

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего профессионального образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

Информатика

Методические указания к выполнению лабораторных работ
(для очной формы обучения)

Часть 2

Составители: к.т.н., доц. Решетов Л.А., М.В.Соколовская

Рецензент: к.т.н., доц. В.П.Попов

Приведены методические указания к выполнению лабораторных работ по дисциплинам «Информатика», «Информационные технологии», «Программирование». Дан краткий теоретический материал по тематике лабораторных работ.

Могут быть полезны для студентов по другим дисциплинам. Приведенные материалы соответствуют ФГОС и рабочим программам указанных дисциплин.

Подготовлены к публикации кафедрой «Прикладная математика» и рекомендованы к изданию редакционно-издательским советом Санкт-Петербургского государственного университета аэрокосмического приборостроения.

ВВЕДЕНИЕ

Целью проведения лабораторных работ является получение начальных практических навыков алгоритмизации и структурного программирования на языках C/C++, необходимых студентам для разработки и проектирования и освоения специальности.

В процессе изучения дисциплины студенты знакомятся с основными понятиями построения программы; описания типов и ввода-вывода данных; операций и операторов, реализующих последовательные, ветвящиеся и циклические алгоритмы; принципа модульного программирования по организации ветвящихся и циклических алгоритмов. Показаны способы решения типовых задач с целью выработки у студентов практических навыков по составлению и отладке программ при выполнении лабораторных работ.

Задачами лабораторных работ является изучение:

- основных понятий информатики;
- структуры и принципов работы ЭВМ как технического средства реализации информационных технологий;
- средств и алгоритмов представления, хранения и обработки информации;
- организации и средств человеко-машинного интерфейса.

Для успешного прохождения данных лабораторных работ по «Информатике» студент должен быть подготовлен по следующим дисциплинам:

- «Информатика» в пределах общеобразовательной программы средней школы;
- «Математика» в пределах общеобразовательной программы средней школы;
- «Дискретная математика», в разделах математической логики.

Для более углубленного усвоения материала курса и лабораторных работ, в конце указан список литературы.

Лабораторная работа №1

БАЗОВЫЕ НАВЫКИ РАБОТЫ В СИ

Цель работы: получить общее представление о составлении программы, трассировке и тестировании.

Задание на лабораторную работу

Написать программу вычисления выражения. Вывести промежуточные и итоговое значения на экран. Провести письменно проверку вычисления.

Вариант 1.

$$\frac{\frac{13,75 + 9\frac{1}{6}}{1,2} + \frac{6,8 - 3\frac{3}{5} \cdot 5\frac{5}{6}}{27\frac{1}{6}}}{\frac{10,3 - 8\frac{1}{2} \cdot \frac{5}{9}}{3\frac{2}{3} - 3\frac{1}{6} \cdot 5,6}}$$

Вариант 2.

$$\frac{\frac{3\frac{1}{3} + 2,5}{2,5 - 1\frac{1}{3}} \cdot \frac{4,6 - 2\frac{1}{3}}{4,6 + 2\frac{1}{3}} \cdot 5,2 \cdot \frac{0,05}{\frac{1}{7} - 0,125} + 5,7}{\frac{0,05}{\frac{1}{7} - 0,125} + 5,7}$$

Вариант 3.

$$\frac{5\frac{4}{45} - 4\frac{1}{6} : 5\frac{8}{15}}{4\frac{2}{3} + 0,75} \cdot 3\frac{9}{13} \cdot 34\frac{2}{7} + \frac{0,3 : 0,01}{70} + \frac{2}{7}$$

Вариант 4.

$$\frac{\frac{3}{5} + 0,425 - 0,005}{30,5 + \frac{1}{6} + 3\frac{1}{3}} : 0,1 + \frac{6\frac{3}{4} + 5\frac{1}{2}}{26 : 3\frac{5}{7}} - 0,05$$

Вариант 5.

$$\frac{26\frac{2}{3} : 6,4 \cdot 19,2 : 3\frac{5}{9} - \frac{8\frac{4}{7} : 2\frac{26}{777}}{0,5 : 18\frac{2}{3} \cdot 11} - \frac{1}{18}}$$

Вариант 6.

$$\frac{3\frac{1}{3} \cdot 1,9 + 19,5 : 4\frac{1}{2}}{\frac{62}{75} - 0,16} : \frac{3,5 + 4\frac{2}{3} + 2\frac{2}{15}}{0,5 \cdot 1\frac{1}{20} + 4,1}$$

Вариант 7.

$$\frac{\frac{1}{5} : \frac{17}{40} + 0,6 - 0,005}{\frac{5}{6} + 1\frac{1}{3} - 1\frac{23}{30}} \cdot 1,7 + \frac{4,75 + 7\frac{1}{2}}{33 : 4\frac{5}{7}} : 0,25$$

Вариант 8.

$$\frac{(3,2 - 1,7) : 0,003}{\frac{29}{35} - \frac{3}{7} \cdot 4 : 0,2} - \frac{1\frac{13}{20} - 1,5 \cdot 1,5}{2,44 + 1\frac{14}{25} \cdot \frac{1}{8}} : 62\frac{1}{20} + 1,364 : 0,1$$

Вариант 9.

$$\frac{(3,4 - 1,275) \cdot \frac{16}{17}}{\frac{5}{18} \cdot 1\frac{7}{85} + 6\frac{2}{17}} + 0,5 \cdot 2 + \frac{12,5}{5,75 + \frac{1}{2}}$$

Вариант 10.

$$\frac{1\frac{1}{7} - \frac{23}{49}}{0,6 : 3\frac{3}{4} \cdot 2\frac{1}{2} + 3,75 : 1\frac{1}{2}} : 2,2$$

Вариант 11.

$$(520 \cdot 0,43) : 0,26 - 217 \cdot 2\frac{3}{7} - 31,5 : 12\frac{3}{5} + 114 \cdot 2\frac{1}{3} + 61\frac{1}{2}$$

Вариант 12.

$$\frac{4,625 - \frac{13}{18} \cdot \frac{9}{26} : \frac{9}{4} + 2,5 : 1,25 : 6,75}{\frac{1}{2} - 0,375} : 0,125 + \frac{5}{6} - \frac{7}{12} : (0,358 - 1,4796 : 13,7)$$

Вариант 13.

$$5\frac{4}{7} : 8,4 \cdot \frac{6}{7} \cdot 6 - \frac{(2,3 + 5 : 6,25) \cdot 7}{8 \cdot 0,0125 + 6,9} - 20,384 : 1,3 \cdot 7$$

Вариант 14.

$$\frac{2\frac{3}{4} : 1,1 + 3\frac{1}{3}}{2,5 - 0,4 \cdot 3\frac{1}{3}} : \frac{5}{7} - \frac{2\frac{1}{6} + 4,5 \cdot 0,375}{2,75 - 1\frac{1}{2}}$$

Вариант 15.

$$\frac{3,75 + 2\frac{1}{2}}{2\frac{1}{2} + 1,875} - \frac{2\frac{3}{4} + 1,5}{2,75 - 1\frac{1}{2}} \cdot \frac{10}{11}$$

Вариант 16.

$$\frac{3\frac{7}{12} - 2\frac{11}{18} + 2\frac{1}{24} \cdot 1\frac{5}{31} - \frac{3}{52} \cdot 3\frac{1}{2} + \frac{5}{6} \cdot 1\frac{7}{13}}{\frac{19}{84} \cdot 5\frac{13}{42} - 2\frac{13}{28} - \frac{5}{24} + 1\frac{2}{27} - \frac{1}{3} \cdot \frac{4}{9}}$$

Вариант 17.

$$\frac{(2,7 - 0,8) \cdot 2\frac{1}{3}}{(5,2 - 1,4) : \frac{3}{70}} + 0,125 : 2\frac{1}{2} + 0,13$$

Вариант 18.

$$\frac{\frac{1}{6} + 0,1 + \frac{1}{15} : \frac{1}{6} + 0,1 - \frac{1}{15} \cdot 2,52}{0,5 - \frac{1}{3} + 0,25 - \frac{1}{5} : 0,25 - \frac{1}{6} \cdot \frac{7}{13}}$$

Вариант 19.

$$\frac{1,88 + 2\frac{3}{25} \cdot \frac{3}{16}}{0,625 - \frac{13}{18} : \frac{26}{9}} + \frac{\frac{0,216}{0,15} + 0,56}{7,7 : 24\frac{3}{4} + \frac{2}{15}} : 0,5 \cdot 4,5$$

Вариант 20.

$$\frac{0,128 : 3,2 + 0,86 \cdot \left(1 \frac{32}{63} - \frac{13}{21}\right) \cdot 3,6}{\frac{5}{6} \cdot 1,2 + 0,8} - \frac{0,505 \cdot \frac{2}{5} - 0,002}{}$$

Вариант 21.

$$\frac{3,75 : 1 \frac{1}{2} + \left(1,5 : 3 \frac{3}{4}\right) \cdot 2 \frac{1}{2} + \left(1 \frac{1}{7} - \frac{23}{49}\right) : \frac{22}{147}}{2 : 3 \frac{1}{5} + \left(3 \frac{1}{4} : 13\right) : \frac{2}{3} - \left(2 \frac{5}{18} - \frac{17}{36}\right) \cdot \frac{18}{65}}$$

Вариант 22.

$$\frac{0,5 + \frac{1}{4} + 0,1(6) + 0,125 \cdot (3,75 - 0,625) \cdot \frac{48}{125}}{0, (3) + 0,4 + \frac{14}{15}} + \frac{12,8 \cdot 0,25}{}$$

Вариант 23.

$$\frac{3 \frac{1}{3} : 10 + 0,175 : 0,35 \cdot \left(\frac{11}{18} - \frac{1}{15}\right) : 1,4}{1,75 - 1 \frac{11}{17} \cdot \frac{51}{56}} - \frac{\left(0,5 - \frac{1}{9}\right) \cdot 3}{}$$

Вариант 24.

$$\frac{0,125 : 0,25 + 1 \frac{9}{16} : 2,5}{(10 - 22 : 2,3) \cdot 0,46 + 1,6} + \left(\frac{17}{20} + 1,9\right) \cdot 0,5$$

Вариант 25.

$$\left(26 \frac{2}{3} : 6,4\right) \cdot \left(19,2 : 3 \frac{5}{9}\right) - \frac{8 \frac{4}{7} : 2 \frac{26}{77}}{0,5 : 18 \frac{2}{3} \cdot 11} - \frac{1}{18}$$

Порядок выполнения работы

1. Открыть одно из приложений и создать проект (под своей фамилией).

2. Написать программу вычисления по варианту, используя вспомогательные переменные (не менее 4).

3. Проверить выходные данные ручным счетом.

4. Предъявить программу и расчеты преподавателю.

Письменный отчет НЕ оформляется.

Теоретическая часть

Средства программирования вычислительных процессов

Для создания программ используются системы программирования (или инструментальные среды). В системы программирования на языках C/C++ входят:

- интегрированная среда программирования,
- компилятор исходного текста программы,
- редактор связей,
- библиотеки заголовочных файлов,
- библиотеки функций,
- программы-утилиты.

Структура программы на C/C++

Программа на языке высокого уровня содержит формализованный текст, включающий описательные и исполняемые предложения. Программа на языках C/C++ представляет собой набор функций. Одна из функций является обязательной и за ней закреплено имя **main()**.

Структура программы:

```
тип main(параметры)    /*имя функции, указывающий точку входа*/  
{                          /*блок инструкций*/  
инструкция 1;  
...  
инструкция K;  
return (значение);
```

}

Директивы

Программа содержит директивы препроцессора — указания для предварительной обработки текста исходного файла перед его компиляцией. Директивой является строка, начинающаяся с символа «#».

Базовые элементы языков C/C++

В зависимости от назначения и использования в тексте программы множество базовых символов можно разделить на несколько подгрупп.

1. Буквы латинского алфавита — заглавные и строчные (A...Z , a...z).
2. Десятичные цифры от 0 до 9.
3. Знаки операций «+», «-», «*», «/», «<», «>», «=».
4. Скобки «(», «)», «[», «]», «{», «}».
5. Знаки пунктуации (разделители) «,», «.», «;», «:», «?», «_» (нижнее подчёркивание), « » (пробел).
6. Спецсимволы «#», «\$», «!», «%», «|», «^», «&», «~», «\», «'» (апостроф), «"» (кавычки).

Составные символы образуются в виде комбинаций базовых символов без пробелов, представляя собой разделители или знаки операции:

«<=», «>=», «!=» (не равно), «||» (логическое ИЛИ), «&&» (логическое И), «++», «--», «<<», «>>» и др.

Комментарий — это текст, для пояснений в программе, заключённый между символами «/*» и «*/», в котором могут использоваться буквы русского алфавита (кириллица). В языке C++ можно использовать однострочный комментарий после символов «//».

Идентификатор — это совокупность букв, цифр и знака подчёркивания, задающая имя переменной, название функции или ключевое слово в программе.

Ключевые слова — это predetermined идентификаторы языка, используемые для составления текста программы, например:

asm	default	for	pascal	switch
auto	do	goto	register	typedef
break	double	huge	return	union
case	else	if	short	unsigned
cdecl	enum	int	signed	void

char	extern	interrupt	sizeof	volatile
const	far	static	while	
	long			
continue	float	near	struct	

Лексема — это синтаксически *неделимая* единица текста программы, имеющая определенный смысл для компилятора, от 1 до нескольких символов алфавита, например, знаки пунктуации и операции, скобки, составные символы, а также ключевые слова, константы и идентификаторы (имена) переменных и функций. Границами лексем служат пробельные символы или другие лексемы (например, знаки пунктуации, операций, скобок).

Данные и их типы

К *данным* относится любая информация, представленная в таком виде, который позволяет автоматизировать её сбор, хранение и обработку в ЭВМ (числа, символы, биты и др.). Данные в программе могут быть *исходными* (задаваемыми на входе программы) или *результатами* обработки (промежуточными или выходными).

Константы

Константы задают *неизменяемые* величины в программе, определяемые их типом и формой представления (число, символ, строка, указатель (адрес)).

Числовая константа — это целое или вещественное число.

Символьная константа — это символ, заключённый в апострофы, например, 'F', '5', ' ' (пробел), '/'. Значением символьной константы является числовой код символа согласно ASCII-таблице ('\t' — табуляция, '\n' — перевод на новую строку).

Переменные

Переменные представляют в программе данные, значение которых может *меняться* в ходе работы программы, и определяются своим описанием до *первого* использования следующим образом

тип имя [=инициатор], ...;

тип — *ключевое* слово, определяющее объём памяти (число байтов), выделенный для хранения значения переменной (как объекта программы):

char — символьный — -128...127 (1 байт),

int — целый — -32768...32767 (2 байта),

float — вещественный — $3.4 \cdot E^{-38} \dots 3.4 \cdot E^{+38}$ (4 байта),

double — вещественный — $1.7 \cdot E^{-308} \dots 1.7 \cdot E^{+308}$ (8 байт);

имя — уникальный *идентификатор* переменной, задающий символический адрес объекта программы в памяти ЭВМ;

инициатор — *начальное* значение переменной, которое может отсутствовать в описании.

Примеры описания переменных:

int k, max=0; **float** r_2, p1=0.5; **char** sim='x'.

