

**ГОСУДАРСТВЕННЫЙ КОМИТЕТ СВЯЗИ,
ИНФОРМАТИЗАЦИИ И ТЕЛЕКОММУНИКАЦИОННЫХ
ТЕХНОЛОГИЙ РЕСПУБЛИКИ УЗБЕКИСТАНА**

**УРГЕНЧСКИЙ ФИЛИАЛ ТАШКЕНТСКОГО
УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Факультет «Компьютер инжиниринга»
Кафедра «Информационно-образовательные технологии»

Ш.Юсупова, У.Сапаев, О.Хужаев, С.Отмуратов

ОСНОВЫ ПРОГРАММИРОВАНИЯ

Лабораторный практикум по курсу
«Основы программирования»
для студентов 1-го курса

Ургенч-2014

Лабораторный практикум по курсу «Основы программирования» для студентов 1-го курса по направлений:

1. 5330500-«Компьютер инжиниринг» («Компьютер инжиниринг», «IT-сервис», «Информационный безопасность», «Мультимедийные технологии»)
2. 5111018-«Профессиональное образование в области ИКТ»
3. 5311300-«Телекоммуникационные технологии» («Телекоммуникационные технологии», «Телерадиовещания», «Мобильные системы»)

Данная методичка по дисциплине «Основы программирования» обсуждена и утверждена на заседании кафедры «Информационно-образовательные технологии» от _____ 2014 г. протокол № _____

зав.каф. _____

Данная методичка по дисциплине «Основы программирования» обсуждена и утверждена на заседании научно-методическим совете ТУИТ УФ

Протокол № _____

« _____ » _____ 2014 г.

Составители:

ТУИТ Ургенчский филиал,
асс. Юсупова Ш.Б.

ТУИТ Ургенчский филиал,
асс. О.Хужаев

ТУИТ Ургенчский филиал,
стар.преп. У.Сапаев

ТУИТ Ургенчский филиал,
асс. С.Отамуратов

Рецензенты:

Ургенчский государственный
университет, доц., к.т.н. Г.Матлатипов

ТУИТ Ургенчский филиал,
доц., к.т.н. Ф.Юсупов

Оглавление

Предисловие

Лабораторная работа №1	6
Архитектура персонального компьютера	6
Лабораторная работа №2	13
Основные функции алгебры логики. Таблицы истинности конъюнкции, дизъюнкции, отрицания. Таблицы истинности эквивалентности и импликации.	13
Лабораторная работа №3	26
3.1. Системы счисления	26
3.2. Системы счисления. Представления символьной информации в компьютере	32
Лабораторная работа №4	41
4.1. Составление алгоритмов линейной структуры	41
4.2. Составление алгоритмов разветвляющейся структуры	45
4.3. Составление алгоритмов циклической структуры	49
Лабораторная работа №5	54
Программирование линейных алгоритмов	54
Лабораторная работа №6	60
Программирование разветвляющихся алгоритмов	60
Лабораторная работа №7	63
Программирование циклических алгоритмов	63
Лабораторная работа №8	66
Программирование с использованием одномерных массивов	66
Список литературы	68

Предисловие

В основу лабораторных практикумов положен семестровый курс «Основы программирования», читаемый авторами в течение ряда лет в Ташкенском университете информационных технологии Ургенчского филиала. В сборнике основные понятия и факты «Основы программирования» первоначально вводятся основные понятия и термины. Далее приведены краткие сведения, примеры с решениями и задачи для самостоятельного решения.

В лабораторных практикумах содержатся следующие разделы: «Архитектура ПК», «Основные функции алгебры логики, Таблицы истинности конъюнкции, дизъюнкции, отрицания. Таблицы истинности эквивалентности и импликации», «Арифметические основы компьютера. Системы счисления (2,8,16), действия в них. Выполнение заданий лабораторной работы», «Алгоритмы различной структуры», «Алгоритмы, требования, способы представления. Составление алгоритмов линейной структуры. Составление алгоритмов разветвляющейся структуры. Составление алгоритмов циклической структуры», «Программы линейной структуры», «Правила работы в Интегрированной среде C++», «Программы линейной структуры», «Функции ввода/ вывода(`printf()`/`scanf()`). Управляющая строка, спецификаторы формата. Ввод/вывод потоком», «Алгоритмы и программы разветвляющейся структуры», «Алгоритмы и программы циклической структуры», «Организация массивов», «Обработка элементов одномерного массива. Выполнение, корректировка и отладка программ»

Настоящий сборник рассчитан для студентов специальностей высших экономических и технических учебных заведений, а также для всех заинтересованных в освоении «Основы программирования» в решении практических и лабораторных задач.

Авторы

ЛАБОРАТОРНАЯ РАБОТА №1
ТЕМА: Архитектура персонального компьютера

<u>Цель и задачи работы:</u>	изучение основных блоков персонального компьютера и оценка его стоимости
<u>Срок выполнения:</u>	2 недели
<u>Характер:</u>	индивидуальный, опрос
<u>Инструменты:</u>	персональный компьютер, отвертка

Этапы выполнения

1. Изучение комплектации рабочей станции
2. Проведение экономического исследования
3. Защита работы

Этап 1. Изучение комплектации рабочей станции

1. Зарисовать вид рабочей станции с 2-3 периферийными устройствами (рис.1)
2. Обозначить и подписать на рис.1 все узлы персонального компьютера
3. Раскрыть корпус системного блока
4. Зарисовать внутренние компоненты системного блока (рис.2)
5. Обозначить и подписать на рис.2 все узлы системного блока.

Этап 2. Проведение экономического исследования

1. Заполнить таблицу 1 для исследуемой на этапе 1 конфигурации ПК
2. Вариант 1. Используя прайс-листы магазинов по продаже компьютерной техники и/или журналы «Мир техники», «Хабаровский компьютерный рынок» предложите альтернативную конфигурацию персонального компьютера за ту же стоимость. Но с использованием компонентов других производителей. Письменно обоснуйте ваши предложения.
3. Вариант 2. Предложите минимальную конфигурацию для рабочего места секретаря. Письменно обоснуйте ваши предложения.
4. Вариант 3. Предложите минимальную конфигурацию для рабочего места в игровом зале. Письменно обоснуйте ваши предложения.

Этап 1. Изучение комплектации рабочей станции

ПК и их значение

Архитектура компьютера обычно определяется совокупностью ее свойств, существенных для пользователя. Основное внимание структуре и функциональных возможностях машины. Которое можно разделить на основные и дополнительные.

Основные функции определяют назначение ЭВМ: обработка и хранение информации, обмен информацией с внешними объектами. *Дополнительные* функции повышают эффективность выполнения основных функций: обеспечивают эффективные режимы ее работы, диалог с пользователем, высокую надежность и др. названные функции ЭВМ реализуются с помощью ее компонентов: аппаратных и программных средств.

Структура компьютера – это некоторая модель, устанавливающая состав, порядок и принципы взаимодействия входящих в нее компонентов.

Персональный компьютер – это настольная или переносная ЭВМ, удовлетворяющая требованиям общедоступности и универсальности применения.

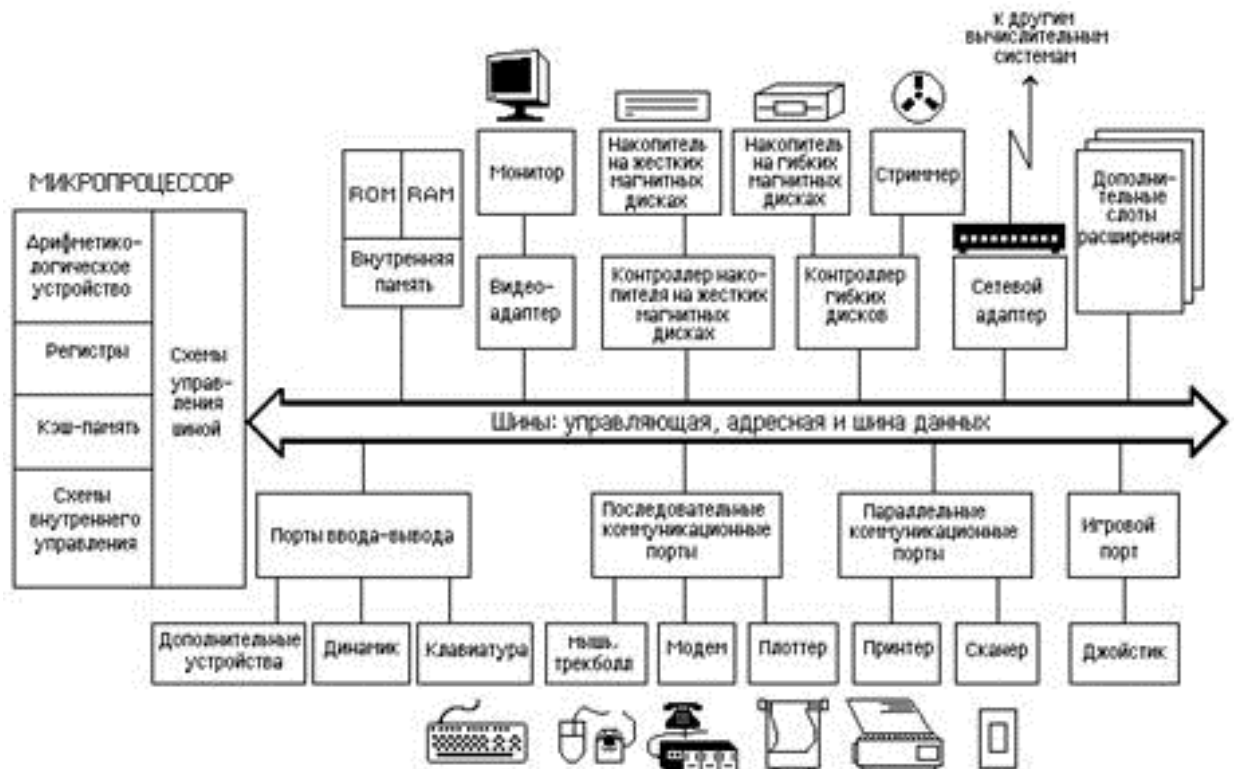
Достоинствами ПК являются:

