами правил:
  - TRUE OR формула  $\rightarrow$  TRUE
  - TRUE AND формула  $\rightarrow$  формула
  - формула OR TRUE  $\rightarrow$  TRUE
  - формула AND TRUE  $\rightarrow$  формула
  - NOT FALSE  $\rightarrow$  TRUE
  - NOT TRUE  $\rightarrow$  FALSE
23. Реализовать упрощение логических формул относительно правил преобразования, заданных следующих схемами правил:
  - формула OR формула  $\rightarrow$  формула
  - формула AND формула  $\rightarrow$  формула
  - FALSE OR формула  $\rightarrow$  формула
  - формула OR FALSE  $\rightarrow$  формула
  - FALSE AND формула  $\rightarrow$  FALSE
  - формула AND FALSE  $\rightarrow$  FALSE
24. Составить таблицу истинности для заданной логической формулы.
25. Заданное простое выражение, содержащее вызовы функций и обращения к элементам массивов, преобразовать в обратную польскую запись. Например,  $y=f(A[I, J+g(x+y), K*L], x(I, J), 5)$ .
26. Что будет выведено на экран в результате выполнения правдоподобной простой программы.

Построить обратную польскую запись для оператора цикла.

### Контрольная работа №1.

#### Написать программу

1. Дан целочисленный массив A размера 10. Вывести номер первого из тех его элементов  $A[i]$ , которые удовлетворяют двойному неравенству:  $A[1] < A[i] < A[10]$ . Если таких элементов нет, то вывести 0.
2. Дан целочисленный массив размера N. Преобразовать его, прибавив к четным числам последний элемент. Первый и последний элементы массива не изменять.
3. Заменить все положительные элементы целочисленного массива размера N на значение максимального.
4. Дан массив размера 10. Переставить в обратном порядке элементы массива, расположенные между его минимальным и максимальным элементами.

5. Дан массив размера  $N$  и число  $k$  ( $0 < k < 5$ ,  $k < N$ ). Осуществить циклический сдвиг элементов массива влево на  $k$  позиций.
6. Проверить, образуют ли элементы целочисленного массива размера  $N$  геометрическую прогрессию. Если да, то вывести знаменатель прогрессии, если нет — вывести 0.
7. Дан массив ненулевых целых чисел размера  $N$ . Проверить, чередуются ли в нем четные и нечетные числа. Если чередуются, то вывести 0, если нет, то вывести номер первого элемента, нарушающего закономерность.
8. Дан массив размера  $N$ . Найти количество его локальных минимумов (справа и слева находятся элементы с большим значением).
9. Дан массив размера  $N$ . Определить количество участков, на которых его элементы монотонно возрастают.
10. Дан массив размера  $N$ . Определить количество его промежутков монотонности (то есть участков, на которых его элементы возрастают или убывают).
11. Дан массив размера  $N$ . Найти номера двух ближайших чисел из этого массива.
12. Дан целочисленный массив размера  $N$ . Удалить из массива все элементы, встречающиеся ровно два раза.
13. Дан целочисленный массив размера  $N$ . Если он является перестановкой, то есть содержит все числа от 1 до  $N$ , то вывести 0, в противном случае вывести номер первого недопустимого элемента.

## **Контрольная работа №2.**

### **Написать программу**

- Matrix1. Дана матрица размера  $M \times N$ . Поменять местами ее строки так, чтобы их минимальные элементы образовывали возрастающую последовательность.
- Matrix2. Дана матрица размера  $M \times N$ . Элемент называется локальным минимумом, если он меньше всех окружающих его элементов. Заменить все локальные минимумы данной матрицы на 0.
- Matrix3. Дана матрица размера  $N \times M$ . Перед первым столбцом, содержащим только положительные элементы, добавить столбец, состоящий из единиц.
- Matrix4. Дана матрица размера  $N \times M$ . Преобразовать матрицу, поменяв местами минимальный и максимальный элемент в каждом столбце.
- Matrix5. Дана матрица размера  $N \times M$ . Найти минимальное значение среди сумм элементов всех ее столбцов и номер столбца с этим минимальным значением.
- Matrix6. Дана матрица размера  $N \times M$ . Найти максимальный среди минимальных элементов каждой строки.
- Matrix7. Дана целочисленная матрица размера  $N \times M$ . Вывести номер ее последней строки, содержащей равное количество положительных и отрицательных элементов (нулевые элементы не учитываются). Если таких строк нет, то вывести 0.
- Matrix8. Дана матрица размера  $N \times M$ . Вывести номер ее первой строки, содержащей только положительные элементы. Если таких строк нет, то вывести 0.
- Matrix9. Дана целочисленная матрица размера  $M \times N$ . Различные строки матрицы назовем похожими, если совпадают множества чисел, встречающихся в этих строках. Найти количество строк, похожих на первую строку.
- Matrix10. Дана целочисленная матрица размера  $M \times N$ . Найти количество ее столбцов, все элементы которых различны.
- Matrix11. Дана целочисленная матрица размера  $M \times N$ . Вывести номер ее первой строки, содержащей максимальное количество одинаковых элементов.
- Matrix12. Дана квадратная матрица порядка  $M$ . Найти суммы элементов ее диагоналей, параллельных главной (начиная с одноэлементной диагонали  $A[1,M]$ ).
- Matrix13. Дана квадратная матрица порядка  $M$ . Заменить нулями элементы матрицы, лежащие ниже побочной диагонали.