Ввод-вывод данных

Для того чтобы данные попали в память ЭВМ, пользователь должен их ввести, а для получения результатов работы программы должен уметь вывести их в приемлемом для человека виде. В языках C/C++ отсутствуют встроенные операторы ввода-вывода. Ввод-вывод рассматривается как поток данных, управляемый с помощью функций ввода-вывода

Любая программа на C/C++ использует функции из стандартных библиотек, для чего их заголовочные файлы необходимо подключить к программе с помощью директив препроцессора, например:

```
#include <stdio.h> /* подключение функций стандартного ввода-вывода*/
```

```
#include <conio.h> /* подключение функций консольного ввода-вывода*/
```

Под *консолью* пользователя понимается *клавиатура* как стандартное устройство *ввода* и *экран* дисплея (стандартное устройство *вывода*). На эти устройства можно ссылаться с помощью predefinedных в файле **stdio.h** указателей. *Стандартный* ввод-вывод может быть двух видов: форматированный и не форматированный.

Форматированный ввод

Эта операция выполняется *функцией* *ввода*, имеющей вид:

```
scanf("форматная строка", &аргумент, ... )
```

форматная строка — это список *спецификаций* форматов вводимых данных, разделённых *запятыми*;

&аргумент — адрес переменной (аргумента) в оперативной памяти, по которому будет записано значение переменной при вводе с клавиатуры.

Спецификация формата аргумента начинается с символа % (процент) и в простом случае должна содержать символ типа аргумента по форме: % тип.

Символ типа определяет, как интерпретируется компилятором вводимое значение аргумента:

d, i — десятичное целое со знаком;

o — восьмеричное число;

x — шестнадцатеричное число;

f, e — вещественное число с фиксированной или плавающей точкой;

c — ввод одного символа;

s — ввод строки символов, пока не встретится символ первого пробела

Пример. Функция ввода данных может иметь вид:

```
scanf (“%d%f%c” ,&m, &p1, &sim);
```

При вводе с клавиатуры *числовые* данные должны *разделяться* пробельным символом (путем нажатия клавиш *<пробел>* или *<Enter>*). *Символьные* данные *не разделяются*, поскольку пробельные символы прочитываются как *вводимые* символы.

Форматированный вывод

Для вывода данных из компьютера на экран дисплея используется *функция вывода*:

```
printf(“форматная строка”, аргумент1, ...)
```

форматная строка может содержать компоненты трех типов:

➤ *обычные символы*, которые можно использовать для пояснения при выводе;

➤ *спецификации форматов данных*, задаваемых в списке выводимых переменных;

➤ *управляющие символные константы* (**\n** — новая строка; **\b** — возврат на шаг; **\t** — горизонтальная табуляция и т.д.);

аргумент1, аргумент2,... — список переменных или выражений, значения которых выводятся в соответствии с заданными спецификациями форматов.

При выводе *вещественных* чисел (`%f`) компилятор выводит *все* цифры дробной части числа. Для вывода числа с заданной точностью необходимо уточнить формат вывода в виде: `%m.nf`, где *m* — количество *всех* символов (знак, все цифры, разделитель точка), а *n* — количество цифр *дробной части*.

Неформатированный ввод-вывод

Для ввода-вывода символов и строк можно использовать соответствующие функции стандартного ввода-вывода (**stdio.h**).

Функция ввода символов (без параметров):

getchar()

Функция вывода символа:

putchar(ch).

В C++ для поддержки ввода-вывода данных используется целая иерархия классов. Двухнаправленный поток поддерживается классом **iostream**, объявление которого помещено в заголовочный файл, подключаемый к программе строкой:

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

Стандартный ввод-вывод C++ поддерживает два потоковых объекта для ввода/вывода:

cin — стандартный ввод, как правило, с клавиатуры (аналог `stdin` в C).

cout — стандартный вывод, как правило, на экран (аналог `stdout` в C).

Для каждого потока переопределены два оператора:

`>>` — оператор извлечения (чтения) из потока (для ввода с клавиатуры: "считать из");

`<<` — оператор записи (вставки) в поток (для вывода на экран: "вывести на").

Функции ввода/вывода способны воспринимать аргументы любых основных типов данных (`char`, `char*(string)`, `int`, `long`, `float`, `double`).

Лабораторная работа № 2

ИСПОЛЬЗОВАНИЕ УСЛОВНОГО ОПЕРАТОРА

Цель работы: изучить введение базовых математических функции и структуру «Если».

Задание и порядок выполнения лабораторной работы

Ввести с клавиатуры вещественные числа (исходные данные). Используя логическую функцию вычислить по варианту значение M . Вывести на экран промежуточные результаты.

Вариант 1.

Исходные данные: A, B, C . Вычислить

$$M = \begin{cases} e^{B+C}, & \text{если } \max(A, B, C) > 0 \\ \min(A, B, C), & \text{иначе} \end{cases}$$

Вариант 2.

Исходные данные: A, B, C, D . Вычислить

$$M = \begin{cases} \min(A, B, C) + \pi * D, & \text{если } D < 0 \\ \sqrt{\cos(A)}, & \text{иначе} \end{cases}$$

Вариант 3.

Исходные данные: A, B, C, D . Вычислить

$$M = \max(\max(A, B), \min(A, C, \cos(D))).$$

Вариант 4.

Исходные данные: A, B, C . Вычислить

$$M = \begin{cases} \max(A + \cos(B), C), & \text{если } A + C \neq 0 \\ \min(A + C, 3, B), & \text{иначе} \end{cases}$$

Вариант 5.

Исходные данные: A, B, C . Вычислить

$$M = \begin{cases} \sin(B * \pi), & \text{если } B - \min(A, C, \cos(B)) > 0 \\ \sqrt{C} - 0.1, & \text{иначе} \end{cases}$$

Вариант 6.

Исходные данные: A, B . Вычислить

$$M = \begin{cases} \cos(4\pi \cdot \min(A, B, 1)), & \text{если } A \neq B \\ \cos(\max(A, B)), & \text{иначе} \end{cases}$$

Вариант 7.

Исходные данные: A, B, C, D. Вычислить

$$M = \begin{cases} e^A + \cos(B), & \text{если } \min(B, C, D) < A \\ \max(D - \sin(A), C), & \text{иначе} \end{cases}$$

Вариант 8.

Исходные данные: A, B, C. Вычислить

$$M = \max(B, C) + T^2, \text{ где } T = \min(\sin(A), B, C).$$

Вариант 9.

Исходные данные: A, B, C. Вычислить

$$M = \begin{cases} \max(\sin(A), \cos(B), \sin(2 \cdot C)), & \text{если } A + \cos(C) > B \\ \max(A, \sin(B)), & \text{иначе} \end{cases}$$

Вариант 10.

Исходные данные: A, B. Вычислить

$$M = \begin{cases} \max(A, B), & \text{если } A < 0 \text{ и } B < 0 \\ \min(\sin(B), \sin(2 \cdot B), \sin(A+1)), & \text{иначе} \end{cases}$$

Вариант 11.

Исходные данные: A, B, C. Вычислить

$$M = \begin{cases} \max(A, B, C - 15), & \text{если } C - 15 > 0 \\ C + \cos(A), & \text{если } C - 15 < 0 \text{ и } A \neq 0 \\ 0, & \text{в остальных случаях} \end{cases}$$

Вариант 12.

Исходные данные: A, B, C. Вычислить

$$T = M - \sqrt{|A|}, \text{ где } M = \begin{cases} \max(A, B), & \text{если } A > \min(\cos(B), C) \\ 0, & \text{если } A < \min(\cos(B), C) \\ 1, & \text{в остальных случаях} \end{cases}$$

Вариант 13.

Исходные данные: A, B, C, D. Вычислить

$$M = D^2 + \min(\min(A, B), \cos(B+C), \sin(C)).$$

Вариант 14.

Исходные данные: A, B, C, D. Вычислить

$$M = \begin{cases} \max(A, B, \min(C, D)), & \text{если } A + B > 0 \\ e^{B+3}, & \text{иначе} \end{cases}$$

Вариант 15.

Исходные данные: A, B, C, D. Вычислить

$$T = M^2 + e^A, \text{ где } M = \max(\sin(A), B, \min(C, D)).$$

Вариант 16.

Исходные данные: A, B, C. Вычислить

$$M = \begin{cases} \max((A), (B), (C)), & \text{если } A > 0 \text{ и } B < 0 \\ A^2, & \text{если } A < 0 \text{ и } B \neq 0 \\ 1, & \text{в остальных случаях} \end{cases}$$

Вариант 17.

Исходные данные: A, B, C. Вычислить

$$M = \begin{cases} \min(A, B, C) + B, & \text{если } A - C > \max(A, B) \\ \cos(A * B) - \sin(A), & \text{иначе} \end{cases}$$

Вариант 18.

Исходные данные: A, B, C. Вычислить

$$M = \begin{cases} \max(A, B, \sin(A + B) * C), & \text{если } A > \min(\sin(A), \cos(B)) \\ C, & \text{иначе} \end{cases}$$

Вариант 19.

Исходные данные: A, B, C. Вычислить

$$M = \begin{cases} \max((A), (B), \sin(C)), & \text{если } C > 0 \\ \min(B, C) + \cos(A), & \text{если } C < 0 \text{ и } A > 0 \\ 0, & \text{в остальных случаях} \end{cases}$$

Вариант 20.

Исходные данные: A, B, C, D. Вычислить

$$M = \min(A, B, \min(C, D)) + \max(\sin(A), D).$$

Вариант 21.

Исходные данные: A, B, C. Вычислить

$$M = \min(\max(A, B), \cos(B + C), \operatorname{tg}(C)).$$

Вариант 22.

Исходные данные: A, B, C. Вычислить

$$M = \begin{cases} \min(\sin(A), (B + C)) + \cos(A - C), & \text{если } A > 0 \\ \max(A, C) + \sin(B - C), & \text{иначе} \end{cases}$$

Вариант 23.

Исходные данные: A, B, C. Вычислить

$$M = \begin{cases} \max(\sin(A), \cos(B+C), \cos(A-C)), & \text{если } A-C > B+C \\ \max(A, \sin(B-C)), & \text{иначе} \end{cases}$$

Вариант 24.

Исходные данные: A, B, C, D. Вычислить

$$M = \min(\max(|A|, B), \max(\operatorname{tg}(A-7), B+C, D)).$$

Вариант 25.

Исходные данные: A, B, C. Вычислить

$$M = \begin{cases} \max(A, B, C-15), & \text{если } C > 0 \\ \operatorname{tg}(C) + \min(\cos(A), \sin(B)) & \text{если } C < 0 \text{ и } A+B < 0 \\ \cos(A+C), & \text{в остальных случаях} \end{cases}$$

Содержание отчёта

1. Название, цель работы, номер варианта и условие (Написать, используя вставку «Формула» в текстовом редакторе).
2. Нарисовать алгоритм.
3. Представить программу, реализующую задачу.
4. Тестовый пример — «скриншот» и ручной счет.
5. Примеры («скриншоты») работы программы, представляющие различные условия задачи.
6. Выводы по работе.

Теоретическая часть

Выражения и операции

Выражение — это комбинация операндов, знаков операций и круглых скобок, используемая для получения значения определенного типа. Операндом (участником операции) может быть константа, переменная, функция или другое выражение. Знаки операции определяют действия (элементарные операции) над операндами в порядке, заданном приоритетами операций и круглыми скобками, которые образуют вложенные подвыражения. Вычисление выражения начинается с самых внутренних скобок.

Унарные операции (с одним операндом) имеют приоритет над бинарными

(с двумя операндами). Мультипликативные операции (умножение, деление, в том числе и деление нацело, деление по модулю (%)) старше аддитивных (сложение, вычитание), за которыми идут операции отношения и логические операции.

Функции в выражении вычисляются для получения значения, участвующего в операции. Для использования стандартных математических функций в программе необходимо подключить их командой препроцессора

#include<math.h> .

Для аргументов функций наиболее используемы в выражениях следующие функции:

sin(x) ~ $\sin x$; **asin(x)** ~ $\arcsin x$;

cos(x) ~ $\cos x$; **acos(x)** ~ $\arccos x$;

tan(x) ~ $\operatorname{tg} x$; **atan(x)** ~ $\operatorname{arctg} x$;

fabs(x) ~ $|x|$ — для вещественного x (результат **double**);

abs(i) ~ $|i|$ — для целого i (результат **int**);

log(x) ~ $\ln x$; **log10(x)** ~ $\lg x$;

exp(x) ~ e^x , где e — основание натурального логарифма (2.7);

sqrt(x) ~ \sqrt{x} ;

floor(x) ~ наибольшее целое число $\leq x$ (результат **double**);

ceil(x) ~ наименьшее целое число $\geq x$ (результат **double**);

srand(u) ~ инициализация генератора случайных чисел (результат **double**);

rand ~ равномерное псевдослучайное число от 0 до 232-1 (результат **int**).

Аргументы функции любой сложности, включая и вложенные функции, должны заключаться в круглые скобки после имени функции.

Общее правило вычисления выражения: перед вычислением все входящие в него переменные должны быть определены, то есть должны иметь численные значения.

По роли, выполняемой в программе, выражения можно разделить на арифметические и логические.

Арифметические выражения и операции

Арифметические выражения представляют собой запись на языке C в одну строку (или более) математической формулы с учетом порядка вычисления

выражения и старшинства операций с целью получения числового значения (целого или вещественного).

Выражение можно переносить на другую строку без повторения знака операций в отличие от записи математической формулы, но нельзя опускать знаки операций:

Работа с ветвящимися процессами

Операторы

Оператор — это предложение в тексте программы, заканчивающееся точкой с запятой (;), выполняющее некоторое действие в соответствии с алгоритмом задачи. К простым операторам относятся: оператор-выражение, операторы присваивания, вызова процедуры, операторы передачи управления, пустой оператор, составной оператор.

К сложным операторам относятся: условный оператор, оператор-переключатель, операторы цикла.

Условный оператор (if)

В разветвляющихся вычислительных процессах на некотором этапе вычислений осуществляется проверка заданного условия (в виде выражения) и выбор одной из двух ветвей продолжения вычислительного процесса (ветви «да» и «нет»). Для реализации разветвления в программе используется условный оператор (в полной и сокращенной форме) и оператор выбора.

Полный условный оператор имеет вид:

if (условие) {Блок инструкций 1}

else {Блок инструкций 2};

if (если), **else** (иначе) — ключевые слова;

условие — это выражение (не обязательно логическое), значение которого сравнивается с 0. При этом, если значение выражения не равно 0 (условие истинно), то выполняется «Блок инструкций 1», а «Блок инструкций 2» пропускается, иначе (выражение = 0, то есть условие ложно) выполняется «Блок инструкций 2»;

«Блок инструкций 1», «Блок инструкций 2» — одиночные или составные операторы.

Составной оператор — это группа операторов, заключенная в фигурные

скобки, которая выступает как единое целое в сложных операторах (условном и операторах цикла):

{оператор 1; оператор 2; ...}.

Неполный условный оператор образуется, если отсутствует часть **else** условного оператора:

if (условие) оператор.

Тогда, если условие истинно (не равно 0), то выполняется оператор, иначе он пропускается, и выполняется оператор, следующий за условным оператором. В схеме алгоритма это выражается отсутствием действий по ветви «нет».

Лабораторная работа № 3

ИСПОЛЬЗОВАНИЕ ЛОГИЧЕСКОЙ ФУНКЦИИ

Цель работы: повысить навыки работы со структурой «Если» и освоение работы с логической функцией.