- малая стоимость, находящаяся в пределах доступности для индивидуального покупателя;
- автономность эксплуатации без специальных требований к условиям окружающей среды;
- гибкость архитектуры, обеспечивающая ее адаптивность к разнообразным применениям в сфере управления, науки, образования, в быту;
- «дружественность» операционной системы и прочего программного обеспечения, обуславливающая возможность работы с ней пользователя без специальной профессиональной подготовки;
- высокая надежность работы (более 5 тыс. ч наработки на отказ).

Структура персонального компьютера

Рассмотрим состав и назначение основных блоков ПК.

Рис. 1. Структурная схема ПК



Обычно персональный компьютер состоит из трех частей:

- системного блока;
- клавиатуры, позволяющей вводить символы в компьютер;
- монитора (или дисплея) – для изображения текстовой или графической информации.

Компьютеры выпускаются и в портативном варианте (как дипломат) или блокнотном (ноутбук) исполнении. Здесь системный блок, монитор и клавиатура заключены в один корпус: системный блок "спрятан" под клавиатурой, а монитор сделан как крышка к клавиатуре.

Хотя из этих частей компьютера системный блок выглядит наименее эффектно, именно он является в компьютере "главным". В нем располагаются все основные узлы компьютера:

- электронные схемы, управляющие работой компьютера (микропроцессоры, оперативная память, контроллеры устройств);
- блок питания, преобразующий электропитание сети в постоянный ток низкого напряжения, подаваемый на электронные схемы компьютера;
- накопители (или дисководы) для гибких магнитных дисков, используемые для чтения и записи на гибкие магнитные диски (дискеты);
- накопитель на жестких магнитных дисках, предназначенные для чтения и записи на несъемные жесткие магнитные диски (винчестер).

Микропроцессор – важнейший элемент компьютера, так как им определяется скорость выполнения машиной программ пользователя. Со времени появления персональных компьютеров (ПК) сменилось несколько поколений процессоров, что составляет следующий ряд в порядке увеличения скорости: 8088, 486, Pentium 1-4, AMD Athlon XP, Pentium D, E, Athlon X2 и другие.

Параметры процессора:

- разрядность – ширина "такта", по которому передается компьютерная информация: 8, 16, 32 или 64 бита;
 - тактовая частота, характеризующая число команд, выполняемых процессором за одну секунду, измеряется в мегагерцах (МГц)

Микропроцессор включает в себя:

- арифметико-логическое устройство (АЛУ), которое выполняет операции (микрооперации), необходимые для выполнения команд микропроцессора;
- устройство управления (УУ) – управляет всеми частями компьютера посредством принципов программного управления;
- микропроцессорная память (МПП). В микропроцессоре есть несколько ячеек собственной памяти, они называются регистрами. Некоторые из них предназначены для хранения операндов – величин, участвующих в текущей операции. Такие регистры называются регистрами общего назначения (RON).

Регистр команд (РК) предназначен для хранения текущей команды. В регистре – счетчике команд (СК) хранится текущий адрес. Перед выполнением программы необходимо задать ее начальный адрес – записать его в счетчик команд.

Через интерфейсную систему, основу которой составляет **системная шина персонального** компьютера, микропроцессор соединяется с:

а) *Основной памятью компьютера:*

- оперативное запоминающее устройство (RAM) хранит работающую программу и данные;
- постоянное запоминающее устройство (ROM) - хранит информацию, которая необходима для постоянной работы.

RAM и ROM разбиты на ячейки, каждой из которых присвоен порядковый номер (адрес).

б) *Внешней памятью:*

- накопители на жестких магнитных дисках – устройства с несъемными носителями (винчестеры). Жесткие диски служат для постоянного хранения в компьютере программ и данных.

Выполнение многих современных прикладных программ без них невозможно. Большинство жестких дисков, в отличие от имеющих меньшую емкость дискет, нельзя снять, поэтому их иногда называют несъемными дисками. Можно, однако приобрести и съемные жесткие диски. Они ценны, когда нужно сохранять конфиденциальность информации и как средство переноса больших объемов данных между компьютерами. Емкость современных накопителей на жестких магнитных дисках может достигать нескольких Гбайт.

Более популярны накопители на оптических дисках благодаря большой емкости и надежности.

Неперезаписываемые лазерно-оптические диски обычно называют компакт-дисками ПЗУ – Compact disc (CD/DVD) ROM. Они имеют емкость от 700 Мбайт до 9 Гбайт, время доступа – от 30 до 300 мс.

- накопители на гибких магнитных дисках. Для данных накопителей носителями являются дискеты (флоппи-диски)-морально устарели
- в) *С монитором через видеоадаптер.*
г) *С принтером через адаптер принтера.*
д) *С источником питания.*
ж) *С каналом связи через сетевой адаптер.*

Сетевой адаптер дает возможность подключить компьютер в локальную сеть. При этом пользователь может получать доступ к данным, находящимся на других компьютерах.

з) *таймером* (таймер – внутренние электронные часы, которые подключены к автономному источнику питания (аккумулятору), продолжающий работать даже после отключения машины от питающей сети).

и) Микропроцессор через интерфейс связан с *клавиатурой*, а также имеет *генератор тактовых импульсов*, который генерирует последовательность электрических импульсов, а частота этих импульсов определяет тактовую частоту машины (ее быстродействие).

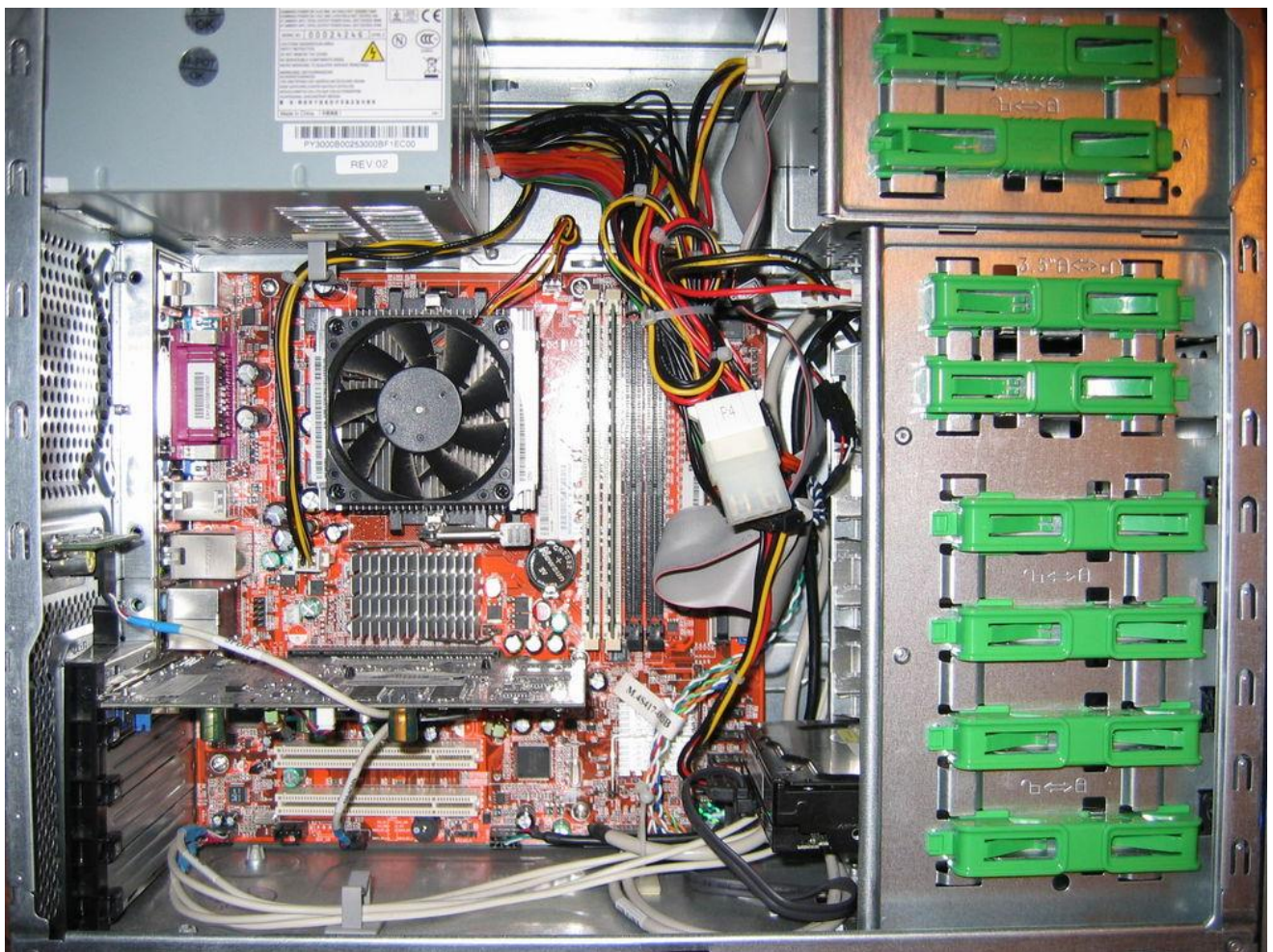
к) С *математическим сопроцессором*.