- Matrix14. Дана матрица размера  $N \times M$ . Вывести количество строк, элементы которых монотонно убывают.
- Matrix15. Дана матрица размера  $N \times M$ . Найти минимальный среди элементов тех столбцов, которые упорядочены либо по возрастанию, либо по убыванию. Если такие столбцы отсутствуют, то вывести 0.
- Matrix16. Даны два числа  $k_1$  и  $k_2$  и матрица размера  $N \times M$ . Поменять местами строки матрицы с номерами  $k_1$  и  $k_2$ .
- Matrix17. Дана матрица размера  $N \times M$ . Поменять местами столбцы, содержащие минимальный и максимальный элементы матрицы.
- Matrix18. Дана матрица размера  $N \times M$ . Поменять местами столбец с номером  $k_1$  и первый из столбцов, содержащих только положительные элементы.
- Matrix19. Дано число  $k$  и матрица размера  $N \times M$ . Удалить строку матрицы с номером  $k$ .
- Matrix20. Дана матрица размера  $N \times M$ . Удалить все столбцы, содержащие только положительные элементы.
- Matrix21. Дана матрица размера  $N \times M$ . Продублировать строку матрицы, содержащую ее минимальный элемент.
- Matrix22. Дана матрица размера  $N \times M$ . После последнего столбца, содержащего только отрицательные элементы, добавить столбец, состоящий из единиц.
- Matrix23. Дана целочисленная матрица размера  $M \times N$ . Найти элемент, являющийся максимальным в своей строке и минимальным в своем столбце. Если такой элемент отсутствует, то вывести 0.
- Matrix24. Дана матрица размера  $M \times N$ . Элемент называется локальным максимумом, если он больше всех окружающих его элементов. Заменить все локальные максимумы данной матрицы на 0.
- Matrix25. Дана матрица размера  $M \times N$ . Поменять местами ее столбцы так, чтобы их максимальные элементы образовывали убывающую последовательность.

## **2. Список вопросов и (или) заданий для проведения промежуточной аттестации**

**Зачет** выставляется по итогам текущей аттестации.

## Приложение №2 к рабочей программе дисциплины «Структуры и алгоритмы компьютерной обработки данных»

### Методические указания для студентов по освоению дисциплины

Итак, вы выбрали для вашего образования математический факультет классического университета. Это означает, что в вашем дипломе в графе "Квалификация" будет присутствовать термин *математик*. Какие условия необходимы для овладения профессией математика? Отметим следующие: твёрдый характер; критическое отношение к себе; способность заниматься математикой и желание это делать; регулярные занятия математикой. Начинать нужно с работы по тем позициям, где вы сами видите свои недостатки. Однако даже в случае, когда эти условия соблюдены, в учебной работе студента могут присутствовать определённые трудности.

Одна из главных трудностей заключается в том, что студенты часто неправильно отвечают для себя на вопрос, в чём заключается понимание в математике и компьютерных науках, каков их уровень понимания, какова степень математизации их мышления. Необходимо регулярное посещение лекций и практических занятий, но и это не гарантирует хорошего понимания предмета. Для усвоения материала требуется большая и, что особенно важно, **сознательная самостоятельная работа** по теоретическим вопросам и решению задач, в частности, выполнение всех домашних заданий. Знать, помнить определения и формулировки теорем, конечно, необходимо, но это ещё не значит полностью понимать материал. Не следует заучивать математические факты так, как учат, например, стихи. Надо выработать привычку осмыслить их, проанализировать. "Чистое" знание определения без умения его применять в несложной ситуации должно быть оценено неудовлетворительно. Научитесь задавать себе разнообразные вопросы, связанные с определениями, теоремами, методами решения задач, алгоритмами. При определённой тренировке ваше понимание значительно улучшится.

Особо следует сказать о необходимости и пользе изучения математических доказательств. Сейчас доказательство изживается из школьной математики. Однако именно доказательства, а не формулировки результатов, составляют суть математики. Именно доказательный стиль мышления выделяет математика из представителей многих других профессий и именно доказательства наиболее значительны для повышения степени математизации мышления. Не следует думать, что, выслушав доказательство на лекции, вы его полностью усвоили. Попробуйте воспроизвести его самостоятельно - как правило, вы встретитесь со значительными трудностями. В этом нет ничего необычного.

По нашему мнению, даже в каждом простом на вид доказательстве закодированы те откровения, находки и открытия, которые были сделаны его автором много лет назад. И хотя они сглажены при изложении на лекции или на страницах учебника, они существуют и требуют осмысления. Каждый скачок в познании, сделанный давным-давно учёным-математиком должен иметь своё отражение в голове студента, изучающего этот предмет много лет спустя. Поэтому математика трудна не только для творчества, но и для изучения. В известном смысле изучение математики само является творчеством, только творчеством для себя. Трудность математического знания имеет и другую сторону: математические истины устойчивы, непеременимы и даже вечны. Это очень привлекательное качество нашей науки.

При самостоятельной подготовке по предмету обращайтесь внимание на исторические факты, касающиеся дисциплины, на роль основоположников и творцов науки. Обдумывайте вопросы, связанные с историей данной области науки и становлением ее идеологии.

Необходимо работать по развитию умения читать специальную литературу по предмету, разбираться в смысловых элементах структурированного математического

текста (*теорема, лемма, следствие, доказательство, пример, замечание, алгоритм, компьютерная программа* и др.).

Обязательными является серьезная самостоятельная практическая работа по дисциплине (решение упражнений и задач, создание алгоритмов и их компьютерная реализация и др.) При решении упражнений двигайтесь от простого к сложному в соответствии с требованиями преподавателя.

Математику нельзя изучить созерцательно. Лишь активная самостоятельная работа, являясь трудной составляющей образовательного процесса, позволяет усваивать математические факты.