Задание и порядок выполнения лабораторной работы

Разработать и отладить программу, обеспечивающую вычисление и вывод на экран значений функции $F(x)$ на интервале от x_1 до x_2 с шагом dx (исходный вектор x должен содержать не менее 5 точек). При этом функция $F(x)$ должна принимать действительные значения, если логическая функция G не равна нулю, и целое значение в противном случае. Параметры a , b , c и значения x_1 , x_2 , dx должны вводиться с клавиатуры. Через А,В,С обозначены целые части соответственно параметров a, b, c . Выводить на экран промежуточные значения (А, В, С, G).

Вариант 1.

$$F(x) = \begin{cases} a \cdot x^2 - b, & \text{если } x < 0 \text{ и } b \neq 0 \\ \frac{(x-a)}{(x-c)}, & \text{если } x > 0 \text{ и } b = 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

$$G = (A \text{ ИЛИ } B) \text{ И } (A \text{ ИЛИ } C)$$

Вариант 2.

$$F(x) = \begin{cases} \frac{x-a}{x}, & \text{если } x+5 > 0 \text{ и } c \neq 0 \\ \frac{1}{a \cdot x} - b, & \text{если } x+5 < 0 \text{ и } c = 0 \\ \frac{10 \cdot x}{(c-4)}, & \text{в остальных случаях} \end{cases}$$

$$G = (A \text{ И } B) \text{ ИЛИ } (B \text{ И } C)$$

Вариант 3.

$$F(x) = \begin{cases} \frac{a}{(x-c)}, & \text{если } a > 0 \text{ и } c = 0 \\ a \cdot x^2 + b \cdot x + c, & \text{если } a < 0 \text{ и } c \neq 0 \\ a \cdot (x+c), & \text{в остальных случаях} \end{cases}$$

$$G = A \text{ И } (B \text{ ИЛИ } C)$$

Вариант 4.

$$F(x) = \begin{cases} \frac{(x-a)}{(-c)}, & \text{если } c > 0 \text{ и } x = 0 \\ (-a) \cdot x - c, & \text{если } c < 0 \text{ и } x \neq 0 \\ \frac{b \cdot x}{(c-a)}, & \text{в остальных случаях} \end{cases}$$

G=A ИЛИ B ИЛИ C

Вариант 5.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } x > 0 \text{ и } b = 0 \\ \frac{a-x}{(10+b)}, & \text{если } x < 0 \text{ и } b \neq 0 \\ 3 \cdot x + \frac{2}{c}, & \text{в остальных случаях} \end{cases}$$

G=(A ИЛИ B) И НЕ(C)

Вариант 6.

$$F(x) = \begin{cases} \frac{(x+a)}{(x+c)}, & \text{если } c > 0 \text{ и } b = 0 \\ a \cdot x^2 + b^2 \cdot x, & \text{если } c < 0 \text{ и } b \neq 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

G=(A И B) ИЛИ (A И C)

Вариант 7.

$$F(x) = \begin{cases} \frac{(x-a)}{x}, & \text{если } x > 5 \text{ и } c = 0 \\ a \cdot x^2 - b, & \text{если } x < 5 \text{ и } c \neq 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

G=(A ИЛИ B) МОД2 (A ИЛИ C)

Вариант 8.

$$F(x) = \begin{cases} \frac{(a-x)}{(c-x)}, & \text{если } c > 0 \text{ и } a = 0 \\ (-a) \cdot x^2, & \text{если } c < 0 \text{ и } a \neq 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

$G = (A \text{ МОД } B) \text{ И НЕ } (A \text{ ИЛИ } C)$

Вариант 9.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } a > 0 \text{ и } x = 0 \\ a \cdot x^2 + b^2 \cdot x, & \text{если } a < 0 \text{ и } x \neq 0 \\ 1 + \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

$G = \text{НЕ } (A \text{ ИЛИ } B) \text{ И } (B \text{ ИЛИ } C)$

Вариант 10.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } x > 3 \text{ и } b = 0 \\ a \cdot x^2 - b \cdot x + c, & \text{если } x < 3 \text{ и } b \neq 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

$G = \text{НЕ } (A \text{ ИЛИ } B) \text{ И } (A \text{ МОД } 2 \text{ } C)$

Вариант 11.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } x > 15 \text{ и } c = 0 \\ a \cdot x^2 + \frac{b}{c}, & \text{если } x < 1 \text{ и } c \neq 0 \\ \frac{x^2}{c^2}, & \text{в остальных случаях} \end{cases}$$

$G = (A \text{ И } B) \text{ МОД } 2 \text{ } C$

Вариант 12.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } x > 0,6 \text{ и } b+c=0 \\ a \cdot x^3 + b^2 + c, & \text{если } x < 0,6 \text{ и } b+c \neq 0 \\ \frac{x}{c} + \frac{x}{a}, & \text{в остальных случаях} \end{cases}$$

$$G = (A \text{ ИЛИ } B) \text{ И } C$$

Вариант 13.

$$F(x) = \begin{cases} \frac{(x-a)}{x}, & \text{если } x-1 > 0 \text{ и } b+x=0 \\ a \cdot x^2 + b, & \text{если } x-1 < 0 \text{ и } b-x \neq 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

$$G = (A \text{ ИЛИ } B) \text{ МОД}2 (B \text{ И } C)$$

Вариант 14.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } x+c > 0 \text{ и } a=0 \\ a \cdot x^3 - b, & \text{если } x+c < 0 \text{ и } a \neq 0 \\ \frac{x}{c} + \frac{c}{x}, & \text{в остальных случаях} \end{cases}$$

$$G = (A \text{ МОД}2 B) \text{ ИЛИ } (A \text{ МОД}2 C)$$

Вариант 15.

$$F(x) = \begin{cases} \frac{x}{(x-c)} + 5,5, & \text{если } x > 0 \text{ и } b=0 \\ a \cdot x^2 + b, & \text{если } x < 0 \text{ и } b \neq 0 \\ \frac{x}{(-c)}, & \text{в остальных случаях} \end{cases}$$

$$G = \text{НЕ } (A \text{ ИЛИ } B \text{ ИЛИ } C)$$

Вариант 16.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } x \neq 0 \text{ и } b=0 \\ a \cdot (x+c)^3 - b, & \text{если } x=0 \text{ и } b \neq 0 \\ a + \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

$$G = (A \text{ МОД } 2 B) \text{ И НЕ } (A \text{ ИЛИ } C)$$

Вариант 17.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } x+10 > 0 \text{ и } b=0 \\ a \cdot x^2 - c \cdot x + b, & \text{если } x+10 < 0 \text{ и } b \neq 0 \\ \frac{(-x)}{(a-c)}, & \text{в остальных случаях} \end{cases}$$

$$G = (A \text{ ИЛИ } B) \text{ И НЕ } (A \text{ ИЛИ } C)$$

Вариант 18.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } x > 0 \text{ и } b=0 \\ a \cdot x^3 + b \cdot x^2, & \text{если } x < 0 \text{ и } b \neq 0 \\ \frac{(x+5)}{c \cdot (x-10)}, & \text{в остальных случаях} \end{cases}$$

$$G = \text{НЕ } (A \text{ И } B \text{ И } C)$$

Вариант 19.

$$F(x) = \begin{cases} \frac{(x-b \cdot c)}{a \cdot x}, & \text{если } x > 5 \text{ и } b=0 \\ a \cdot (x+7)^2 - b, & \text{если } x < 5 \text{ и } b \neq 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

$$G = (A \text{ МОД } B) \text{ ИЛИ } (A \text{ МОД } C)$$

Вариант 20.

$$F(x) = \begin{cases} \frac{(x-a)}{(x-c)}, & \text{если } x > 0 \text{ и } b = 0 \\ \frac{(2 \cdot x - c)}{(c \cdot x - a)}, & \text{если } x < 0 \text{ и } b \neq 0 \\ \frac{(-x)}{c} - \frac{c}{2 \cdot x}, & \text{в остальных случаях} \end{cases}$$

$G = \text{НЕ} (A \text{ ИЛИ } B) \text{ И } \text{НЕ} (A \text{ ИЛИ } C)$

Вариант 21.

$$F(x) = \begin{cases} \frac{(x-b)}{(-c)}, & \text{если } x+c > 0 \text{ и } a < 0 \\ a \cdot x^3 + b^2, & \text{если } x+c < 0 \text{ и } a \neq 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

$G = \text{НЕ} (A \text{ ИЛИ } B) \text{ ИЛИ } \text{НЕ}(B \text{ ИЛИ } C)$

Вариант 22.

$$F(x) = \begin{cases} \frac{(-a)}{(x^2 - c)}, & \text{если } x > 0 \text{ и } a > 0 \\ a \cdot x^2, & \text{если } x < 0 \text{ и } a > 0 \\ 0, & \text{в остальных случаях} \end{cases}$$

$G = (A \text{ И } B) \text{ ИЛИ } (A \text{ ИЛИ } \text{НЕ}(C))$

Вариант 23.

$$F(x) = \begin{cases} \frac{(x+a)}{(c-x)}, & \text{если } c > 0 \text{ и } x = 0 \\ a \cdot x^3 - b, & \text{если } c < 0 \text{ и } x \neq 0 \\ \frac{a}{x}, & \text{в остальных случаях} \end{cases}$$

$G = (A \text{ МОД} 2 B) \text{ ИЛИ } \text{НЕ}(C)$

Вариант 24.

$$F(x) = \begin{cases} \frac{(2 \cdot x - a)}{(x - 2 \cdot c)}, & \text{если } x > 0 \text{ и } a = 0 \\ x^3 - b^3, & \text{если } x < 0 \text{ и } a \neq 0 \\ \frac{x}{c} - \frac{a}{x}, & \text{в остальных случаях} \end{cases}$$

$$G = (A \text{ И } B) \text{ МОД}2 (A \text{ И } C)$$

Вариант 25.

$$F(x) = \begin{cases} \frac{(-x - a)}{(-c)}, & \text{если } c > 0 \text{ и } x = 0 \\ a \cdot x - c, & \text{если } x < 0 \text{ и } a \neq 0 \\ \frac{x}{c} + \frac{c}{x}, & \text{в остальных случаях} \end{cases}$$

$$G = \text{НЕ}(A \text{ ИЛИ } B) \text{ ИЛИ } (A \text{ И } C)$$

Содержание отчёта

1. Название, цель работы, номер варианта и условие (Написать, используя вставку «Формула» в текстовом редакторе).
2. Нарисовать алгоритм.
3. Представить программу (с комментариями).
4. Примеры («скриншоты») работы программы, реализующие все условия задачи.
5. Выводы по работе.

Теоретическая часть

Логические выражения и операции

Выражения, обрабатывающие логические значения с помощью операций отношения и логических операций, сравнивают каждый операнд с нулем. Если операнд равен нулю, то его значение *ложно*, а если операнд не равен нулю, то его значение *истинно*. Результат операций отношения или логических может

быть 0 (ложь) или 1 (истина).

Отношение сравнивает два операнда, связанных операцией отношения. Операндами отношения являются арифметические выражения. Логические операции имеют относительные приоритеты, начиная со старшего (1):

«!» — логическое «НЕ» — приоритет 1;

«||» — логическое «ИЛИ» — приоритет 4;

«&&» — логическое «И» — приоритет 3;

«<, <=, >, >=, ==, !=» — отношения — приоритет 2.

Лабораторная работа № 4

МНОЖЕСТВЕННЫЙ ВЫБОР

Цель работы: изучить возможность решения задач множественного выбора с помощью функции «switch».

Задание на лабораторную работу

Написать программу вывода значения на экран, используя оператор выбора.

Вариант 1.

Определить название месяца года, следующего за заданным месяцем.

Вариант 2.

Для целого числа d от 1 до 99, обозначающего денежную единицу, дописать слово “копейка” в правильной форме.

Вариант 3.

Определить название столицы по заданному названию страны.

Вариант 4.

Определить величину в метрах некоторой длины, заданной в одной из указанных единиц измерения (километр, метр, дециметр, сантиметр, миллиметр).

Вариант 5.

Определить написание заданной десятичной цифры римскими цифрами.

Вариант 6.

Определить двоичный код заданной десятичной цифры.

Вариант 7.

Определить сезон года (зима, весна, лето, осень), на который приходится заданный месяц.

Вариант 8.

Определить название континента (Азия, Америка, Африка, Европа) по заданному названию страны.

Вариант 9.

Определить название цвета радуги, следующего за заданным цветом.

Вариант 10.

Определить название интервала (секунда, терция, кварта, квинта, секста, септима), образованного двумя заданными нотами (до, ре, ми, фа, соль, ля, си).

Вариант 11.

Определить название десятичной цифры по заданному ее значению.

Вариант 12.

Для целого числа k от 1 до 99 вывести фразу «Мне k лет», учитывая при этом, что при некоторых значениях k слово «лет» надо заменить на слово «год» или «года».

Вариант 13.

Для натурального числа k вывести фразу «Мы нашли k грибов в лесу», согласовав слово «гриб» с числом k .

Вариант 14.

Для целого числа d от 1 до 999, обозначающего денежную единицу, дописать слово «рубль» в правильной форме.

Вариант 15.

Определить название k -го месяца после заданного месяца года.

Вариант 16.

Вычислить стоимость междугородного телефонного разговора заданной продолжительности. Цена одной минуты определяется по указанному коду города.

Вариант 17.

Вывести указанное слово из группы однотипно склоняемых слов (степь, боль, тетрадь, дверь) в заданном падеже (им., род., дат., вин., твор., предл.).

Вариант 18.

Корабль сначала шел по заданному курсу (север, восток, юг, запад). Затем его курс был изменен согласно заданному приказу (вперед, вправо, назад, влево). Определить новый курс корабля.

Вариант 19.

Определить количество дней в указанном месяце заданного года.

Вариант 20.

Определить, образует ли заданная тройка чисел y (год), m (месяц), d

(день) правильную дату.

Вариант 21.

По заданной дате d (день), m (месяц), y (год) определить дату d_1 , m_1 , y_1 следующего дня.

Вариант 22.

Определить порядковый номер того дня високосного года, который имеет заданную дату d (день), m (месяц).

Вариант 23.

Определить d (день), m (месяц) – дату k -го по счету дня високосного года. Считая, что год не високосный и его 1 января приходится на день недели wd_1 , определить wd – день недели, на который приходится день с датой d (день), m (месяц).

Вариант 24.

Считая, что год не високосный и его 1 января приходится на день недели wd_1 , определить количество понедельников в году, приходящихся на 13-е числа.

Вариант 25.

Определить название планеты по ее порядковому номеру от Солнца.

Порядок выполнения работы

1. Открыть одно из приложений и создать проект (под своей фамилией).
2. Написать программу выбора по варианту.
3. Предъявить программу преподавателю.

Письменный отчет НЕ оформляется.

Теоретическая часть

Оператор выбора (switch)

Для определения варианта продолжений действий алгоритма, когда число ветвей *больше двух*, в программе используется оператор вида:

```
switch (выражение)
{
case метка1: Блок инструкций 1; break;
case метка2: Блок инструкций 2; break;
```

.....

default : Блок инструкций K; break;

}

switch (переключатель), **case** (вариант), **default** (отсутствие варианта) — ключевые слова;

выражение — переключатель для получения значения *целого* или *символьного* типа;

метка — одно из значений типа выражение, которые могут следовать в *любом* порядке;

блок инструкций — последовательность операторов варианта, которая не заключается в операторные скобки и может даже отсутствовать.