Обычно универсальные микропроцессоры относительно медленно выполняют арифметические операции над числами с плавающей запятой. Это объясняется тем, что они работают с целыми числами, и при использовании чисел, представленных в другой форме, им требуются дополнительные команды преобразования.

Для ускорения выполнения арифметических операций часто используется отдельный процессор, называемый математическим сопроцессором. Он выполняет арифметические операции над числами с плавающей запятой самостоятельно, без дополнительных программных средств. Благодаря этому в несколько раз возрастает скорость вычислительного процесса.

Узлы системного блока

Рис.2



1. Процессор и вентилятор;
2. Материнская плата;
3. Южный мост;
4. Северный мост;
5. Видеокарта;
6. Встроенный звуковой кодек;
7. жесткий диск;

8. оперативная память;
9. блок питания;
10. оптический привод;
11. Флоппи дисковод;
12. Корпус;

Этап 2. Проведение экономического исследования

Рассмотренные конфигурации ПК по заданию приведены в таблице 1, цены указаны действующие.

Таблица 1.

Наименование	Ваша модель		Вариант 1		Вариант 2	
	Модель	Цена	Модель	Цена	Модель	Цена
Монитор	Samsung SyncMaster 765MB, 17''	3000	LG LCD Flatron L1770 HQ, 17''	4905	LG LCD Flatron L1770 HQ, 17''	4905
Корпус	ATX Microlab Black	500	ATX Microlab Black	500	ATX Microlab Black	500
Блок питания	ATX 400W NAVI	950	ATX 400W NAVI	950	ATX 400W NAVI	950
Материнская плата	Gigabyte nForce4, S939, DDR, PCI-Ex16, SATA, GLan, IEEE 1394	1200	Gigabyte Intel945P, LGA775, DDR, PCI-Ex16, SATA, GLan, IEEE 1394	2100	Microstar Intel945G, LGA775, DDR, Video, SATA, GLan	1900
Жесткий диск	Western Digital Sata II, 160 Gb, (WD1600JS), cache 8Mb	1655	Seagate Sata II, 160 Gb, (ST3160815), cache 8Mb	1700	Seagate Sata II, 160 Gb, (ST3160815), cache 8Mb	1700
Процессор	AMD Athlon 64 3200+ cache 128Kb+512Kb socket 939(box)	1200	Intel Pentium D 3000MHz, cache 2x2048Kb, LGA775 (box)	2480	Intel Celeron 3200MHz, cache 256Kb, LGA775 (box)	1800
Оперативная память	Samsung, 512Mb x2, DDR PC3200	1460	Kingston, 1Gb DDR PC5400	980	Samsung, 512Mb, DDR PC3200	730
CD/DVD – rom	DVD/CD –RW NEC-ND3550A, dual layer IDE	900	DVD/CD –RW TEAC-DV-W518 GB dual layer IDE	900	LG DVD±RW/CD –RW (GSA-H42N), dual layer IDE	899
Клавиатура	Defender Office PS/2	400	LogiTech Deluxe 250 PS/2	350	A4-Tech (KB720), white, pS/2	165
Мышь	Genius Optical NetScroll	200	Genius Optical NetScroll	200	Genius Optical NetScroll	200
Видеокарта	PCI-Ex16 ASUS Geforce 6600 256Mb, 128bit (VGA+DVI)	1100	PCI-Ex16 Sapphire Radeon X1300XT 256Mb, 128bit (VGA+DVI)	1400	{встроенная в материнской плате}	-
итого		12565		16465		13749

Комментарии к варианту 1

Альтернативная конфигурация ПК собрана на базе комплектующих INTEL. Оборудование данного производителя обеспечивает лучшую производительность всей системы, однако увеличивается суммарная стоимость системы. В ходе разработки альтернативной конфигурации системы был заменен ЭЛТ монитор на ЖК, в связи с тем, что данная технология морально устарела и данный монитор занимает много полезной площади. Жесткие диски Seagate и Western Digital обладают одинаковыми параметрами работы, однако надежность Seagate будет на пару пунктов больше. Видеокарта на чипсете ATI по своим параметрам является аналогом моей видеокарты по производительности, но в некоторых графических приложениях и играх обеспечивает прирост по производительности

Комментарии к варианту 2

Данная конфигурация компьютера собрана для работы в Офисе. ЖК монитор обладает малыми размерами, что позволяет сэкономить весьма дефицитную полезную площадь рабочего места, также при работе за ЖК монитором не устают глаза, Размер матрицы выбран 17'', так как данный размер монитора позволяет комфортно работать с текстом. Система базируется на процессоре Intel Celeron, данная урезанная версия процессора Intel Pentium D в сочетании с встроенной видеокартой обеспечивает комфортную работу в офисных приложениях и при

работе со стандартными графическими приложениями. Объем оперативной памяти хватает для одновременной работы сразу в нескольких приложениях. Показатели системы удовлетворяют рекомендуемым системным требованиям Windows Vista/Windows XP

Комментарии к варианту 3

Данная конфигурация позволяет справиться с любыми самыми «жадными» приложениями, начиная от сверх современных игр, заканчивая сложными графическими пакетами обработки 3D графики, следовательно данный компьютер будет полезен не только заядлым геймерам, но и серьезным разработчикам софта и даже самих игр. Мощный четырехядерный процессор последней разработки Intel обеспечивает просто фантастическую мощь-2.4 GHz в четырех ядрах, а наличие большого объема Кеша позволяет оптимизировать выполнение сложных задач. В системе установлено 2 гигабайта оперативной памяти, что является вполне закономерным решением, все современные приложения требуют от 1 до 2 гигабайт памяти, пропускная частота памяти в 800Mhz обеспечивает эффективную работу системы в целом. В компьютере установлена материнская плата на чипсете Nforce 680, данный вид чипсета оптимизирован для работы с несколькими процессорами. Также на материнской плате установлен контроллер Gigabit LAN , что позволит подключить компьютер к сети. Отдельно стоит сказать о технологии PCI-Ex16 SLI- Данная технология обеспечивает одновременную работу двух топовых видеокарт в режиме моста, разделение задач между двумя видеокартами обеспечивает эффективную работу графики, так что пользователь надолго забудет, о торможении кадров и выпадении текстур. Выбор жесткого диска объемом в 750Gb является оптимальным с точки зрения соотношения объем/качество/цена. Данная емкость позволяет хранить огромное количество игр, фильмов, музыки и других файлов пользователя. Блок питания фирмы Niperg на 580W обеспечивает запас по мощности, для дальнейшей модернизации. 22 дюймовый монитор с функцией HDTV и откликом 2мс позволит полностью насладиться качеством 3D графики современных игр и приложений. Привод гибких дисков Plextor может прочитать, все современные форматы данных даже на очень некачественных носителях и высокую скорость передачи обеспечивает подключение по интерфейсу SATA. Стильный алюминиевый корпус полностью дополняет всю систему.

ВЫВОД: Выполняя лабораторную работу я ознакомился с основными узлами персонального компьютера, оценил стоимость различных конфигураций систем и провел исследование различных конфигураций системных блоков. В ходе работы я пришел к выводу, что на компьютерном рынке существуют различные классы и типы комплектующих и при покупке или сборке компьютера следует руководствоваться требованиями, предъявляемым программным обеспечением и областью применения данного оборудования

Задачи:

1. Монитор и их тип
2. Об оперативной памяти
3. Тип CD и DVD - ROM
4. Интерфейс IDE
5. Клавиатура
6. Микропроцессор и их типы
7. Принтер устройства и их типов
8. Объединительная плата.
9. Мышка
10. Модемы.