Действие оператора выбора состоит в следующем:

- по значению выражения определяется совпадающая с ним метка варианта (ветви) действий алгоритма.
- после выполнения операторов варианта с данной меткой выполняются операторы следующих вариантов, если только не встретится оператор передачи управления **goto** или **break**. Оператор **goto** может передать управление любому оператору программы, а оператор **break** (прервать) передает управление оператору, следующему за оператором **switch**.
- если нужная метка не найдена и есть необязательное ключевое слово **default**, то выполняются операторы, идущие за ним, в противном случае управление передается оператору, следующему за оператором **switch**.

ИСПОЛЬЗОВАНИЕ ФУНКЦИИ ЦИКЛА

Цель работы: изучить основные принципы программирования циклических задач.

Задание и порядок выполнения лабораторной работы

Разработать программу, с использованием фиксированного числа повторений (20 — 40), вычисления ряда. Результат проверить формулой (правая часть равенства). Выводить все значения (не менее 8 дробных знаков). Количество повторений вводить с клавиатуры.

Вариант 1.

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1)} = \ln(2)$$

Вариант 2.

$$\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} = \ln(2)$$

Вариант 3.

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{(2 \cdot k + 1)} = \frac{\pi}{4}$$

Вариант 4.

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{(3 \cdot k + 1) \cdot (3 \cdot k + 2)} = \frac{2}{3} \cdot \ln(2)$$

Вариант 5.

$$\sum_{k=0}^{\infty} \frac{1}{(3 \cdot k + 1) \cdot (3 \cdot k + 4)} = \frac{1}{3}$$

Вариант 6.

$$\sum_{k=0}^{\infty} \frac{1}{(4 \cdot k + 1) \cdot (4 \cdot k + 3)} = \frac{\pi}{8}$$

Вариант 7.

$$\sum_{k=0}^{\infty} \frac{1}{(4 \cdot k + 1) \cdot (4 \cdot k + 5)} = \frac{1}{4}$$

Вариант 8.

$$\sum_{k=0}^{\infty} \frac{1}{(4 \cdot k + 3) \cdot (4 \cdot k + 7)} = \frac{1}{12}$$

Вариант 9.

$$\sum_{k=0}^{\infty} \frac{1}{(5 \cdot k + 1) \cdot (5 \cdot k + 6)} = \frac{1}{5}$$

Вариант 10.

$$\sum_{k=0}^{\infty} \frac{1}{(5 \cdot k + 2) \cdot (5 \cdot k + 7)} = \frac{1}{19}$$

Вариант 11.

$$\sum_{k=0}^{\infty} \frac{1}{(5 \cdot k + 3) \cdot (5 \cdot k + 8)} = \frac{1}{15}$$

Вариант 12.

$$\sum_{k=0}^{\infty} \frac{1}{(5 \cdot k + 4) \cdot (5 \cdot k + 9)} = \frac{1}{20}$$

Вариант 13.

$$\sum_{k=0}^{\infty} \frac{1}{(6 \cdot k + 1) \cdot (6 \cdot k + 7)} = \frac{1}{6}$$

Вариант 14.

$$\sum_{k=0}^{\infty} \frac{1}{(6 \cdot k + 5) \cdot (6 \cdot k + 11)} = \frac{1}{30}$$

Вариант 15.

$$\sum_{k=1}^{\infty} \frac{1}{k \cdot (k+1) \cdot (k+2)} = \frac{1}{4}$$

Вариант 16.

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{k \cdot (k+1) \cdot (2 \cdot k+1)} = 3 - \pi$$

Вариант 17.

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1)(2 \cdot k+1)(2 \cdot k+3)} = 1 - \ln(2)$$

Вариант 18.

$$\sum_{k=0}^{\infty} \frac{1}{(k+1) \cdot (2 \cdot k+1) \cdot (2 \cdot k+3)} = 2 \cdot \ln(2) - 1$$

Вариант 19.

$$\sum_{k=0}^{\infty} \frac{1}{(k+1) \cdot (2 \cdot k + 1) \cdot (4 \cdot k + 1)} = \frac{\pi}{3}$$

Вариант 20.

$$\sum_{k=0}^{\infty} \frac{1}{(k+1) \cdot (4 \cdot k + 1) \cdot (4 \cdot k + 3)} = \frac{\pi}{3} - \ln(2)$$

Вариант 21.

$$\sum_{k=0}^{\infty} \frac{1}{(k+1) \cdot (4 \cdot k + 3) \cdot (4 \cdot k + 5)} = 3 \cdot \ln(2) - 2$$

Вариант 22.

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{(2 \cdot k + 1)(2 \cdot k + 3)(2 \cdot k + 5)} = \frac{\pi}{8} - \frac{1}{3}$$

Вариант 23.

$$\sum_{k=0}^{\infty} \frac{1}{(2 \cdot k + 1) \cdot (2 \cdot k + 3) \cdot (2 \cdot k + 5)} = \frac{1}{12}$$

Вариант 24.

$$\sum_{k=0}^{\infty} \frac{1}{(2 \cdot k + 1) \cdot (4 \cdot k + 1) \cdot (4 \cdot k + 3)} = \frac{1}{2} \cdot \ln(2)$$

Вариант 25.

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{(4 \cdot k + 1) \cdot (4 \cdot k + 3) \cdot (4 \cdot k + 5)} = \frac{\pi}{16} - \frac{1}{8}$$

Содержание отчёта

1. Название, цель работы, номер варианта и условие (Написать, используя вставку «Формула» в текстовом редакторе).
2. Нарисовать алгоритм.
3. Представить программу (с комментариями), реализующую задачу.
4. Примеры («скриншоты») работы программы, подтверждающие правильность решения.
5. Выводы по работе.

Теоретическая часть**Циклические вычислительные процессы**

В *циклических* алгоритмах выполняются *повторяющиеся* вычисления и действия. Управление циклом осуществляется с помощью переменной, называемой *параметром цикла*. Циклы могут быть *одиночными* и *вложенными*, причем глубина вложения не ограничена. *Окончание* циклов может осуществляться либо по достижении *фиксированного* числа повторений, либо по достижении заданной *точности* вычислений (итерационные циклы), либо *принудительно* при выполнении некоторого условия

Циклы с фиксированным числом повторений

Этот наиболее гибкий по своим возможностям оператор цикла имеет вид

for (инициатор; условие_выполнения; модификатор) /*заголовок цикла*/
{Блок инструкций} /*тело цикла*/

for (для) — ключевое слово;

инициатор — выражение, задающее *начальное* значение переменной оператора цикла;

условие_выполнения — выражение, значение которого определяет *условие* продолжения цикла;

модификатор — инструкция *изменения* параметра цикла;

блок инструкций — одиночный, составной или пустой оператор, выполняющий *действия* в цикле.

Существенно то, что условие цикла проверяется *перед* выполнением тела цикла. Это значит, что цикл может ни разу не выполниться, если условие сразу будет *ложным*.

Лабораторная работа № 6

ВЫЧИСЛЕНИЕ ИНТЕГРАЛА

Цель работы: закрепить базовые навыки использования циклической функций с фиксированным числом повторений в решении математических задач.

Задание и порядок выполнения лабораторной работы

Представить интеграл в виде ряда и вычислить его, используя функцию цикла. Границы интервала и шаг вводить с клавиатуры.

Вариант 1.

$$\int \frac{x^2 - x + 2}{x^4 + 10 \cdot x^2 + 9} dx$$

Вариант 2.

$$\int \frac{x \cdot \sin(3 \cdot x)}{(x^2 + 4)^2} dx$$

Вариант 3.

$$\int \frac{x - 1}{(x^2 + 4)^2} dx$$

Вариант 4.

$$\int \frac{(x - 1) \cdot \sin(x)}{(x^2 + 9)^2} dx$$

Вариант 5.

$$\int \frac{1}{(x^4 + 1)^2} dx$$

Вариант 6.

$$\int \frac{\cos(2 \cdot x)}{(x^2 + 1)^2} dx$$

Вариант 7.

$$\int \frac{1}{(x^2 + 4)^2 \cdot (x^2 + 16)} dx$$

Вариант 8.

$$\int \frac{x^2 \cdot \cos(x)}{(x^2 + 1)^2} dx$$

Вариант 9.

$$\int \frac{1}{(x^2 - x + 1)^2} dx$$

Вариант 10.

$$\int \frac{(x + 1) \cdot \cos(x)}{(x^4 + 5 \cdot x^2 + 6)^2} dx$$

Вариант 11.

$$\int \frac{1}{(x^2 + 9) \cdot (x^2 + 4)} dx$$

Вариант 12.

$$\int \frac{x \cdot \sin\left(\frac{x}{2}\right)}{(x^2 + 9) \cdot (x^2 + 1)} dx$$

Вариант 13.

$$\int \frac{1}{x^4 + 10 \cdot x^2 + 9} dx$$

Вариант 14.

$$\int \frac{(x^2 + 3) \cdot \cos(2 \cdot x)}{(x^4 + 3 \cdot x^2 + 2)} dx$$

Вариант 15.

$$\int \frac{\cos(2 \cdot x)}{(x^2 + 2 \cdot x + 2)} dx$$

Вариант 16.

$$\int \frac{(x^3 - 2) \cdot \cos\left(\frac{x}{2}\right)}{(x^2 + 1)^2} dx$$

Вариант 17.

$$\int \frac{1}{(x^2 + 4) \cdot (x^2 + 9)^2} dx$$

Вариант 18.

$$\int \frac{(x^2 - x) \cdot \sin(x)}{(x^4 + 9 \cdot x^2 + 20)} dx$$

Вариант 19.

$$\int \frac{1}{(x^2 + 9) \cdot (x^2 + 1)^2} dx$$

Вариант 20.

$$\int \frac{x \cdot \cos(x)}{(x^2 - 2 \cdot x + 17)^2} dx$$

Вариант 21.

$$\int \frac{x^2 + 1}{(x^2 + x + 1)^2} dx$$

Вариант 22.

$$\int \frac{x \cdot \sin(2 \cdot x) - \sin(x)}{(x^2 + 4)^2} dx$$

Вариант 23.

$$\int \frac{x^2 + 1}{(x^2 + 4 \cdot x + 13)^2} dx$$

Вариант 24.

$$\int \frac{\cos(5 \cdot x)}{(x^2 + 1)^2 \cdot (x^2 + 4)} dx$$

Вариант 25.

$$\int \frac{x^2}{(x^2 + 1)^2} dx$$

Вариант 26.

$$\int \frac{x^3 \cdot \sin(x)}{(x^4 + 9 \cdot x^2 + 20)} dx$$

Вариант 27.

$$\int \frac{x^2}{(x^2 + 2 \cdot x + 2)^2} dx$$

Вариант 28.

$$\int \frac{(x + 1) \cdot \sin(2 \cdot x)}{(x^2 + 2 \cdot x + 2)} dx$$

Вариант 29.

$$\int \frac{\cos(3 \cdot x)}{(x^2 + 1)^2 \cdot (x^2 + 9)} dx$$

Вариант 30.

$$\int \frac{(x + 1) \cdot \sin(2 \cdot x)}{(x^2 + 9)^2} dx$$

Содержание отчёта

1. Название, цель работы, номер варианта и условие (Написать, используя вставку «Формула» в текстовом редакторе).
2. Представить программу (с комментариями).
3. Тестовый пример — «скриншот» и ручной счет.
4. Примеры («скриншоты») работы программы, с использованием различных интервалов интегрирования
5. Выводы по работе.

ИСПОЛЬЗОВАНИЕ РАЗЛИЧНЫХ ФУНКЦИИ ЦИКЛА

Цель работы: закрепить базовые навыки использования различных циклических функций в решении математических задач.

Задание и порядок выполнения лабораторной работы

Разработать программу, состоящую из трех частей:

- вычисление значения функции, описанной в левой части варианта;
- вычисление суммы ряда с использованием оператора цикла **for** — количество повторений 20 — 30 (вводится с клавиатуры). Все значения выводятся на экран;
- вычисление суммы ряда с использованием оператора **while() do {}** — точность вычисления задается с клавиатуры (1,0E-05 — 1,0E-07). Все значения выводятся на экран.

Вариант 1.

$$\operatorname{arctg}(x) = \sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{(2n+1)}}{(2n+1)} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots, \quad |x| < 1$$

Вариант 2.

$$\operatorname{arctg}(x) = \frac{\pi}{2} - \sum_{n=0}^{\infty} (-1)^{n+1} \cdot \frac{1}{(2n+1) \cdot x^{(2n+1)}} = \frac{\pi}{2} - \left(\frac{1}{x} - \frac{1}{3 \cdot x^3} + \frac{1}{5 \cdot x^5} - \frac{1}{7 \cdot x^7} + \dots \right), \quad x > 1$$

Вариант 3.

$$\operatorname{arcth}(x) = \sum_{n=0}^{\infty} \frac{x^{(2n+1)}}{(2n+1)} = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots, \quad |x| < 1$$

Вариант 4.

$$\operatorname{arcth}(x) = \sum_{n=0}^{\infty} \frac{1}{(2n+1) \cdot x^{(2n+1)}} = \frac{1}{x} + \frac{1}{3 \cdot x^3} + \frac{1}{5 \cdot x^5} + \frac{1}{7 \cdot x^7} + \dots, \quad |x| > 1$$

Вариант 5.

$$\ln(x) = \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{(x-1)^n}{n} = \frac{(x-1)^1}{1} - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \dots, \quad 0 < x < 2$$

Вариант 6.

$$\ln(1+x) = \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{x^n}{n} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots, \quad -1 < x \leq 1$$

Вариант 7.

$$\ln(1-x) = - \sum_{n=1}^{\infty} \frac{x^n}{n} = - \left[x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \dots \right], \quad -1 \leq x < 1$$

Вариант 8.

$$\ln \left| \frac{1+x}{1-x} \right| = 2 \cdot \sum_{n=0}^{\infty} \frac{x^{(2n+1)}}{(2 \cdot n + 1)} = 2 \left[x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots \right], \quad x^2 < 1$$

Вариант 9.

$$\ln \left| \frac{x+1}{x-1} \right| = 2 \cdot \sum_{n=0}^{\infty} \frac{1}{(2 \cdot n + 1) \cdot x^{2n+1}} = 2 \left[\frac{1}{x} + \frac{1}{3 \cdot x^3} + \frac{1}{5 \cdot x^5} + \frac{1}{7 \cdot x^7} + \dots \right], \quad x^2 > 1$$

Вариант 10.

$$e^x \cdot (1+x) = \sum_{n=0}^{\infty} \frac{x^n \cdot (n+1)}{n!} = 1 + \frac{2 \cdot x}{1!} + \frac{3 \cdot x^2}{2!} + \frac{4 \cdot x^3}{3!} + \dots, \quad |x| < 2,4$$

Вариант 11.

$$e^{-x^2} = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{n!} = \frac{x^0}{0!} - \frac{x^2}{1!} + \frac{x^4}{2!} - \frac{x^6}{3!} + \dots, \quad x < 1$$

Вариант 12.

$$\ln(x) = 2 \cdot \sum_{n=0}^{\infty} \frac{(x-1)^{(2n+1)}}{(2 \cdot n + 1) \cdot (x+1)^{(2n+1)}} = 2 \left[\frac{(x-1)}{(x+1)} + \frac{(x-1)^3}{3 \cdot (x+1)^3} + \frac{(x-1)^5}{5 \cdot (x+1)^5} + \dots \right], \quad x > 0$$

Вариант 13.

$$\ln(x) = \sum_{n=1}^{\infty} \frac{(x-1)^n}{n \cdot x^n} = \frac{(x-1)}{x} + \frac{(x-1)^2}{2 \cdot x^2} + \frac{(x-1)^3}{3 \cdot x^3} + \dots, \quad x > 0,5$$