СПИСОК ЛИТЕРАТУРЫ:

1. Смирнов А.Д. Архитектура вычислительных систем. М.: Наука, 1990. 320 с.
2. Вычислительные машины системы и сети. /Под ред. А.П. Пятибратова. М.: Финансы и статистика, 1991. 400 с.
3. Ахметов К.С., Борзенко А.Е. Современный персональный компьютер. М.: ТОО КомпьютерПресс, 1995. 317 с

Понятие высказывания

Высказывание – это любое утверждение, относительно которого можно сказать, истинно оно или ложно. Приведем примеры высказываний:

- Число 14 делится на 2 и 7.
- Париж – столица Испании.
- Хабаровск стоит на Амуре.
- $3 > 1$.

Высказывания «Число 14 делится на 2 и 7» и «Хабаровск стоит на Амуре» истинны, а высказывания «Париж – столица Испании» и « $3 > 1$ » ложны.

По своей сути высказывания фактически являются двоичными объектами, и потому истинному значению высказывания ставят в соответствие 1 (Истина, True), а ложному – 0 (Ложь, False). В дальнейшем будем обозначать высказывания буквами латинского алфавита.

Например, запись $A = 1$ означает, что высказывание A истинно.

Высказывания могут быть простыми и сложными. Простые высказывания соответствуют логическим переменным, которые принимают значение 0 или 1. Сложные высказывания являются аналогом логических функций и могут образовываться путем объединения переменных с помощью логических операций. Любая логическая функция может быть задана с помощью таблицы истинности, в левой части которой записываются возможные наборы переменных (аргументов), а в правой – соответствующие им значения функции. Это возможно по той причине, что все сочетания логических аргументов легко перечислить.

Из вышеприведенных высказываний второе и третье являются простыми, а первое высказывание, образованное из простых высказываний «Число 14 делится на 2» и «Число 14 делится на 7», является сложным высказыванием.

Логические операции над логическими переменными

Отрицание

Операция отрицания является унарной, т.к. имеет один аргумент. Иначе ее называют *инверсией*, *дополнением*, *НЕ* и обозначают \bar{X} или $\neg X$, NOT X .

Отрицанием \bar{A} некоторого высказывания A называется такое высказывание, которое истинно, когда A ложно, и ложно, когда A истинно.

Определение отрицания может быть записано с помощью таблицы истинности:

X	\bar{X}
0	1
1	0

Логическое умножение

Операцию логического умножения еще называют *конъюнкцией*, *логическим И*. Для обозначения данной операции используют символы \wedge , $\&$, точку, которую можно опускать, AND.

Конъюнкцией двух высказываний X и Y называется такое высказывание, которое истинно тогда и только тогда, когда истинны оба высказывания X и Y .

Определение конъюнкции может быть записано в виде таблицы истинности:

X	Y	X & Y
0	0	0
0	1	0
1	0	0
1	1	1

Определение конъюнкции двух высказываний естественным образом распространяется на любое конечное число составляющих: конъюнкция $X_1 \& X_2 \& X_3 \& \dots \& X_n$ истинна тогда и только тогда, когда истинны все высказывания $X_1, X_2, X_3, \dots, X_n$, следовательно, принимает значение «ложь», когда ложно хотя бы одно из этих высказываний.

Логическое сложение

Операцию логического сложения иначе называют *дизъюнкцией*, *логическим ИЛИ*. Для обозначения логического сложения используют символы \vee , $+$, OR.

Таким образом, дизъюнкцией двух высказываний называется такое новое высказывание, которое истинно тогда и только тогда, когда истинно хотя бы одно из этих высказываний.

Определение дизъюнкции может быть записано в виде таблицы истинности:

X	Y	$X \vee Y$
0	0	0
0	1	1
1	0	1
1	1	1

Определение дизъюнкции двух высказываний естественным образом распространяется на любое конечное число составляющих: дизъюнкция $X_1 \vee X_2 \vee X_3 \vee \dots \vee X_n$ истинна тогда и только тогда, когда истинно хотя бы одно из этих высказываний, а, следовательно, принимает значение «ложь», когда все высказывания ложны.

Логическое следование

Операцию логического следования иначе называют *импликацией* и для обозначения используют символ \rightarrow .

Импликацией $X \rightarrow Y$ называется высказывание, которое ложно тогда и только тогда, когда X истинно и Y ложно.

Определение импликации может быть записано в виде таблицы истинности:

X	Y	$X \rightarrow Y$
0	0	1
0	1	1
1	0	0
1	1	1

Логическое тождество

Операцию логического тождества иначе называют *эквиваленцией*, *эквивалентностью* и для обозначения используют символы $=$, \leftrightarrow , \sim .

Таким образом, эквиваленцией двух высказываний X и Y называется такое высказывание, которое истинно тогда и только тогда, когда оба эти высказывания A и B истинны или оба ложны.

Определение эквиваленции может быть записано в виде таблицы истинности:

X	Y	$X \leftrightarrow Y$
0	0	1
0	1	0
1	0	0
1	1	1

Операция исключающее ИЛИ

Операция исключающее ИЛИ (*неравнозначность*, *сложение по модулю два*) обозначается символом \oplus и отличается от логического ИЛИ только при $X=1$ и $Y=1$. Таким образом, неравнозначностью двух высказываний X и Y называют такое высказывание, которое истинно тогда и только тогда, когда одно из этих высказываний истинно, а другое ложно.

Определение данной операции может быть записано в виде таблицы истинности:

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

Операция исключающее ИЛИ фактически сравнивает на совпадение два двоичных разряда.

Логические операции инверсии, дизъюнкции, конъюнкции образуют полную систему логических операций, из которых можно построить сколь угодно сложное логическое выражение. При вычислении значения логического выражения принято следующее старшинство (приоритет) логических операций: сначала выполняется инверсия, затем конъюнкция и в последнюю очередь – дизъюнкция. Для изменения указанного порядка используют скобки.

Законы логики

Законы логики записываются в виде формул, которые позволяют производить эквивалентные преобразования логических выражений.

Закон тождества

Всякое высказывание тождественно самому себе:

$$X = X.$$

Закон идемпотентности

Закон означает отсутствие показателей степени:

$$X \vee X = X,$$

$$X \& X = X.$$

Закон исключения констант

$$\text{Для логического сложения } X \vee 1 = 1, \quad X \vee 0 = X.$$

$$\text{Для логического умножения } X \& 1 = X, \quad X \& 0 = 0.$$

Закон поглощения

$$\text{Для логического сложения } X \vee (X \& Y) = X.$$

$$\text{Для логического умножения } X \& (X \vee Y) = X.$$

Закон противоречия

Высказывание не может быть одновременно истинным и ложным. Если высказывание X истинно, то его отрицание \bar{X} должно быть ложным. Следовательно, логическое произведение высказывания и его отрицания должно быть ложно:

$$X \& \bar{X} = 0.$$

Закон исключенного третьего

Высказывание может быть либо истинным, либо ложным, третьего не дано. Это означает, что результат логического сложения высказывания и его отрицания всегда принимает значение «истина»:

$$X \vee \bar{X} = 1.$$

Закон двойного отрицания

Если дважды отрицать некоторое высказывание, то в результате мы получим исходное высказывание:

$$\bar{\bar{X}} = X.$$

Законы общей инверсии (законы де Моргана)

$$\overline{X \vee Y} = \bar{X} \& \bar{Y},$$

$$\overline{X \& Y} = \bar{X} \vee \bar{Y}.$$

Важное значение для выполнения преобразований логических выражений имеют законы алгебраических преобразований. Многие из них имеют аналоги в обычной алгебре.

Закон переместительный (коммутативности)

В обычной алгебре слагаемые и множители можно менять местами. В алгебре высказываний можно менять местами логические переменные при операциях логического умножения и логического сложения:

$$X \& Y = Y \& X,$$

$$X \vee Y = Y \vee X.$$

Закон сочетательный (ассоциативности)

Если в логическом выражении используются только операция логического умножения или только операция логического сложения, то можно пренебрегать скобками или произвольно их расставлять:

$$(X \& Y) \& Z = X \& (Y \& Z),$$

$$(X \vee Y) \vee Z = X \vee (Y \vee Z).$$

Закон распределительный (дистрибутивности)

В отличие от обычной алгебры, где за скобки можно выносить только общие множители, в алгебре высказываний можно выносить за скобки как общие множители, так и общие слагаемые.

Дистрибутивность умножения относительно сложения:

$$(X \& Y) \vee (X \& Z) = X \& (Y \vee Z).$$

Дистрибутивность сложения относительно умножения:

$$(X \vee Y) \& (X \vee Z) = X \vee (Y \& Z)$$

Логические элементы

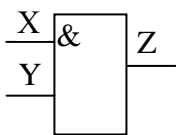
При всей сложности устройства электронных блоков современных компьютеров выполняемые ими действия осуществляются комбинацией относительно небольшого числа типовых логических узлов. Перечислим основные из них:

- регистры;
- комбинационные преобразователи кодов (шифратор, дешифратор, мультиплексор и др.);
- счетчики (кольцевой, синхронный, асинхронный);
- арифметико-логические узлы (сумматор, узел сравнения).

Из этих узлов строятся интегральные микросхемы очень высокого уровня интеграции: микропроцессоры, модули ОЗУ, контроллеры внешних устройств и т.д.

Сами указанные узлы собираются из основных базовых элементов – как простейших, так и более сложных. Условные обозначения основных логических элементов приведены в табл. 1.

Таблица 1

Логическая функция	Условные обозначения логических элементов
Инверсия (НЕ) $Z = \bar{X}$	
Конъюнкция (И) $Z = X \& Y$	

Окончание табл. 1

Логическая функция	Условные обозначения логических элементов
Дизъюнкция (ИЛИ) $Z = X \vee Y$	
Исключающее ИЛИ $Z = X \oplus Y$	
Инверсия конъюнкции (И – НЕ) $Z = \overline{X \& Y}$	
Инверсия дизъюнкции (ИЛИ – НЕ) $Z = \overline{X \vee Y}$	

Решение задач

Пример 1

Какой логической операции соответствует таблица истинности, приведенная ниже?

X	F
0	1
1	0

$$F = \overline{X}.$$

Таблица истинности соответствует операции отрицания.

Пример 2

Найти значение логического выражения

$$A \vee B \& \overline{C} \text{ при } A=0 \text{ (False), } B=1 \text{ (True), } C=0 \text{ (False).}$$

Решение: подставим значения переменных в выражение и вычислим его согласно приоритету выполнения операций:

$$A \vee B \& \overline{C} = 0 \vee 1 \& \overline{0} = 0 \vee 1 \& 1 = 0 \vee 1 = 1 \text{ (True).}$$

Ответ: заданное логическое выражение принимает значение True.

Пример 3

Найти значение логического выражения

$$(a < z) \vee (z > 2) \vee (a \neq 5) \text{ при } a = 5, z = -4.$$

Решение: подставим значения переменных в выражение и вычислим его согласно приоритету выполнения операций:

$$(5 < -4) \vee (-4 > 2) \vee (5 \neq 5) = 0 \vee 0 \vee 0 = 0.$$

Ответ: заданное логическое выражение принимает значение False.

Пример 4

Упростить логическое выражение: $(A \& B) \vee (A \& \bar{B})$. Правильность упрощения проверить с помощью таблиц истинности.

Решение: воспользуемся законом дистрибутивности и вынесем за скобки A:

$$(A \& B) \vee (A \& \bar{B}) = A \& (B \vee \bar{B}).$$

По закону исключенного третьего $B \vee \bar{B} = 1$, следовательно,

$$A \& (B \vee \bar{B}) = A \& 1 = A.$$

Таким образом, в результате упрощения получили $(A \& B) \vee (A \& \bar{B}) = A$.

Составим таблицу истинности (табл. 2) для выражения $(A \& B) \vee (A \& \bar{B})$.

Таблица 2

A	B	\bar{B}	$A \& B$	$A \& \bar{B}$	$(A \& B) \vee (A \& \bar{B})$
0	0	1	0	0	0
0	1	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1

Из таблицы истинности видно, что справедливо $(A \& B) \vee (A \& \bar{B}) = A$.

Ответ: $(A \& B) \vee (A \& \bar{B}) = A$.

Пример 5

По заданной логической схеме (рис. 1) составить логическое выражение и заполнить для него таблицу истинности.

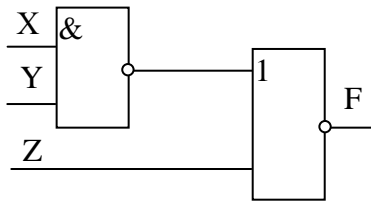


Рис. 1

Решение: используя обозначения логических элементов (табл. 1), составим логическое выражение

$$F = \overline{\overline{X \& Y} \vee Z}.$$

Заполним для F таблицу истинности (табл. 3).

Таблица 3

X	Y	Z	X & Y	$\overline{X \& Y}$	$\overline{X \& Y} \vee Z$	$\overline{\overline{X \& Y} \vee Z}$
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	0	1	1	0
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	0	1	0

Пример 6

Задано логическое выражение $F = \overline{\overline{X \vee Y} \& \overline{Z}}$. Составить логическую схему для данного выражения и заполнить таблицу истинности.

Решение: используя обозначения логических элементов (табл. 1), составим логическую схему (рис. 2).

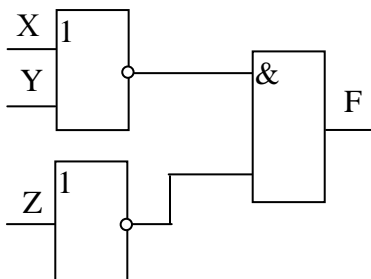


Рис. 2

Заполним для F таблицу истинности (табл. 4).

Таблица 4

X	Y	Z	$X \vee Y$	$\overline{X \vee Y}$	\overline{Z}	$\overline{X \vee Y} \& \overline{Z}$
0	0	0	0	1	1	1
0	0	1	0	1	0	0
0	1	0	1	0	1	0
0	1	1	1	0	0	0
1	0	0	1	0	1	0
1	0	1	1	0	0	0
1	1	0	1	0	1	0
1	1	1	1	0	0	0

Задания для самостоятельного решения

Задание 1

Какой логической операции соответствует таблица истинности (табл. 5)?

Таблица 5

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	1

Задание 2

Какой логической операции соответствует таблица истинности (табл. 6)?

Таблица 6

X	Y	F
0	0	1
0	1	1
1	0	0
1	1	1

Задание 3

Найти значения приведенных ниже логических выражений:

1. $A \& B \vee \bar{C}$ при $A = \text{False}$, $B = \text{True}$, $C = \text{False}$.
2. $(x = y) \vee (z < 4)$ при $x = 5$, $y = 7$, $z = 0$.
3. $A \vee \bar{B}$ при $A = \text{False}$, $B = \text{False}$.
4. $(a < z) \vee (z > -10) \& (a \neq 5)$ при $a = 8$, $z = -6$.
5. $\overline{A \vee B} \& C$ при $A = \text{False}$, $B = \text{False}$, $C = \text{True}$.

Задание 4

Упростить логическое выражение $\overline{A \vee B} \vee \overline{\overline{A \vee B}}$. Правильность упрощения проверить с помощью таблицы истинности.

Задание 5

Доказать данное равенство с помощью таблицы истинности.

$$X \leftrightarrow Y = (\bar{X} \& \bar{Y}) \vee (X \& Y).$$

Задание 6

По заданной логической схеме (рис. 3) составить логическое выражение и заполнить для него таблицу истинности.

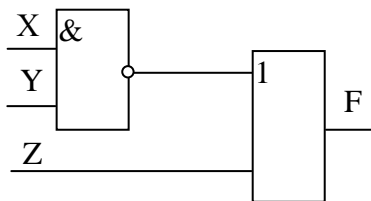


Рис. 3

Задание 7

По заданной логической схеме (рис. 4) составить логическое выражение и заполнить для него таблицу истинности.

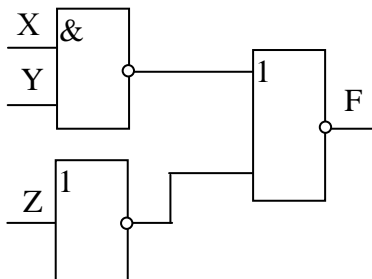


Рис. 4

Задание 8

По заданной логической схеме (рис. 5) составить логическое выражение и заполнить для него таблицу истинности.

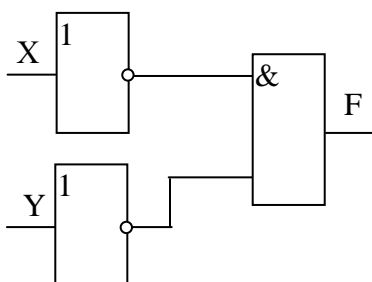


Рис. 5

Задание 9

По заданным логическим выражениям составить логические схемы и заполнить таблицы истинности.

$$\overline{\overline{X \& \overline{Y} \vee Z}} = F;$$

$$X \vee \overline{\overline{Y} \& Z} = F;$$

$$\overline{\overline{X} \vee Y \& \overline{Z}} = F.$$

Библиографический список

1. Могилев А. В. Информатика : учеб. пособие для студ. высш. учеб. заведений / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. – М. : Издательский центр «Академия», 2008. – 848 с.
2. Могилев А. В. Практикум по информатике : учеб. пособие для студ. высш. учеб. заведений / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. – М. : Издательский центр «Академия», 2008. – 608 с.
3. Алексеев П. А. Информатика / П. А. Алексеев. – М. : СОЛОН-Р, 2002 – 368 с.
4. Акулов О. А. Информатика : учеб. пособие для студ. вузов / О.А. Акулов, Н. В. Медведев. – М. : Омега-Л, 2005. – 552 с.
5. Лихтарников Л. М. Математическая логика. Курс лекций. Задачник-практикум и решения : учеб. пособие / Л. М. Лихтарников, Т. Г. Сукачева. – СПб. : Лань, 2008. – 288 с.

Лабораторная работа №3.1

Системы счисления

Цель работы: Ознакомления и изучения системы счисления. Выполнение на них арифметических действий.