Вариант 14.

$$\sin(x) = \sum_{n=1}^{\infty} (-1)^{n-1} \cdot \frac{x^{(2n-1)}}{(2 \cdot n - 1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots, |x| < \infty$$

Вариант 15.

$$\cos(x) = \sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n}}{(2 \cdot n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots, |x| < \infty$$

Вариант 16.

$$\text{sh}(x) = \sum_{n=1}^{\infty} \frac{x^{(2n-1)}}{(2 \cdot n - 1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots, \quad |x| < \infty, \quad \text{sh}(x) = \frac{e^x - e^{-x}}{2}$$

Вариант 17.

$$\text{ch}(x) = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2 \cdot n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \frac{x^8}{8!} + \dots, \quad |x| < \infty, \quad \text{ch}(x) = \frac{e^x + e^{-x}}{2}$$

Вариант 18.

$$\sin^2(x) = \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{2^{2n-1} \cdot x^{2n}}{(2 \cdot n)!} = \frac{2^1 \cdot x^2}{2!} - \frac{2^3 \cdot x^4}{4!} + \frac{2^5 \cdot x^6}{6!} - \frac{2^7 \cdot x^8}{8!} + \dots, \quad x < 1$$

Вариант 19.

$$\cos^2(x) = 1 - \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{2^{(2n-1)} \cdot x^{2n}}{(2 \cdot n)!} = 1 - \left[\frac{2^1 \cdot x^2}{2!} - \frac{2^3 \cdot x^4}{4!} + \frac{2^5 \cdot x^6}{6!} - \frac{2^7 \cdot x^8}{8!} + \dots \right], \quad x < 1$$

Вариант 20.

$$\text{arctg}(x) = \frac{\pi}{2} + \sum_{n=0}^{\infty} (-1)^{n+1} \cdot \frac{x^{(2n+1)}}{2 \cdot n + 1} = \frac{\pi}{2} - \left[x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots \right], \quad |x| < 1$$

Вариант 21.

$$\operatorname{arctg}(x) = \frac{\pi}{2} + \sum_{n=0}^{\infty} (-1)^{n+1} \cdot \frac{1}{(2 \cdot n + 1) \cdot x^{(2n+1)}} = \frac{\pi}{2} + \left[\frac{1}{x} - \frac{1}{3 \cdot x^3} + \frac{1}{5 \cdot x^5} - \frac{1}{7 \cdot x^7} + \dots \right], \quad |x| > 1$$

Вариант 22.

$$\operatorname{arctg}(x) = \sum_{n=0}^{\infty} (-1)^n \cdot \frac{1}{(2 \cdot n + 1) \cdot x^{(2n+1)}} = \left[\frac{1}{x} - \frac{1}{3 \cdot x^3} + \frac{1}{5 \cdot x^5} - \frac{1}{7 \cdot x^7} + \dots \right], \quad |x| > 1$$

Вариант 23.

$$\operatorname{arcsin}(x) = x + \sum_{n=1}^{\infty} \frac{1 \cdot 3 \cdot 5 \dots (2 \cdot n - 1) \cdot x^{(2n+1)}}{2 \cdot 4 \cdot 6 \dots (2 \cdot n) \cdot (2 \cdot n + 1)} = x + \frac{x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} + \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots, \quad |x| < 1$$

Вариант 24.

$$\operatorname{arccos}(x) = \frac{\pi}{2} - x - \sum_{n=1}^{\infty} \frac{1 \cdot 3 \cdot 5 \dots (2 \cdot n - 1) \cdot x^{(2n+1)}}{2 \cdot 4 \cdot 6 \dots (2 \cdot n) \cdot (2 \cdot n + 1)} = \frac{\pi}{2} - \left[x + \frac{x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} + \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots \right], \quad |x| < 1$$

Вариант 25.

$$\operatorname{arcsh}(x) = x + \sum_{n=1}^{\infty} (-1)^n \cdot \frac{1 \cdot 3 \cdot 5 \dots (2 \cdot n - 1) \cdot x^{(2n+1)}}{2 \cdot 4 \cdot 6 \dots (2 \cdot n) \cdot (2 \cdot n + 1)} = x + \frac{x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} - \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots, \quad |x| < 1$$

Вариант 26.

$$\operatorname{arcch}(x) = \ln(2 \cdot x) - \sum_{n=1}^{\infty} \frac{1 \cdot 3 \cdot 5 \dots (2 \cdot n - 1)}{2 \cdot 4 \cdot 6 \dots (2 \cdot n) \cdot (2 \cdot n) \cdot x^{2n}} = \ln(2 \cdot x) - \frac{1}{2 \cdot 2 \cdot x^2} - \frac{1 \cdot 3}{2 \cdot 4 \cdot 4 \cdot x^4} - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6 \cdot 6 \cdot x^6} - \dots, \quad x > 1$$

Вариант 27.

$$\sin^3(x) = \frac{1}{4} \cdot \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{3^{2n+1} - 3}{(2 \cdot n + 1)!} \cdot x^{2n+1} = \frac{1}{4} \left[\frac{3^3 - 3}{3!} \cdot x^3 - \frac{3^5 - 3}{5!} \cdot x^5 + \frac{3^7 - 3}{7!} \cdot x^7 - \dots \right], \quad x < 1$$

Вариант 28.

$$\cos^3(x) = \frac{1}{4} \cdot \sum_{n=0}^{\infty} (-1)^n \cdot \frac{3^{2n} + 3}{(2 \cdot n)!} \cdot x^{2n} = \frac{1}{4} \left[\frac{3^0 + 3}{0!} \cdot x^0 - \frac{3^2 + 3}{2!} \cdot x^2 + \frac{3^4 + 3}{4!} \cdot x^4 - \dots \right], \quad x < 1$$

Вариант 29.

$$\ln \left[\frac{x+1}{x} \right] = 2 \cdot \sum_{n=1}^{\infty} \frac{1}{(2 \cdot n - 1) \cdot (2 \cdot x + 1)^n} = 2 \cdot \left[\frac{1}{(2 \cdot x + 1)} + \frac{1}{3 \cdot (2 \cdot x + 1)^3} + \frac{1}{5 \cdot (2 \cdot x + 1)^5} + \dots \right]^2, \quad (2 \cdot x + 1)^2 > 1$$

Вариант 30.

$$\ln(x) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1} \cdot (x-1)^{(2n+1)}}{n} = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} + \dots, \quad 0 < x \leq 2$$

Содержание отчёта

6. Название, цель работы, номер варианта и условие (Написать, используя вставку «Формула» в текстовом редакторе).
7. Представить программу, содержащую три части (с комментариями).
8. Примеры («скриншоты») работы программы, подтверждающие правильность решения.
9. Выводы по работе.

Лабораторная работа № 8

ИСПОЛЬЗОВАНИЕ МАССИВОВ

Цель работы: изучить базовые принципы работы с массивами числовых данных.

Задание и порядок выполнения лабораторной работы

Ввести с клавиатуры целое число, которое будет задавать размерность матрицы. Задать матрицу случайных вещественных чисел (числа должны быть положительными и отрицательными). Провести преобразования по варианту (если дополнительно не указано, сравнение проводить по абсолютной величине). Выводить на экран промежуточные значения.

Вариант 1.

В заданной матрице поменять местами первую строку и строку, содержащую наибольший элемент матрицы.

Вариант 2.

В заданной матрице поменять местами последний столбец и столбец, содержащий наименьший элемент матрицы.

Вариант 3.

В заданной матрице поменять местами последний столбец со столбцом, содержащим наибольший элемент матрицы.

Вариант 4.

В заданной матрице поменять местами последнюю строку со строкой, содержащей наименьший положительный элемент матрицы.

Вариант 5.

В заданной матрице поменять местами последнюю строку со строкой, содержащей минимальный положительный элемент матрицы.

Вариант 6.

В заданной матрице поменять местами последний столбец и столбец, содержащий минимальный элемент матрицы.

Вариант 7.

В заданной матрице поменять местами минимальный элемент главной диагонали и максимальный элемент побочной диагонали.

Вариант 8.

В заданной матрице поменять местами последнюю строку со строкой, содержащей максимальный элемент матрицы.

Вариант 9.

В заданной матрице поменять местами последний столбец и столбец, содержащий наименьший положительный элемент матрицы.

Вариант 10.

В заданной матрице поменять местами первую строку и строку, содержащую наибольший отрицательный элемент матрицы.

Вариант 11.

В заданной матрице поменять местами столбец, содержащий максимальный отрицательный элемент матрицы, и столбец, содержащий минимальный положительный элемент матрицы.

Вариант 12.

В заданной матрице поменять местами последний столбец и столбец, содержащий минимальный элемент матрицы.

Вариант 13.

В заданной матрице поменять местами вторую строку и строку, содержащую наибольший элемент матрицы.

Вариант 14.

В заданной матрице поменять местами строку, содержащую минимальный элемент матрицы, и строку, содержащую хотя бы один нулевой элемент матрицы.

Вариант 15.

В заданной матрице поменять местами последний столбец и столбец, содержащий наибольший элемент матрицы.

Вариант 16.

В заданной матрице поменять местами третий столбец со столбцом, содержащим наибольший элемент матрицы.

Вариант 17.

В заданной матрице поменять местами первую строку со строкой, содержащей наименьший элемент матрицы.

Вариант 18.

В заданной матрице поменять местами последнюю строку со строкой, содержащей наименьший отрицательный элемент матрицы.

Вариант 19.

В заданной матрице поменять местами первую строку со строкой, содержащей максимальный отрицательный элемент матрицы.

Вариант 20.

В заданной матрице поменять местами первый столбец и столбец, содержащий максимальный элемент матрицы.

Вариант 21.

В заданной матрице поменять местами максимальный элемент главной диагонали и максимальный элемент побочной диагонали.

Вариант 22.

В заданной матрице поменять местами строку, содержащую максимальный элемент матрицы, и строку, содержащую минимальный элемент матрицы.

Вариант 23.

В заданной матрице поменять местами последний столбец и столбец, содержащий максимальный положительный элемент матрицы.

Вариант 24.

В заданной матрице поменять местами первую строку и строку, содержащую наименьший отрицательный элемент матрицы.

Вариант 25.

В заданной матрице поменять местами столбец, содержащий максимальный положительный элемент матрицы, и столбец, содержащий минимальный положительный элемент матрицы.

Вариант 26.

В заданной матрице поменять местами строку, содержащую максимальный элемент матрицы, и строку, содержащую минимальный элемент матрицы.

Вариант 27.

В заданной матрице поменять местами первый столбец и столбец,

содержащий минимальный элемент матрицы.

Вариант 28.

В заданной матрице поменять местами последнюю строку со строкой, содержащей максимальный элемент матрицы.

Вариант 29.

В заданной матрице поменять местами третью строку и строку, содержащую наибольший элемент матрицы.

Вариант 30.

В заданной матрице поменять местами третий столбец со столбцом, содержащим наибольший элемент матрицы.

Содержание отчёта

10. Название, цель работы, номер варианта и условие.

11. Представить программу с комментариями, реализующую задачу.

12. Примеры («скриншоты») работы программы (матрицы различной размерности и значениями), подтверждающие правильность решения.

13. Выводы по работе.

Теоретическая часть

Работа с массивами

Массив — это последовательность данных *одинакового типа*, имеющих общее *имя* массива. Элементы (компоненты) массива расположены в памяти друг за другом и различаются номерами (*индексами*). Причем индексация начинается с *нуля*. Количество *индексов*, необходимых для описания массива и организации доступа к элементам массива, определяют *размерность* массива — *одномерный* (в математике — вектор), *двухмерный* (в математике — матрица, таблица) и т.д.

Описание массива

Описание массива должно соответствовать формату:

тип имя [размер1] [размер2]...[размерK];

причем скобки [] обязательны для каждой размерности.

Примеры.

```
int ar[5];           /*одномерный массив из 5-ти целых чисел*/  
char st [20], buf[80]; /*символьные массивы*/  
float R[2][3];     /*двухмерный массив вещественных чисел*/
```

В математике массив **R** представляется матрицей из 2-х строк и 3-х столбцов:

$$R_{2 \times 3} = \begin{bmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \end{bmatrix}$$

В памяти ЭВМ двухмерные массивы заполняются *построчно*:

$R[2][3] = \{R[0][0], R[0][1], R[0][2], R[1][0], R[1][1], R[1][2], R[1][3]\}$.

По всем размерностям «счет» идет от 0, до (N-1), где N — число элементов по данной размерности.

Занесение данных в массив

Исходные данные в элементы массива можно занести тремя способами:

- инициализацией массива;
- функцией ввода;
- присваиванием значений.

Инициализация массива выполняется при *описании* массива путем задания начальных значений элементам в списке инициаторов, заключенном в фигурные скобки { }, а сами инициаторы разделяются запятыми.

Одномерный массив можно инициировать полностью: **int** ar [5]={1,2,3,4,5} или частично: **int** ar[5]={4,5}; тогда ar[0]=4, ar[1]=5, остальные элементы равны 0, поэтому легко выполнить *обнуление* всего массива: **int** ar[5]={0}.

Двухмерный массив можно инициировать:

одним списком: **int** ar [2][3]={1,2,3, 4,5,6};

отдельными списками построчно: **int** ar [2][3]={ {1,2,3}, {4,5,6} };

неполными списками: **int** ar [2][3]={ {1,2}, {4} };

что соответствует списку { {1,2,0}, {4,0,0} }.

Ввод данных в массив функцией scanf() выполняется *поэлементно*, для чего используется оператор цикла (ввод одномерного массива: **for** (i=0; i<5; i++) **scanf**("%d",&ar[i])).

Присваивание значений элементам массива выполняется при вычислении *выражений*, определяющих элементы массива.

ИСПОЛЬЗОВАНИЕ МОДУЛЬНОГО ПРОГРАММИРОВАНИЯ

Цель работы: изучение создания вспомогательных функций в C/C++ и обращение к ним.

Задание и порядок выполнения лабораторной работы

Ввести с клавиатуры 2 (3) вещественных числа, вычислить значения u и v (x, y, z) для каждой функции $f(u,v)$ ($f(x,y,z)$); определить вид функций $f(u,v)$ и вычислить значение, найти результат Z . Расчет функции $f(u,v)$ описать отдельной функцией. Выводить промежуточные результаты.

Вариант 1.

Вычислить $Z = f(\sin^2(a), b - 1) + f(\sin(a) - \cos(b), b^2 - 1) + 1,2$, где

$$f(u,v) = \begin{cases} u + \sin(v), & u > 0 \\ u + \cos(v), & u < 0, v > 0. \\ |u + v|, & u < 0, v \leq 0 \end{cases}$$

Вариант 2.

Вычислить $Z = f(\sqrt{|x|} + y, x) + f(e^x, |y|)$, где $f(u,v) = \begin{cases} \ln(|u|) + \ln(|v|), & u \cdot v > 0 \\ |u + v|, & u \cdot v \leq 0 \end{cases}$.

Вариант 3.

Вычислить $Z = f(\sqrt{|x|} + 1, 1 - y) + f(|x| - |y|, |x - y|)$, где

$$f(u,v) = \begin{cases} u + 2 \cdot v, & u \geq 0 \\ u + v, & u \leq 1 \\ u^2 + 2 \cdot v + 1, & -1 < u < 0 \end{cases}$$

Вариант 4.

Вычислить $Z = f(\sin(x) + \cos(x), x + y) + f(\sin^3(x) + \cos^3(x), x^3 + y^3)$, где

$$f(u,v) = \begin{cases} u + v, & u > 1 \\ u - v, & 0 \leq u \leq 1. \\ v - u, & u < 0 \end{cases}$$

Вариант 5.

Вычислить $Z = f(\sin(|x|) + \cos(|y|), |y|) + f(\sqrt{|x|}, \sin(|y|))$, где

$$f(u,v) = \begin{cases} |u| \cdot |v|, & u > 0 \\ u^2, & u \leq 0, v > 0. \\ v^2, & u \leq 0, v \leq 0 \end{cases}$$

Вариант 6.

Вычислить $Z = f(\sin(|x + y|), \cos(|x + y|)) + 2 \cdot f(|x + y|, |x - y|)$, где

$$f(u, v) = \begin{cases} u + v, & u > 0, v > 0 \\ |u + v|, & u > 0, v \leq 0 \\ |u - v|, & u \leq 0, v > 0 \\ |u| + |v|, & u \leq 0, v \leq 0 \end{cases}$$

Вариант 7.

Вычислить $Z = f(\sqrt[3]{|x| + |y|}, \sqrt{|x + y|}) + f(\lg(|x| + |y|), \lg(|x + y|))$, где

$$f(u, v) = \begin{cases} u \cdot v, & u > 0, v > 0 \\ |u + v|, & u > 0, v \leq 0 \\ |u| + u \cdot v, & u \leq 0, v > 0 \\ |v| + u \cdot v, & u \leq 0, v \leq 0 \end{cases}$$

Вариант 8.

Вычислить $Z = \ln(f^2(a \cdot b, a + b, a - b) + f^2(2 \cdot a, 2 \cdot b, a \cdot b))$, где

$$f(x, y, z) = \begin{cases} x \cdot y \cdot z, & x \cdot y \cdot z > 0 \\ x + y + z, & x \cdot y \cdot z < 0 \\ x^2 + y^2 + z^2, & x \cdot y \cdot z = 0 \end{cases}$$

Вариант 9.

Вычислить $Z = f(|a| + |b|, |a + b|) - 2 \cdot f(|a| - |b|, |a \cdot b|)$, где

$$f(u, v) = \begin{cases} u^2 + v^2, & u \cdot v > 0 \\ \cos(u) + \cos(v), & u \cdot v < 0 \\ u + v, & u \cdot v = 0 \end{cases}$$

Вариант 10.

Вычислить $Z = f(\sin(a), \cos(a)) \cdot \sqrt{f(a^2, e^a)}$, где

$$f(u, v) = \begin{cases} \arcsin(u + v), & u + v > 0 \\ \arccos(u + v), & u + v < 0 \\ u^2 + v^2, & u + v = 0 \end{cases}$$

Вариант 11.

Вычислить $Z = f(\sqrt{a^2 + b^2}, \sqrt{|a^2 - b^2|}) \cdot f(\sqrt{|a + b|}, \sqrt{|a - b|})$, где

$$f(u, v) = \begin{cases} |u| + |v|, & u \cdot v > 0 \\ u^2 + v^2, & u \cdot v < 0 \\ e^u + e^v, & u \cdot v = 0 \end{cases}$$

Вариант 12.

Вычислить $Z = \sqrt{f(\sqrt{|x| + |y|}, \sqrt{|x \cdot y|}) \cdot f(\ln(|x| + |y|), \ln(|x \cdot y|))}$, где

$$f(u, v) = \begin{cases} \sin(u) \cdot \sin(v), & u + v > 0 \\ \cos(u) \cdot \cos(v), & u + v < 0 \\ u^2 + v^2, & u + v = 0 \end{cases}$$

Вариант 13.

Вычислить $Z = f\left(\exp\left(\frac{1.5}{(x+y)}\right), \exp\left(\frac{2}{((x)+(y))}\right)\right) + 2 \cdot f(\log_2((x+y)), \log_2((x \star y)))$, где

$$f(u, v) = \begin{cases} \sqrt{u^2 + v^2}, & u + v > 0 \\ \frac{1}{u + v}, & u + v < 0 \\ u \cdot v, & u + v = 0 \end{cases}.$$

Вариант 14.

Вычислить $Z = f(\sqrt{x+y}, \sqrt{x \cdot y}, x+y) \cdot f(\lg(x+y), f(\lg(x \cdot y), x-y))$, где

$$f(u, v) = \begin{cases} \sin^2(u \cdot v), & u + v > 0 \\ \cos^2(u \cdot v), & u + v < 0 \\ u^2 - v^2, & u + v = 0 \end{cases}.$$

Вариант 15.

Вычислить $Z = \sin(f(f(a+b, a-b), f(a^2, b^2)))$, где $f(u, v) = \begin{cases} u^2 + v^2, & u + v < 0 \\ u^2 - v^2, & u + v \geq 0 \end{cases}$.

Вариант 16.

Вычислить $Z = \cos(f(f(a^2 - 1, b), f(a - b, b)) + f(a^2, b^2 + 0, 2))$, где

$$f(u, v) = \begin{cases} (u + v), & u \cdot v \leq 0 \\ (u - v), & u \cdot v < 0 \end{cases}$$

Вариант 17.

Вычислить $Z = \arctg(f(\sin^2(a), \cos^2(a)) + f(2 \cdot \sin(a), 2 \cdot \cos(a)))$, где

$$f(u, v) = \begin{cases} u + v, & u \geq 0 \\ v, & u < 0, v > 0 \\ v^2, & u < 0, v \leq 0 \end{cases}$$

Вариант 18.

Вычислить $Z = f(\sin(|x|) + \cos(|y|), \cos(|y|)) + f(\sqrt{|x|}, \sin(|y|))$, где

где

$$f(u, v) = \begin{cases} |u| \cdot |v|, & u > 0 \\ u^2, & u < 0, v > 0 \\ v^2, & u = 0, v < 0 \end{cases}$$

Вариант 19.

Вычислить $Z = f(\sin(x) + x, x + \cos(y)) + f(\sin(x) + \cos^3(x), x^3 + y)$, где

$$f(u, v) = \begin{cases} u + v, & u + v > 1 \\ u - v, & 0 \leq u + v \leq 1 \\ v - u, & u < 0 \end{cases}$$

Вариант 20.

Вычислить $Z = \sqrt{|f(\sqrt{|x|+|y|}, \sqrt{|x \cdot y|})| + \sqrt{(f(\ln(|x|+|y|), \ln|x \cdot y|))}}$, где

$$f(u, v) = \begin{cases} \sin(u) \cdot \sin(v), & u + v < 0 \\ \cos(u) \cdot \cos(v), & u + v > 0. \\ u^2 + v^2, & u + v = 0 \end{cases}$$

Вариант 21.

Вычислить $Z = f(f(|a|+|b|, |a+b|), 2 \cdot f(|a|-|b|, |a \cdot b|))$, где

$$f(u, v) = \begin{cases} u^2 + v^2, & u \cdot v < 0 \\ \cos(u) + \cos(v), & u \cdot v > 0. \\ u + v, & u \cdot v = 0 \end{cases}$$

Вариант 22.

Вычислить $Z = f(\sin(|x-y|), \cos(|x+y|)) + 2 \cdot f(\cos(|x+y|), \sin(|x-y|))$, где

$$f(u, v) = \begin{cases} u + v, & u > 0, v > 0 \\ |u - v|, & u > 0, v \leq 0 \\ |u + v|, & u \leq 0, v > 0 \\ |u| - |v|, & u \leq 0, v \leq 0 \end{cases}$$

Вариант 23.

Вычислить $Z = \sqrt{|f(\sqrt{|a|+|b|}, \sqrt{|a-b|}, \sqrt{|a|-|b|})| + 1g(f(1g(|a|+|b|), 1g(|a-b|), 1g(|a|-|b|)))}$,

где

$$f(x, y, z) = \begin{cases} \ln(x) \cdot \ln(y) \cdot \ln(z), & x + y + z < 0 \\ \cos(x) \cdot \cos(y) \cdot \sin(z) & x + y + z > 0. \\ x^2 + y^2 + z^2, & x + y + z = 0 \end{cases}$$

Вариант 24.

Вычислить $Z = \arctg(f(f(a+b, a-b)), f(a^2, b^2))$, где

$$f(u, v) = \begin{cases} \cos(u^2 + v^2), & u \cdot v < 0 \\ \cos(u^2 - v^2), & u \cdot v \geq 0 \end{cases}$$

Вариант 25.

Вычислить $Z = f(\sin(a), \cos(a)) \cdot \sqrt{|f(a^2, e^a)|}$, где

$$f(u, v) = \begin{cases} \sqrt{u^2 + v^2}, & u + v > 0 \\ \frac{v}{u+v}, & u + v < 0 \\ \frac{u \cdot v}{u}, & u + v = 0 \end{cases}$$

Вариант 26.

Вычислить $Z = f(\sin(|x+y|) + \cos(|x-y|), |x|+|y|) + 2 \cdot f(\sqrt{|x-y|}, \sin(|x+y|))$, где

$$f(u, v) = \begin{cases} |u| \cdot |v|, & u - v > 0 \\ u^2 + v^2, & u \leq 0, v > 0. \\ u^2 - v^2, & u \leq 0, v \leq 0 \end{cases}$$

Вариант 27.

Вычислить $Z = f(f(\sin(\sqrt{|x-y|}) + \cos(|y|), |y|), f(\sqrt{|x|}, (|x| + |y|)))$, где

$$f(u, v) = \begin{cases} |u - v| \cdot |v|, & u > 0 \\ v - u^2, & v > 0 \\ u + v^2, & u \leq 0, v \leq 0 \end{cases}.$$

Вариант 28.

Вычислить $Z = \sqrt{f(\cos(|x|), |y|) + f(\sqrt{|x|}, \sin(|x-y|))}$, где

$$f(u, v) = \begin{cases} u + v, & u > 0 \\ 2 \cdot u^2, & u \leq 0, v \geq 0. \\ \frac{v^2}{u}, & u < 0, v < 0 \end{cases}$$

Вариант 29.

Вычислить $Z = f(\operatorname{tg}(|x|), (x - y)) + f(\cos(3 \cdot x), \operatorname{tg}(|y|))$, где

$$f(u, v) = \begin{cases} \ln(|u|) \cdot |v|, & u > 0 \\ \cos(u^2), & u \leq 0, v > 0. \\ v^2, & u \leq 0, v \leq 0 \end{cases}$$

Вариант 30.

Вычислить $Z = \sqrt{f(\cos(|x+y|), \sqrt[3]{|y|}) + f(x^2, \sin(|2 \cdot y|))}$, где

$$f(u, v) = \begin{cases} |u| \cdot |v|, & u - v > 0 \\ u + v, & u \leq 0, u - v > 0. \\ (u - v)^2, & u \leq 0, v \leq 0 \end{cases}$$

Содержание отчёта

1. Название, цель работы, номер варианта и условие (Написать, используя вставку «Формула» в текстовом редакторе).
2. Нарисовать алгоритм, используя predetermined процесс (описать отдельно вспомогательные функции).
3. Представить программу, содержащую вспомогательную функцию (с комментариями), реализующую задачу.
4. Тестовый пример — «скриншот» и ручной счет.
5. Примеры («скриншоты») работы программы, реализующие все возможные пути решения.

6. Выводы по работе.

Теоретическая часть

Создание вспомогательных функций

Разработка больших программ осуществляется с использованием принципа *модульного программирования*, суть которого состоит в том, что алгоритм задачи разбивается на ряд подзадач. Этот процесс называют *функциональной декомпозицией* задачи или принципом "разделяй и властвуй". Для реализации алгоритмов подзадач на ЯВУ разрабатываются вспомогательные программные единицы, называемые *подпрограммами*. Для решения задачи *в целом* в программе, называемой *главной* (основной), организуются *обращения* (вызовы) к подпрограммам. Подпрограммы делятся на *стандартные*, которые находятся в библиотеках подпрограмм на ЭВМ, и *пользовательские*, разрабатываемые самим пользователем.

В языках С/С++ подпрограммы называются функциями. *Функция* — это логически самостоятельная *именованная* часть программы, которой могут *передаваться* параметры и которая может *возвращать* какие-то результаты. С-программа состоит из одной или нескольких функций, причем только *одна* является *обязательной* и определяется как *главная* функция с именем **main()**. С этой функции начинается исполнение программы. Последовательность функций может быть произвольной. Функции могут вызывать друг друга (кроме функции **main**).

Однако для успешной компиляции программы должно соблюдаться правило: *вызову* функции предшествует либо определение (описание) функции, задающее информацию о заголовке функции. Синтаксис языка С запрещает внутри определения функции помещать определение другой функции, то есть *не может* быть вложенных функций.

Определение функции

Описание функции имеет следующую структуру:

```
тип имя_функции (список параметров)    /*заголовок функции*/  
{                                       /*тело функции*/  
    описания данных;                    /*необязательная часть*/
```

```
    операторы;                               /*обязательная часть*/  
}
```

Рассмотрим составляющие заголовка функции:

тип — задает *тип* возвращаемого функцией значения; если функция *не* возвращает значения, то используется ключевое слово **void** (отсутствие типа).

имя_функции (произвольный идентификатор) — это особый тип указателя-константы, называемого *указателем на функцию*, и его значением является *адрес* точки входа в функцию.

список параметров — задает последовательность *формальных параметров*, разделяемых запятыми. Каждый параметр определяется его *типом* и *именем* с возможными модификаторами, описывающими: массив — [], функцию — (). При отсутствии параметров список может быть *пустым*, либо содержит ключевое слово **void**.

В конце заголовка функции разделитель точка с запятой НЕ ставится.

Совокупность предложений в фигурных скобках {...} называют *телом* функции. Тело функции образует *блок* (или *составной оператор*), определяющий действия функции и её результаты. Блок может содержать описания переменных, используемых в этом блоке, поэтому эти переменные являются *локальными*, если не объявлено иначе. Операторы тела функции выполняются, пока не встретится оператор **return** (возврат) или конец функции (последняя закрывающая фигурная скобка — «}»), при этом управление возвращается в точку *вызова* функции.

Функция возвращает значение, если встретится *оператор возврата* вида **return** выражение (или **return** (выражение)), при этом значение выражения приводится к *типу* функции. Если оператор **return** не содержит выражения или отсутствует, то значение не возвращается, а только передается управление в точку *вызова* функции с типом **void**.

Вызов функции

Обращение к функции имеет вид:

имя_функции (список аргументов)

список аргументов — последовательность выражений, разделенных запятыми, представляющих фактические параметры.

Возможны два способа вызова функции:

- Вызов функции – *операнд*. В этом случае значение, возвращаемое функцией, используется в выражении, где функция участвует как операнд.
- Вызов функции – *оператор*. Если вместо типа функции стоит слово `void` или возвращаемое значение функции игнорируется, а используются только *фактические* параметры, то вызов функции задается как *отдельный* оператор.

Лабораторная работа № 10

ИСПОЛЬЗОВАНИЕ ВСПОМОГАТЕЛЬНЫХ ФУНКЦИЙ И МАССИВОВ

Цель работы: закрепить навыки работы с массивами и навыки модульного программирования.

Задание и порядок выполнения лабораторной работы

Заданы 3 массива случайных вещественных чисел, определенной размерности: А, В, С. Необходимо найти значение М при помощи вычисления вспомогательных функций. Задание массивов, вывод их на печать и вычисление вспомогательных выражений оформить в виде отдельных функций. Выводить на экран промежуточные значения.