Теоретическая часть:

При работе с компьютерами приходится параллельно использовать несколько позиционных систем счисления – двоичную, восьмеричную, шестнадцатеричную.

Системы счисления – принятый способ наименования и записи чисел с помощью символов, имеющих определенные количественные значения.

Все системы счисления можно разделить на два класса: *позиционные* и *непозиционные*. В *позиционной системе* счисления количественное значение каждой цифры зависит от ее места (позиции) в числе. В *непозиционных системах* счисления значение цифры не зависит от места, которое он занимает в числе. Самый известный пример – римская система счисления. В этой системе счисления используется 7 знаков

I (1) V (5) X (10) L (50) C (100) D (500) M (1000)

Например, III (три) LIX (59) DLV (555)

Для записи чисел в различных системах счисления используется некоторое количество отличных друг от друга знаков. Число таких знаков в позиционной системе счисления называется *основанием системы счисления*.

Некоторые системы счисления

Основание	Системы счисления	Знаки
2	Двоичная	0, 1
5	Пятеричная	0, 1, 2, 3, 4
8	Восьмеричная	0, 1, 2, 3, 4, 5, 6, 7
10	Десятичная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16	Шестнадцатеричная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Соответствие чисел в различных системах счисления

Десятичная	Шестнадцатеричная	Восьмеричная	Двоичная
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	8	10	1000
9	9	11	1001
10	A	12	1010
11	B	13	1011
12	C	14	1100
13	D	15	1101
14	E	16	1110
15	F	17	1111

Перевод чисел из одной системы счисления в другую.

Перевод целых чисел из десятичной системы счисления в другую.

Правило перевода целых чисел из десятичной системы счисления в систему с основанием q :

1. Последовательно выполнять деление исходного числа и получаемых частных на q до тех пор, пока не получим частное, меньшее делителя.
2. Полученные при таком делении остатки – цифры числа в системе счисления q – записать в обратном порядке (снизу вверх).

Пример1. Перевести 26_{10} в двоичную систему счисления. $A_{10} \rightarrow A_2$

Решение:

$$\begin{array}{r|l} 26 & 2 \\ \hline 13 & 2 \\ \hline 12 & 6 \\ \hline 6 & 2 \\ \hline 3 & 2 \\ \hline 1 & 1 \end{array}$$

Ответ: $26_{10} = 11010_2$

Пример2. Перевести 241_{10} в восьмеричную систему счисления. $A_{10} \rightarrow A_8$

Решение:

$$\begin{array}{r|l} 241 & 8 \\ \hline 30 & 8 \\ \hline 24 & 3 \\ \hline 6 & \end{array}$$

Ответ: $241_{10} = 361_8$

Пример3. Перевести 3627_{10} в шестнадцатеричную систему счисления. $A_{10} \rightarrow A_{16}$

Решение:

$$\begin{array}{r|l} 3627 & 16 \\ \hline 226 & 16 \\ \hline 224 & 14 \\ \hline 2 & \end{array}$$

Т.к. в шестнадцатеричной системе счисления 14 – E, а 11 – B, то получаем ответ E2B₁₆.

Ответ: $3627_{10} = E2B_{16}$

Перевод дробных чисел из десятичной системы счисления в другую.

Правило перевода дробных чисел из десятичной системы счисления в систему с основанием q :

1. Последовательно выполнять умножение исходного числа и получаемых дробные части на q до тех пор, пока дробная часть не станет равна нулю или не достигнем требуемую точность.
2. Полученные при таком умножении целые части - числа в системе счисления q – записать в прямом порядке (сверху вниз).

Пример1. Перевести $0,5625_{10}$ в двоичную систему счисления. $A_{10} \rightarrow A_2$

Решение:

$$\begin{array}{l} 0,5625 \\ \downarrow \times 2 \\ 1,1250 \\ \downarrow \times 2 \\ 0,2500 \\ \downarrow \times 2 \\ 0,5000 \\ \downarrow \times 2 \\ 1,0000 \end{array}$$

Ответ: $0,5625_{10} = 0,1001_2$

Пример2. Перевести $0,5625_{10}$ восьмеричную систему счисления. $A_{10} \rightarrow A_8$

Решение:

$$\begin{array}{l} 0,5625 \\ \downarrow \times 8 \\ 5,2500 \\ \downarrow \times 8 \\ 2,0000 \end{array}$$

Ответ: $0,5625_{10} = 0,52_8$

Перевод чисел из любой системы счисления в десятичную.

Правило Для того чтобы число из любой системы счисления перевести в десятичную систему счисления, необходимо его представить в развернутом виде и произвести вычисления.

Пример1. Перевести число $101,01_2$ из двоичной системы счисления в десятичную.

Решение:

$$\begin{array}{r} 2 \quad 1 \quad 0 \quad -1 \quad -2 \\ 1 \quad 0 \quad 1,0 \quad 1 \quad 2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 4 + 0 + 1 + 0 + 0,25 = 5,25_{10} \end{array}$$

Ответ: $101,01_2 = 5,25_{10}$

Пример2. Перевести число $2E_{16}$ в десятичную систему счисления.

Решение:

$$\begin{array}{r} 2 \quad 1 \\ 2 \quad E_{16} = 2 \cdot 16^1 + 14 \cdot 16^0 = 32 + 14 = 46_{10}. \end{array}$$

Ответ: $2E_{16} = 46_{10}$.

Арифметические операции в системах счисления.

Используя таблицу и привычные правила сложения, совсем не трудно складывать и вычитать числа в восьмеричной системе счисления

Правило

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Пример1. Вычислите $634_8 + 275_8$

Решение:

$$\begin{array}{r} 634 \\ + 275 \\ \hline 1131 \end{array} \quad \text{Ответ: } 1131_8$$

Пример2. Вычислите $305,4_8 + 24,75_8$

Решение:

$$\begin{array}{r} 305,4 \\ + 24,75 \\ \hline 332,35 \end{array} \quad \text{Ответ: } 332,35_8$$

Пример3. Вычислите $634_8 - 275_8$

Решение:

$$\begin{array}{r} 634 \\ - 275 \\ \hline 337 \end{array}$$

Ответ: 337_8

Пояснение: Т.к. от 4 не отнять 5, то занимаем у следующего разряда (т.к. система восьмеричная то 1 разряд составляет 8 единиц). От 8 - 5 + 4 = 7

Аналогично, т.к. у тройки одну единицу заняли, то необходимо от 2 отнять 7, поэтому, заняв у следующего разряда, получаем $8 - 7 + 2 = 3$ и т.д.

Пример4. Вычислите $305,4_8 - 24,75_8$

Решение:

$$\begin{array}{r} 305,40 \\ - 24,75 \\ \hline 260,43 \end{array} \quad \text{Ответ: } 260,43_8$$

Используя правило умножения и сложения восьмеричных чисел не трудно и перемножать данные числа

Правило

*	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	38	44	52
7	0	7	16	25	34	43	52	61

Пример. Вычислите $63_8 \cdot 27,5_8$

Решение:

$$\begin{array}{r} 27,5 \\ \times 63 \\ \hline 1067 \\ 2156 \\ \hline 2264,7 \end{array}$$

Ответ: $2264,7_8$

Сложение и вычитание осуществляется аналогично таким же действиям в восьмеричной системе счисления

Правило

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Пример.

Вычислите $E5F6_{16} + A07_{16}$

$$\begin{array}{r} E5F6 \\ + A07 \\ \hline E5FD \end{array}$$

Ответ:

$E5FD_{16}$

Правило

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	7E	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	48	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	45	B4	C3	D2	E1

Пример

FFA,3

* D,E

DFAEA

CFB47

DDAF,5A

Задания:

Простые задания:

Даны:

- 121521213231 число, перевести 2,8 и 16 ичную систему счисления.
- 784654656431 число, перевести 2,8 и 16 ичную систему счисления.
- 121655845123 число, перевести 2,8 и 16 ичную систему счисления.
- 145455213231 число, перевести 2,8 и 16 ичную систему счисления.
- 984654654651 число, перевести 2,8 и 16 ичную систему счисления.
- 131231542231 число, перевести 2,8 и 16 ичную систему счисления.
- 121521213231 число, перевести 2,8 и 16 ичную систему счисления.
- 874548465655 число, перевести 2,8 и 16 ичную систему счисления.
- 888745475555 число, перевести 2,8 и 16 ичную систему счисления.
- 487451253213 число, перевести 2,8 и 16 ичную систему счисления.
- 314572213213 число, перевести 2,8 и 16 ичную систему счисления.
- 212187444123 число, перевести 2,8 и 16 ичную систему счисления.
- 154875213231 число, перевести 2,8 и 16 ичную систему счисления.

Задачи повышенной сложности:

Из данной таблицы найти значения X, Y и Z. Выполнить действия в данных системах счисления.