Вариант 1.

Размерность всех массивов — $n \times n$; $M = \sqrt{S_A^2 + S_B^2 + S_C^2}$, где $S_A = \sum_{i=1}^n a_{ii}$.

Вариант 2.

Размерность (n) массивов: k , $2 \times k$, $2 \times k - 3$; $M = S_A + S_B + S_C$, где

$$S_A = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (a_i - P_A)^2}, P_A = \frac{1}{n} \cdot \sum_{i=1}^n a_i.$$

Вариант 3.

Размерность всех массивов — $n \times n$; $M = \sqrt{S_A^2 + S_B^2 + S_C^2}$, где

$$S_A = \max(a_{ij}) - \min(a_{ij}).$$

Вариант 4.

Размерность всех массивов — $n \times n$; $M = (A + B^2)^2 + (B + C^2)^2$.

Вариант 5.

Размерность всех массивов — n ; $M = (S_{AB} + S_{BC} + S_{CA})^2$, где $S_{AB} = \sum_{i=1}^n (a_i * b_i)$.

Вариант 6.

Размерность всех массивов — $n \times n$; $M = A^2 \cdot B^2 \cdot C^2$.

Вариант 7.

Размерность всех массивов — $n \times n$; $M = (S_A + S_B + S_C)^2$, где $S_A = \max_j \sum_{i=1}^n a_{ij}$.

Вариант 8.

Размерность (n) массивов: $k, 2*k, 2*k-3$; $M = \sqrt[3]{S_A \cdot S_B \cdot S_C}$, где $S_A = \sum_{i=1}^n a_i$.

Вариант 9.

Размерность всех массивов — $n \times n$; $M = (S_A + S_B)^2 + S_C^2$, где $S_A = A \cdot A^T$.

Вариант 10.

Размерность всех массивов — $n \times n$; $M = (S_A + S_B) \cdot S_C$, где $S_A = \min_{ij}(a_{ij})$.

Вариант 11.

Размерность (n) массивов: $k, 2*k, 2*k-3$; $M = e^{-(S_A + S_B + S_C)}$, где $S_A = \sum_{i=1}^n a_i$.

Вариант 12.

Размерность всех массивов — $n \times n$; $M = \sqrt{(S_A + S_B)^2 + S_C^2}$, где $\sum_{i=1}^n a_{ij}$.

Вариант 13.

Размерность (n) массивов: $k, 2*k, 2*k-3$; $M = \sqrt[4]{S_A + S_B + S_C}$, где $S_A = \sum_{i=1}^n a_i$.

Вариант 14.

Размерность всех массивов — $n \times n$; $M = \sqrt{S_A^2 \cdot S_B^2 + S_C^2}$, где $S_A = \min_j \sum_{i=1}^n a_{ij}$.

Вариант 15.

Размерность (n) массивов: $k, 2*k, 2*k-3$; $M = k^{-(S_A + S_B + S_C)}$, где $S_A = \sum_{i=1}^n a_i$.

Вариант 16.

Размерность всех массивов — $n \times n$; $M = (A^2 + B^2) \cdot C^2$.

Вариант 17.

Размерность всех массивов — n ; $M = (S_{AB} \cdot S_{BC} + S_{CA})^2$, где $S_{AB} = \sum_{i=1}^n (a_i + b_i)$.

Вариант 18.

Размерность всех массивов — $n \times n$; $M = A^3 + (B + C)^3$.

Вариант 19.

Размерность (n) массивов: $k, 2*k, 2*k-3$; $M = e^{\sqrt{S_A + S_B + S_C}}$, где $S_A = \sum_{i=1}^n a_i$.

Вариант 20.

Размерность всех массивов — n ; $M = \sqrt[3]{S_{AB} \cdot S_{BC} \cdot S_{CA}}$, где $S_{AB} = \sum_{i=1}^n (a_i \cdot b_i)$.

Вариант 21.

Размерность всех массивов — $n \times n$; $M = n \sqrt{S_A + S_B + S_C}$, где $S_A = \min_i \sum_{j=1}^n a_{ij}$.

Вариант 22.

Размерность всех массивов — $n \times n$; $M = \sqrt[3]{(S_A + S_B) \cdot S_C}$, где $S_A = \max(a_{ij}) - \min(a_{ij})$.

Вариант 23.

Размерность всех массивов — $n \times n$; $M = \sqrt{S_A^2 + S_B^2 + S_C^2}$, где $S_A = \frac{1}{n} \cdot \sum_{i,j=1}^n a_{ij}$.

Вариант 24.

Размерность (n) массивов: $\kappa, 2 \cdot \kappa, 2 \cdot \kappa - 3$; $M = (S_A + S_B)^2 + S_C^2$, где

$$S_A = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (a_i - P_A)^2}, \quad P_A = \max(a_i) - \min(a_i).$$

Вариант 25.

Размерность всех массивов — n ; $M = k^{-(S_{AB} + S_{BC} + S_{CA})}$, где $S_{AB} = \sum_{i=1}^n (a_i \cdot b_i)$.

Вариант 26.

Размерность всех массивов — $n \times n$; $M = (A^2 + B)^2 \cdot (B^2 + C)^2$.

Вариант 27.

Размерность всех массивов — $n \times n$; $M = \sqrt[3]{(S_A + S_B)^2 + S_C^2}$, где

$$S_A = \sqrt{\sum_{j=0}^n (\max(a_{ij}) - \min(a_{ij}))^2}.$$

Вариант 28.

Размерность всех массивов — $n \times n$; $M = \sqrt{(S_A + S_B)^2 + S_C^2}$, где $S_A = \max_i \sum_{j=1}^n a_{ij}$.

Вариант 29.

Размерность всех массивов — $n \times n$; $M = \sqrt{S_A^2 + S_B + S_C^2}$, где

$$S_A = \max_j \sum_{i=1}^n a_{ij} - \min_i \sum_{j=1}^n a_{ij}.$$

Вариант 30.

Размерность (n) массивов: k , $2*k$, $2*k-3$; $M = k^{-(S_A+S_B+S_C)}$, где
 $S_A = \max(a_i) - \min(a_i)$.

Содержание отчёта

1. Название, цель работы, номер варианта и условие (Написать, используя вставку «Формула» в текстовом редакторе).
2. Составить алгоритм, с использованием предопределенного процесса.
3. Представить программу, содержащую вспомогательные функции (с комментариями), реализующую задачу.
4. Тестовый пример — «скриншот» и ручной счет.
5. Примеры («скриншоты») работы программы с различными исходными данными.
6. Выводы по работе.

ПРОГРАММИРОВАНИЕ ЗАДАЧ С НЕЧИСЛОВЫМИ ДАНЫМИ

Цель работы: изучить основные принципы работы с символьными переменными.

Задание и порядок выполнения лабораторной работы

Дана строка (не менее 7 слов). Словом текста является последовательность букв алфавита; между соседними словами — не менее одного пробела, за последним словом – точка. Заглавные и строчные буквы алфавита считать эквивалентными.

Вариант 1.

Найти и сохранить в строке те слова, в которых гласные буквы алфавита образуют симметричную последовательность букв (палиндром). Все остальные слова удалить.

Вариант 2.

Разместить слова строки в алфавитном порядке.

Вариант 3.

Найти и сохранить в строке те слова, в которых цифры и буквы латинского алфавита чередуются. Все остальные слова удалить.

Вариант 4.

Найти и сохранить в строке те слова текста, в которых есть хотя бы одна цифра. Все остальные слова удалить.

Вариант 5.

Найти и сохранить в строке те слова текста, которые содержат только гласные буквы алфавита. Все остальные слова удалить.

Вариант 6.

Найти и сохранить в строке те слова текста, которые образованы неубывающей последовательностью символов. Все остальные слова удалить.

Вариант 7.

Найти и сохранить в строке те слова, символы которых образуют симметричную последовательность букв (палиндром). Все остальные слова

удалить.

Вариант 8.

Удалить из строки те слова, которые содержат двойные согласные буквы.

Вариант 9.

Поменять местами в строке первое и последнее слово.

Вариант 10.

Найти и сохранить в строке те слова текста, которые содержат одинаковое количество гласных и согласных букв алфавита. Все остальные слова удалить.

Вариант 11.

Найти и сохранить в строке те слова текста, количество гласных букв в которых превышает количество согласных. Все остальные слова удалить.

Вариант 12.

Найти и сохранить в строке те слова, которые начинаются с гласной буквы. Все остальные слова удалить.

Вариант 13.

Найти и сохранить в строке те слова последовательности, которые отличны от последнего слова и являются симметричными. Все остальные слова удалить.

Вариант 14.

Найти и сохранить в строке те слова последовательности, которые отличны от первого слова и удовлетворяют следующему свойству: первая буква слова входит в него еще один раз. Все остальные слова удалить.

Вариант 15.

Найти и сохранить в строке те слова последовательности, которые отличны от последнего слова и удовлетворяют следующему свойству: слово совпадает с начальным отрезком латинского алфавита (a, ab, abc, abcd,...). Все остальные слова удалить.

Вариант 16.

Найти и сохранить в строке те слова последовательности, которые отличны от первого слова и удовлетворяют следующему свойству: слово совпадает с конечным отрезком латинского алфавита (z, yz, хyz,...). Все

остальные слова удалить.

Вариант 17.

Найти и сохранить в строке те слова последовательности, которые отличны от последнего слова и удовлетворяют следующему свойству: в слове нет повторяющихся букв. Все остальные слова удалить.

Вариант 18.

Найти и сохранить в строке те слова последовательности, которые отличны от первого слова и удовлетворяют следующему свойству: каждая буква входит в слово не менее двух раз. Все остальные слова удалить.

Вариант 19.

Найти и сохранить в строке те слова последовательности, которые отличны от последнего слова и удовлетворяют следующему свойству: в слове гласные буквы чередуются с согласными. Все остальные слова удалить.

Вариант 20.

Найти и сохранить в строке те слова последовательности, которые отличны от первого слова, предварительно преобразовав каждое из них по следующему правилу: перенести первую букву в конец слова. Все остальные слова удалить.

Вариант 21.

Найти и сохранить в строке те слова последовательности, которые отличны от последнего слова, предварительно преобразовав каждое из них по следующему правилу: перенести последнюю букву в начало слова. Все остальные слова удалить.

Вариант 22.

Найти и сохранить в строке те слова последовательности, которые отличны от первого слова, предварительно преобразовав каждое из них по следующему правилу: удалить из слова первую букву. Все остальные слова удалить.

Вариант 23.

Найти и сохранить в строке те слова последовательности, которые отличны от последнего слова, предварительно преобразовав каждое из них по следующему правилу: удалить из слова последнюю букву. Все остальные слова

удалить.

Вариант 24.

Найти и сохранить в строке те слова последовательности, которые отличны от первого слова, предварительно преобразовав каждое из них по следующему правилу: удалить из слова все последующие вхождения первой буквы. Все остальные слова удалить.

Вариант 25.

Найти и сохранить в строке те слова последовательности, которые отличны от последнего слова, предварительно преобразовав каждое из них по следующему правилу: удалить из слова все предыдущие вхождения последней буквы. Все остальные слова удалить.

Вариант 26.

Найти и сохранить в строке те слова, которые встречаются в последовательности по одному разу. Все остальные слова удалить.

Вариант 27.

Найти и сохранить в строке те слова последовательности, которые отличны от последнего слова, предварительно преобразовав каждое из них по следующему правилу: если слово нечетной длины, то удалить его среднюю букву. Все остальные слова удалить.

Вариант 28.

Разместить в строке последовательность ее слов в обратном порядке.

Вариант 29.

Найти и сохранить в строке те слова, перед которыми в последовательности находятся только меньшие (по алфавиту) слова, а за ними только большие. Все остальные слова удалить.

Вариант 30.

Сохранить в строке последовательность слов, удалив из нее повторные вхождения слов.

Содержание отчёта

1. Название, цель работы, номер варианта и условие.
2. Представить программу, с комментариями, реализующую задачу.

3. Примеры («скриншоты») работы программы, подтверждающие правильность решения.
4. Выводы по работе.

Теоретическая часть

Ввод-вывод символов

Для ввода-вывода символов и строк можно использовать соответствующие функции стандартного ввода-вывода (*stdio.h*).

Функция ввода символов (без параметров):

getchar() — определяет ASCII-код нажатой клавиши, записывает его в буфер ввода *stdin*, а после нажатия клавиши <Enter> передает в оперативную память для ввода в переменную, описанную с типом **int** или **char**

Функция вывода символа:

putchar(ch) — преобразует ASCII-код, записанный в переменной *ch* (тип **int** или **char**), в символ, выводимый на экран.

Функция ввода строки символов:

gets(string),

string — указатель (адрес) для ввода строки символов, например, имя массива символов.

В отличие от функции **scanf**(“%s”, *string*), которая вводит символы до первого пробела, функция **gets**(*string*) вводит все символы, включая пробелы.

Функция ввода строки символов

puts(string)

string — либо сама строка, либо указатель (адрес) области памяти, откуда выводится строка, например, имя массива символов.

Эта функция аналогична функции **printf**(“%s”, *string*), но при этом **puts** автоматически выполняет перевод курсора экрана на новую строку.

Консольный ввод-вывод

Функции консольного ввода-вывода не используют входной буфер *stdin* для ввода данных. Все символы, вводимые с клавиатуры, доступны программе немедленно, что предоставляет дополнительные возможности при обработке данных в программе, при использовании директивы препроцессора:

#include<conio.h>

Функция ввода символа (без отображения):

getch() — прочитывает код символа нажатой клавиши и *не* отображает символ на экране.

Это свойство можно использовать в двух случаях:

➤ Для ввода невидимых символов в память и использования их для проверки пароля программы:

➤ Для останова выполнения программы и просмотра текущих результатов на экране данных:

Функция ввода символа (с отображением):

getche() — выполняет *небуферизуемый* ввод символа в отличие от функции **getchar()** и его отображение в отличие от функции **getch()**.

Символьные массивы

Описание массива типа **char string[80]** — позволяет описывать и обрабатывать символьные данные, однако тип **char[]** относится к целому типу, поскольку имеет дело с *ASCII-кодами* символов.

Особым случаем массива является *строковый литерал* (строка-константа) — последовательность любых символов, заключенных в кавычки: “это строковый литерал”. Строковый литерал представляется в памяти как массив элементов типа **char[]**, в конце которого помещен символ ‘\0’ (нуль-байт), поэтому его называют *ASCII-строкой* (от zero – нуль). Как и с любым массивом, со строковым литералом связан *указатель-константа* на первый элемент строки.

Символьные массивы можно инициировать, вводить и выводить поэлементно, как и числовые массивы, однако более рационально выполнять *инициирование* с помощью строковых литералов, а *ввод-вывод* — по ссылке на *имя* символьного массива, которое является указателем-константой на первый элемент массива (с индексом 0).

Совершенно идентичны следующие описания с инициализацией:

char string [10] = { ‘с’, ‘л’, ‘о’, ‘в’, ‘о’, ‘\0’ };

char string [10] = ”слово”;

Последнее описание выглядит проще, а нулевой байт записывается в конце строки *автоматически*. Адрес первого элемента ('с') становится адресом массива string. Строка в массиве ограничена нуль-байтом, а остальные элементы массива не используются. Можно не задавать размер массива, компилятор выделит нужное число элементов плюс нуль-байт при инициализации:

```
char string[]="слово";
```

Стандартные функции ввода-вывода можно использовать для обработки символьных данных.