Вариант	A	B	2	8	16
1	5654653 ₁₀	45678 ₁₀	A+B=(X) ₂	A/B=(Y) ₈	A*B=(Z) ₁₆
2	6554653 ₁₀	21368 ₁₀	A*B=(X) ₂	A-B=(Y) ₈	A+B=(Z) ₁₆
3	5123353 ₁₀	12378 ₁₀	A-B=(X) ₂	A*B=(Y) ₈	A+B=(Z) ₁₆
4	8545253 ₁₀	56423 ₁₀	A*B=(X) ₂	A-B=(Y) ₈	A+B=(Z) ₁₆

5	5542353_{10}	12313_{10}	$A/B=(X)_2$	$A+B=(Y)_8$	$A+B=(Z)_{16}$
6	5121543_{10}	78456_{10}	$A*B=(X)_2$	$A-B=(Y)_8$	$A+B=(Z)_{16}$
7	5121234_{10}	12313_{10}	$A*B=(X)_2$	$A-B=(Y)_8$	$A/B=(Z)_{16}$
8	5121542_{10}	12124_{10}	$A+B=(X)_2$	$A/B=(Y)_8$	$A*B=(Z)_{16}$
9	1231635_{10}	44518_{10}	$A-B=(X)_2$	$A*B=(Y)_8$	$A+B=(Z)_{16}$
10	5542465_{10}	12151_{10}	$A/B=(X)_2$	$A-B=(Y)_8$	$A*B=(Z)_{16}$
11	5645213_{10}	15412_{10}	$A/B=(X)_2$	$A-B=(Y)_8$	$A*B=(Z)_{16}$
12	5545133_{10}	45453_{10}	$A-B=(X)_2$	$A*B=(Y)_8$	$A+B=(Z)_{16}$
13	5452333_{10}	78532_{10}	$A+B=(X)_2$	$A/B=(Y)_8$	$A-B=(Z)_{16}$

Лабораторная работа №3.2

#4. Системы счисления. Представление символьной информации в компьютере

Правило 1.

Для перевода целого числа A из 10-ичной системы счисления в систему счисления с основанием d необходимо A разделить с остатком на число d , записанное в десятичной системе. Затем неполное частное, полученное от деления, нужно снова разделить с остатком на d и т.д., пока последнее полученное неполное частное не станет равным нулю. Представлением числа A в новой системе счисления будет последовательность остатков деления, изображенных d -ичной цифрой и записанных в порядке, обратном порядку их получения.

Пример 1. Переведем число $A=68_{10}$ в двоичную систему ($d = 2$).

$$\begin{array}{l} 68:2=34 (0) \\ 34:2=17 (0) \\ 17:2=8 (1) \\ 8:2=4 (0) \\ 4:2=2 (0) \\ 2:2=1 (0) \\ 1:2=0 (1) \end{array} \uparrow$$

Т.к. ноль и единица в обеих системах счисления обозначаются одинаковыми цифрами 0 и 1, получаем $68_{10}=1000100_2$.

Пример 2. Переведем число $A=1308_{10}$ в шестнадцатеричную систему ($d = 16$).

$$\begin{array}{l} 1308:16=81 (12) \\ 81:16=5 (1) \\ 5:16=0 (5) \end{array} \uparrow$$

Остаток 12 в шестнадцатеричной системе счисления обозначается цифрой С, поэтому окончательно получаем $1308_{10}=51C_{16}$.

Задание 1. Переведите данное число из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.

№ варианта	10-ная	2-ная	8-ная	16-ная
1.	860 78			
2.	250 57			
3.	759 26			
4.	216 33			
5.	530 25			
6.	945 85			
7.	287 20			

8.	485 90			
9.	639 48			
10.	618 55			
11.	772 71			
12.	233 43			
13.	218 77			
14.	898 71			
15.	557 30			
16.	737 92			
17.	575 74			
18.	563 30			
19.	453 41			
20.	572 36			

Проверьте правильность перевода чисел в разные системы счисления с помощью приложения Калькулятор. Переключатель систем счисления в Калькуляторе: Нех – шестнадцатеричная система, Дес – десятичная, Oct – восьмеричная, Bin – двоичная. Например, необходимо перевести число 10010_2 из двоичной системы в шестнадцатеричную. Для этого необходимо переключить Калькулятор в двоичную систему \odot Bin, набрать это число, а затем переключить калькулятор в шестнадцатеричную систему \odot Нех.

Правило 2.

Для перевода в десятичную систему числа A , записанного в d -ичной системе счисления в виде $A_d = (a_n a_{n-1} \dots a_1 a_0)$ необходимо вычислить значение многочлена

$$A_{10} = a_n \times d^n + a_{n-1} \times d^{n-1} + \dots + a_1 \times d^1 + a_0 \times d^0 \text{ средствами десятичной арифметики.}$$

Пример 3. Переведем число $A=101101_2$ в десятичную систему.

В данном двоичном числе $n = 5$.

$$\text{Получаем } 101101_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 8 + 4 + 1 = 45_{10}$$

Пример 4. Переведем число $A=20D_{16}$ в десятичную систему.

В данном двоичном числе $n = 2$.

$$20D_{16} = 2 \cdot 16^2 + 0 \cdot 16^1 + 13 \cdot 16^0 = 512 + 13 = 525_{10}$$

Поскольку $8 = 2^3$, $16 = 2^4$, облегчается перевод восьмеричных и шестнадцатеричных чисел в двоичную форму записи и обратно.

Правило 3.

Перевод из восьмеричных и шестнадцатеричных чисел в двоичную систему осуществляется путем замены каждой цифры эквивалентной ей двоичной тройкой или четверкой цифр соответственно.

Пример 5. Переведем восьмеричное число 702_8 в двоичную систему.

$$702_8 = \underbrace{111}_7 \underbrace{000}_0 \underbrace{010}_2 = 111000010_2$$

Пример 6. Переведем шестнадцатеричное число $1A3_{16}$ в двоичную систему.

$$1A3_{16} = \underbrace{0001}_1 \underbrace{1010}_A \underbrace{0011}_3 = 110100011_2$$

Левые нули старшего разряда при переводе отбрасываем.

Правило 4.

Чтобы перевести число из двоичной системы в восьмеричную или шестнадцатеричную, его нужно разбить справа налево по три или четыре цифры и каждую такую группу заменить соответствующей восьмеричной или шестнадцатеричной цифрой.

Пример 7. Переведем двоичное число 1100101000_2 в шестнадцатеричную систему.

$$1100101000_2 = \underbrace{11}_3 \underbrace{0010}_2 \underbrace{1000}_8 = 328_{16}$$

Задание 2. Переведите данное число из шестнадцатеричной в двоичную систему, из нее в десятичную, из десятичной снова в шестнадцатеричную. Получилось тоже самое число?

№ варианта	16-ная	2-ная	10-ная	16-ная
1	A0C1	?	?	A0C1
2	A2B0	?	?	A2B0
3	C207	?	?	C207
4	9DA6	?	?	9DA6
5	F3C0	?	?	F3C0
6	8BC7	?	?	8BC7
7	F01F	?	?	F01F
8	9C3A	?	?	9C3A
9	428B	?	?	428B
10	70AF	?	?	70AF
11	12CD	?	?	12CD
12	FE98	?	?	FE98
13	E3E1	?	?	E3E1
14	D23C	?	?	D23C
15	49AB	?	?	49AB
16	B820	?	?	B820
17	9F1D	?	?	9F1D
18	C72C	?	?	C72C
19	E6A5	?	?	E6A5
20	501F	?	?	501F

Двоично-десятичная система счисления. Цифры от 0 до 9 представляются 4-разрядными двоичными комбинациями от 0000 до 1001, т.е. двоичными эквивалентами десяти первых шестнадцатеричных цифр. Преобразования из двоично-десятичной системы в десятичную систему (и наоборот) выполняются путем прямой замены четырех двоичных цифр одной десятичной цифрой (или обратной заменой). Например,

$$\begin{array}{c} (0011\ 0111)_{2-10} \\ \downarrow \quad \downarrow \\ (3\ 7)_{10} \end{array}$$

Задание 3. Переведите данное число из десятичной системы счисления в двоично-десятичную.

№ варианта	Десятичная	Двоично-десятичная
1	585 673 626	
2	285 846 163	
3	905 504 515	
4	483 412 738	
5	88 153 718	
6	325 112 713	
7	464 652 93	
8	342 758 430	
9	749 691 1039	
10	817 661 491	
11	596 300 515	
12	322 320 738	

13	780 949 718	
14	164 1020 713	
15	280 700 464	
16	728 383 202	
17	158 177 439	
18	328 537 634	
19	1026 725 100	
20	853 135 66	

Задание 4. Переведите данное число из двоично-десятичной системы счисления в десятичную.