Функция **scanf**("%s", name) менее удобна, поскольку вводит символы до *первого* пробела.

Особый случай — инициализация *двухмерного* массива строковыми литералами. Многомерные массивы могут инициироваться и без указания одной (самой левой) из размерностей массива в []. Компилятор сам определяет число элементов по числу членов в списке инициализации.

Пример,

```
char str [ ] [80] = {"Первая строка",  
                    "Вторая строка",  
                    "Третья строка"};
```

Как было показано выше, элементы массива str[0], str[1], str[2] являются *указателями* на строки двухмерного массива, что можно использовать при обработке таких массивов.

Проще выглядит копирование символов с помощью оператора цикла:

```
for ( i=0; s2[i]=s1[i]; i++ ); /*тело цикла – пустой оператор*/.
```


Лабораторная работа № 12

ИСПОЛЬЗОВАНИЕ ФАЙЛОВ

Цель работы: изучить принципы работы с внешними данными.

Задание и порядок выполнения лабораторной работы

Создается исходный текстовый файл (или несколько файлов). Необходимо создать еще файлы, в которых будут отражены результаты выполнения задания.

Вариант 1.

Сосчитать количество определенных символов из файла и записать результат в новый файл.

Вариант 2.

Определить код символа, стоящего на определенном месте в файле и записать результат в новый файл.

Вариант 3.

Поменять местами четные и нечетные слова файла и записать результат в новый файл.

Вариант 4.

Создать новый файл, являющийся объединением двух имеющихся файлов.

Вариант 5.

Найти в файле определенное сочетание символов, удалить их и записать результат в новый файл.

Вариант 6.

Заменить первый символ каждого слова файла на его код и записать результат в новый файл.

Вариант 7.

Разделить данные файла № 1 по определенному сочетанию символов и записать их соответственно в файлы № 2 и № 3 (до набора определенных символов в один файл, а остальную часть в другой).

Вариант 8.

Удвоить в файле каждое слово с определенным номером (например каждое 3 слово) и записать результат в новый файл.

Вариант 9.

Удалить из файла строку с определенным номером и записать результат в новый файл.

Вариант 10.

Поменять в файле местами две строки с определенными номерами и записать результат в новый файл.

Вариант 11.

Создать файл, являющийся обратным (обратный порядок всех символов) к имеющемуся файлу.

Вариант 12.

Создать новый файл, являющийся пересечением двух имеющихся файлов (набор одинаковых слов).

Вариант 13.

Разделить данные файла № 1 – слова длиннее и короче заданного с клавиатуры числа и записать их соответственно в файлы № 2 и № 3.

Вариант 14.

Поменять определенный символ, встречающийся в файле, на другой и записать результат в новый файл.

Вариант 15.

Разделить данные файла № 1 на две равные части и записать их соответственно в файлы № 2 и № 3.

Вариант 16.

Заменить последний символ каждого слова файла на его код и записать результат в новый файл.

Вариант 17.

Записать нечетные слова файла № 1 в файл № 2, а четные слова файла № 1 – в файл № 3.

Вариант 18.

Удалить из файла каждое слово с определенным номером (например каждое 3 слово) и записать результат в новый файл.

Вариант 19.

Найти в файле те слова текста, которые содержат одинаковое количество

гласных и согласных букв алфавита и записать их в новый файл.

Вариант 20.

Поменять в файле местами два слова с определенными номерами и записать результат в новый файл.

Вариант 21.

Удвоить из файла строку с определенным номером, введенным с клавиатуры и записать результат в новый файл.

Вариант 22.

Поменять попарно местами в файле все слова и записать результат в новый файл.

Вариант 23.

Создать файл, являющийся обратным (обратный порядок слов) к имеющемуся файлу.

Вариант 24.

Сосчитать количество слов в файле и записать результат в новый файл.

Вариант 25.

Разделить данные файла №1 на 3 (три) равные части и записать 1 и 3 части соответственно в файлы № 2 и № 3.

Содержание отчёта

1. Название, цель работы, номер варианта и условие.
2. Представить программу, с комментариями, реализующую задачу.
3. Примеры («скриншоты») работы программы, подтверждающие правильность решения
4. Выводы по работе.

Теоретическая часть

Работа с файлами

Понятие файла и потока ввода-вывода данных в компьютере

Для организации ввода-вывода и хранения информации используется файловая система. *Файл* — это именованный объект, который может хранить

данные, программу, результаты ее работы или любую другую информацию на внешних носителях информации. В языке Си нет встроенных операторов ввода-вывода, а есть *функции* ввода-вывода данных. При этом ввод-вывод данных рассматривается как *поток байтов*, поэтому термины *файл* и *поток* рассматриваются как равноправные.

Функции *потокowego (файлового) ввода-вывода* включены в стандарт ANSI языка Си, и поэтому их часто называют функциями *стандартного ввода-вывода*. Их прототипы находятся в заголовочном файле, подключаемом к программе командой препроцессора

```
#include<stdio.h>.
```

С помощью файла **stdio.h** открывается доступ к *библиотеке функций* стандартного ввода-вывода, которая содержит объектные модули этих функций, необходимые на стадии загрузки программы.

Для каждого файла, открытого в программе как поток ввода-вывода, создается внутри *структурная переменная* по шаблону FILE, определенному в **stdio.h**, где содержится информация о файле (флаги состояния файла, указатель на буфер, размер буфера и др.). Доступ к файлу осуществляется с помощью *указателя*, который описывается следующим образом:

```
FILE *fp;
```

Открытие и закрытие файла

Описанный *указатель* необходимо *связать* с конкретным *файлом* на диске (или устройством ввода-вывода) в момент открытия файла функцией `fopen ()` следующим образом:

```
fp = fopen (“путь_к_файлу”, ”тип_доступа”);
```

путь_к_файлу — *указатель* на строку символов (литерал), задающую спецификацию открытого файла. При этом, задавая путь_к_файлу с использованием символа ‘\’, его необходимо **удваивать** для того, чтобы он не воспринимался компилятором как элемент символа управления, например, “C:\\\$STUDENT\\GR_1234\\myfile1.txt”;

тип_доступа — *указатель* на строку символов, задающую *режим открытия* файла:

r — существующий файл открывается *только для чтения* (read);

w — открывается *существующий* файл (с уничтожением его содержимого) или создается *новый* файл для *записи* (write) ;

a — *существующий* файл открывается для пополнения конца файла (append – дополнение)

r+ — *существующий* файл открывается для *чтения* и *записи*;

w+ — создается файл для *чтения* и *записи*;

a+ — файл открывается для *пополнения* конца файла и доступен для *чтения* и *записи*;

t — файл открывается в *текстовом* режиме (text), этот режим принят по *умолчанию*;

b — файл открывается в *двоичном* режиме (binary).

В *текстовом* режиме при чтении информации из файла пара символов CR LF (возврат каретки, перевод строки) преобразуется в один символ “\n”, а при записи *новой* строки преобразуется в пару символов CR LF.

Функция **fopen()** возвращает *указатель (адрес)* на успешно *открытый* файл или NULL-указатель при невозможности открыть файл. Проверку открытия файла должен выполнять сам программист.

После *завершения* всех операций файлового доступа файл следует *закрыть* функцией вида

fclose(указатель файла);

Пример. Схема открытия и закрытия файла:

```
#include<stdio.h>
void main ( )
{ char *path= “C:\\$STUDENT\\GR_1234\\test.txt”; /* путь к файлу */
  FILE *myfile; /* указатель файла*/
  /****** Проверка открытия файла:******/
  if (( myfile = fopen ( path ,”w+”)) == NULL);
  { printf (“Ошибка при открытии файла % s “ , path );
    exit; /* аварийный выход из программы */
  }
  /*** ..... Работа с файлом myfile .....***/
  fclose (myfile); /* закрытие файла */
}
```

В программе может быть открыто *несколько* файлов в зависимости от установок операционной системы. Перед окончанием работы программы все открытые программистом файлы можно *закрыть* одной функцией

fcloseall();

Потоки стандартного ввода-вывода

В файле заголовков **stdio.h** заданы константы, используемые при работе с файлами:

NULL — нулевой байт, возвращаемый как NULL-указатель в случае ошибки при открытии файла;

EOF — значение конца файла (−1).

Закрытый файл можно *открыть* заново с *другим* типом доступа. Открытый файл можно *повторно* открыть и изменить права доступа.

Форматированный ввод-вывод данных

При *вводе* выполняется заданное *форматной строкой* преобразование байтов строки и присваивание полученных значений специфицированным переменным. При *выводе* выполняется обратное преобразование значений и передача их символьного представления в поток вывода. Функции форматированного ввода-вывода используют *переменное* количество аргументов. Функции стандартного и файлового вида работают аналогично, но последние должны задать рабочий *файл*.

Функция форматированного ввода вида

fscanf (*fr*, "форматная строка", &аргумент,...)

выполняет *ввод (чтение)* данных из открытого файла *fr* по *адресу* аргумента. Символы форматной строки задают способ преобразования вводимых байтов. Отдельное преобразуемое поле начинается с *текущей* позиции в буфере потока. Завершением поля считается символ *пробела* или *перевода строки*. Форматная строка обрабатывается слева направо.

Функция форматированного вывода вида

fprintf (*fr*, "форматная строка", аргумент,...)

выполняет *вывод (запись)* в открытый файл *fr* значение аргумента.

Символы форматной строки задают способ, в соответствии с которым выполняется преобразование чисел в последовательность символов. Полученная последовательность символов помещается в специфицированный поток. Все остальные символы строки формата передаются в поток без преобразования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Прокушев Л.А. Сравнительное изучение языков Си и Паскаль: Учебно-справочное пособие. С-Пб.: СПбГААП, 1997. 152 с.
2. Прокушев Л.А. Программирование на языке высокого уровня: Методические указания к выполнению лабораторных работ. С-Пб.: СПбГААП, 2009. 81 с.
3. Березин Б.И., Березин С.Б. Начальный курс С и С++. М.: Диалог-МИФИ, 2000. 288 с.
4. Решетов Л.А. Матричный проектор для суммы матриц, не удовлетворяющих условию ранговой аддитивности. Сборник докладов научной сессии ГУАП, Ч.1, СПб, ГУАП, 2014, с.279 – 282
5. Соколовская М.В. Информатика: Методические указания к выполнению лабораторных работ (для очной формы обучения) Часть 1. С-Пб.: СПбГААП, 2014. 56 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
Лабораторная работа №1.....	4
БАЗОВЫЕ НАВЫКИ РАБОТЫ В СИ.....	4
<i>Задание на лабораторную работу.....</i>	4
<i>Порядок выполнения работы.....</i>	5
Теоретическая часть.....	5
Средства программирования вычислительных процессов.....	5
Структура программы на С/С++.....	6
Базовые элементы языков С/С++.....	6
Данные и их типы.....	8
<i>Константы.....</i>	8
<i>Переменные.....</i>	8
Ввод-вывод данных.....	9
<i>Форматированный ввод.....</i>	9
<i>Форматированный вывод.....</i>	10
<i>Неформатированный ввод-вывод.....</i>	10
Лабораторная работа № 2.....	12
ИСПОЛЬЗОВАНИЕ УСЛОВНОГО ОПЕРАТОРА.....	12
<i>Задание и порядок выполнения лабораторной работы.....</i>	12
<i>Содержание отчёта.....</i>	15
Теоретическая часть.....	15
Выражения и операции.....	15
<i>Арифметические выражения и операции.....</i>	16
Работа с ветвящимися процессами.....	17
<i>Операторы.....</i>	17
<i>Условный оператор (if).....</i>	17
Лабораторная работа № 3.....	19
ИСПОЛЬЗОВАНИЕ ЛОГИЧЕСКОЙ ФУНКЦИИ.....	19
<i>Задание и порядок выполнения лабораторной работы.....</i>	19
<i>Содержание отчёта.....</i>	24
Теоретическая часть.....	24
Логические выражения и операции.....	24
Лабораторная работа № 4.....	26
МНОЖЕСТВЕННЫЙ ВЫБОР.....	26
<i>Задание на лабораторную работу.....</i>	26
<i>Порядок выполнения работы.....</i>	28
Теоретическая часть.....	28
<i>Оператор выбора (switch).....</i>	28
Лабораторная работа № 5.....	30
ИСПОЛЬЗОВАНИЕ ФУНКЦИИ ЦИКЛА.....	30
<i>Задание и порядок выполнения лабораторной работы.....</i>	30
<i>Содержание отчёта.....</i>	31
Теоретическая часть.....	31
Циклические вычислительные процессы.....	31
<i>Циклы с фиксированным числом повторений.....</i>	32
Лабораторная работа № 6.....	33

ВЫЧИСЛЕНИЕ ИНТЕГРАЛА.....	33
<i>Задание и порядок выполнения лабораторной работы.....</i>	33
<i>Содержание отчёта.....</i>	34
Лабораторная работа № 7.....	35
ИСПОЛЬЗОВАНИЕ РАЗЛИЧНЫХ ФУНКЦИЙ ЦИКЛА.....	35
<i>Задание и порядок выполнения лабораторной работы.....</i>	35
<i>Содержание отчёта.....</i>	37
Теоретическая часть.....	38
<i>Оператор цикла с постусловием.....</i>	38
Лабораторная работа № 8.....	39
ИСПОЛЬЗОВАНИЕ МАССИВОВ.....	39
<i>Задание и порядок выполнения лабораторной работы.....</i>	39
<i>Содержание отчёта.....</i>	42
Теоретическая часть.....	42
Работа с массивами.....	42
<i>Описание массива.....</i>	42
<i>Занесение данных в массив.....</i>	43
Лабораторная работа № 9.....	44
ИСПОЛЬЗОВАНИЕ МОДУЛЬНОГО ПРОГНАММИРОВАНИЯ.....	44
<i>Задание и порядок выполнения лабораторной работы.....</i>	44
<i>Содержание отчёта.....</i>	48
Теоретическая часть.....	49
Создание вспомогательных функций.....	49
<i>Определение функции.....</i>	50
<i>Вызов функции.....</i>	51
Лабораторная работа № 10.....	52
ИСПОЛЬЗОВАНИЕ ВСПОМОГАТЕЛЬНЫХ ФУНКЦИЙ И МАССИВОВ... ..	52
<i>Задание и порядок выполнения лабораторной работы.....</i>	52
<i>Содержание отчёта.....</i>	55
Лабораторная работа № 11.....	56
ПРОГРАММИРОВАНИЕ ЗАДАЧ С НЕЧИСЛОВЫМИ ДАННЫМИ.....	56
<i>Задание и порядок выполнения лабораторной работы.....</i>	56
<i>Содержание отчёта.....</i>	59
Теоретическая часть.....	60
Ввод-вывод символов.....	60
<i>Консольный ввод-вывод.....</i>	60
Символьные массивы.....	61
Лабораторная работа № 12.....	63
ИСПОЛЬЗОВАНИЕ ФАЙЛОВ.....	63
<i>Задание и порядок выполнения лабораторной работы.....</i>	63
<i>Содержание отчёта.....</i>	65
Теоретическая часть.....	65
Работа с файлами.....	65
<i>Понятие файла и потока ввода-вывода данных в компьютере.....</i>	65
<i>Открытие и закрытие файла.....</i>	66
<i>Потоки стандартного ввода-вывода.....</i>	68
<i>Форматированный ввод-вывод данных.....</i>	68
ОГЛАВЛЕНИЕ.....	70