№ варианта	Двоично-десятичная	Десятичная
1	010101010101 10011000 010000010110	
2	000101010001 010101010011 011010001000	
3	010010010100 001000000100 01110000	
4	001101011000 100010010010 010101000110	
5	000110000100 100110000111 100100011000	
6	100101100010 001001000110 011100110110	

7	000110010010 001100011000 011000010000	
8	010110010000 011101100101 011100010111	
9	100100010001 001000111001 001101100011	
10	100001010001 010000000111 001001110001	
11	001100100110 001000010110 010100010010	
12	000110000000 100101010110 011101100001	
13	0001000000010101 100110011001 001101100001	
14	011110000100 001100010001 100101010001	
15	010100110011 100100100101 100010010001	
16	001100110011 001101100010 010001000100	
17	000100110101 001010010011 0001000000100100	
18	000100000100 010110011001 100000110111	
19	100110010110 100100110010 000110010000	
20	100001111001 100000010000 001101000100	

Представление символьной информации в ЭВМ.

Символьная (алфавитно-цифровая) информация хранится и обрабатывается в ЭВМ в форме цифрового кода. Необходимый набор символов, предусмотренный конкретной ЭВМ, обычно включает в себя:

- буквенно-цифровые знаки алфавита (алфавитов);
- специальные знаки (пробел, скобки, знаки препинания и др.);
- знаки операций.

Кроме того, в состав набора входят управляющие символы, соответствующие определенным функциям.

Среди наборов символов наибольшее распространение получили знаки кода ASCII (American Standart Code for Information Interchange) – американский стандартный код обмена информацией.

Восьмибитовая кодировка символов состоит из двух таблиц кодирования: базовой и расширенной. Базовая таблица построена по стандарту ASCII и одинакова для всех компьютеров. Первые 32 кода (с 0 до 31) содержат управляющие коды, с кода 32 по код 127 размещены коды символов английского алфавита, знаков препинания, цифр, арифметических операций и некоторых вспомогательных символов.

Расширенная таблица относится к символам с номерами от 128 до 255 и может отличаться на компьютерах разного типа. В ней содержатся символы псевдографики, символы национального алфавита, специальные знаки.

Основная таблица ASCII

	00	10	20	30	40	50	60	70
0		►		0	@	P	`	p
1	☐	◄	!	1	A	Q	a	q
2	☐	⚡	"	2	B	R	b	r
3	♥	!!	#	3	C	S	c	s
4	♦	¶	\$	4	D	T	d	t
5	⚡	§	%	5	E	U	e	u
6	♣	_	&	6	F	V	f	v
7	•	⚡	'	7	G	W	g	w
8	☐	↑	<	8	H	X	h	x
9	○	↓	>	9	I	Y	i	y
A	☐	→	*	:	J	Z	j	z
B	♂	←	+	;	K	[k	{
C	♀	└	,	<	L	\	l	!
D	ℙ	↔	-	=	M]	m	}
E	ℙ	▲	.	>	N	^	n	~
F	※	▼	/	?	O	_	o	△

Расширенная таблица ASCII (cp866)

	80	90	A0	B0	C0	D0	E0	F0
0	А	Р	а	⌘	Ⓕ	Ⓖ	р	≡
1	Б	С	б	⌘	Ⓕ	Ⓖ	с	±
2	В	Т	в	⌘	Ⓕ	Ⓖ	т	>
3	Г	У	г		Ⓕ	Ⓖ	у	<
4	Д	Ф	д	Ⓕ	—	Ⓖ	ф	ѓ
5	Е	Х	е	Ⓕ	+	Ⓖ	х	Ј
6	Ж	Ц	ж	Ⓕ	Ⓖ	Ⓖ	ц	÷
7	З	Ч	з	Ⓕ	Ⓖ	Ⓖ	ч	≈
8	И	Ш	и	Ⓕ	Ⓖ	Ⓖ	ш	°
9	Й	Щ	й	Ⓕ	Ⓖ	Ⓖ	щ	.
А	К	Ь	к	Ⓕ	Ⓖ	Ⓖ	ь	.
В	Л	Ы	л	Ⓕ	Ⓖ	■	ы	Ј
С	М	Ь	м	Ⓕ	Ⓖ	■	ь	п
Д	Н	Э	н	Ⓕ	=	◐	э	2
Е	О	Ю	о	Ⓕ	Ⓖ	◐	ю	●
Ф	П	Я	п	Ⓕ	Ⓖ	■	я	

Задание 5. Зашифруйте данный текст, используя таблицу ASCII-кодов.

№ варианта	Текст
1	MS Office 2007
2	Автоматизация
3	Информатика
4	Computer
5	Printer
6	Компьютеризация
7	YAMANA
8	Световое перо
9	Микропроцессор
10	Принтер
11	Дисковод
12	Pentium 100
13	Арифмометр
14	Сканер
15	Винчестер
16	Windows XP
17	Ноутбук
18	Клавиатура
19	IBM PC
20	Apple

Задание 6. Дешифруйте данный текст, используя таблицу ASCII-кодов.

№ варианта	Текст
1	8A AE AC AF EC EE E2 A5 E0
2	50 72 6F 67 72 61 6D
3	50 72 6F 63 65 64 75 72 65
4	84 88 91 8A 8E 82 8E 84
5	43 4F 4D 50 55 54 45 52
6	50 52 49 4E 54
7	4D 4F 44 45 4D
8	4C 61 73 65 72
9	88 AD E4 AE E0 AC A0 E2 A8 AA A0
10	42 69 6E 61 72 79
11	49 6E 66 6F 72 6D 61 74 69 6F 6E
12	91 A8 E1 E2 A5 AC A0 20 E1 E7 A8 E1 AB A5 AD A8 EF
13	AC AE A4 A5 AB A8 E0 AE A2 A0 AD A8 A5
14	A2 EB E7 A8 E1 AB A8 E2 A5 AB EC AD EB A9 20 ED AA E1 AF A5 E0 A8 AC A5 AD E2
15	43 6F 6D 70 75 74 65 72 20 49 42 4D 20 50 43
16	8A AE AC AF EC EE E2 A5 E0
17	50 72 6F 67 72 61 6D
18	50 72 6F 63 65 64 75 72 65
19	84 88 91 8A 8E 82 8E 84
20	43 4F 4D 50 55 54 45 52

Лабораторная работа №4.1

Составления алгоритмов линейной структуры

Цель работы: Изучение простых линейных алгоритмов.

Теоретическая часть:

Понятие алгоритма является одним из основных понятий современной информатики. Термин алгоритм происходит от *algorithmi* – латинской формы написания имени выдающегося математика IX века аль-Хорезми, который сформулировал правила выполнения арифметических операций.

Алгоритм – это последовательность выполнения операций, конечный набор правил, позволяющих чисто механически решать любую конкретную задачу из некоторого класса однотипных задач.

Свойства алгоритма:

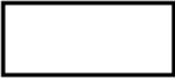


1. Массовость – возможность применения одного и того же алгоритма для решения целого класса однотипных задач с различными исходными данными.
2. Детерминированность (Определённость) – процесс применения правил к исходным данным (путь решения задачи) определён однозначно.
3. Понятность – исполнитель знает, что нужно для выполнения этого алгоритма
4. Дискретность – исполнение алгоритма расчленяется на выполнение отдельных шагов, выполнение очередного шага начинается после завершения предыдущего
5. Конечность – Исполнение алгоритма заканчивается после выполнения конечного числа шагов или за конечное время (не закичивается, не "зависает" – останавливается и пр.).
6. Результативность – на каждом шаге процесса известно, что считать результатом процесса





Существует несколько способов записи алгоритмов, отличающихся друг от друга наглядностью, компактностью, степенью формализации. Наибольшее распространение получили: словесный, графический, программный.

Графический способ записи предполагает использование определенных графических символов – блоков, каждый из которых обозначает определенный тип действия.

Каждый блок предписывает выполнение определенных действий. Совокупность блоков образует схему алгоритма или блок-схему.

Обозначение некоторых блоков в соответствии с ГОСТ 19.701-90 СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ ДАННЫХ И СИСТЕМ

Наименование символа	Обозначение символа	Пояснение
Процесс		Арифметический блок, определяющий действия, которые необходимо выполнить
Предопределенный процесс		Обращение к подпрограмме
Принятие решения		Логический блок, проверяющий истинность или ложность некоторого условия

Наименование символа	Обозначение символа	Пояснение
Передача данных		Ввод или вывод информации
Прерывание		Начало, конец, пуск, останов, вход в подпрограмму
Модификация		Организация циклического процесса при известном количестве повторений
Граница цикла		Начало и конец цикла. Условие помещают внутри символа в начале или конце в зависимости от расположения операции проверки условия

Блоки соединяются линиями потока информации. Внутри блоков записываются выполняемые действия. Линии со стрелками определяют направление вычислений. Практически любой сложный алгоритм можно представить собой комбинацией трех типов базовых структур: линейного, разветвляющегося и циклического.

Линейный алгоритм (следование) состоит из последовательности операций, выполняющихся только один раз в порядке их следования.

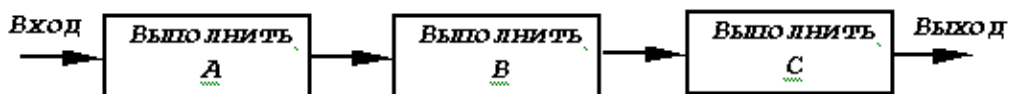


Рис. 1 Линейная структура алгоритма

Задача:

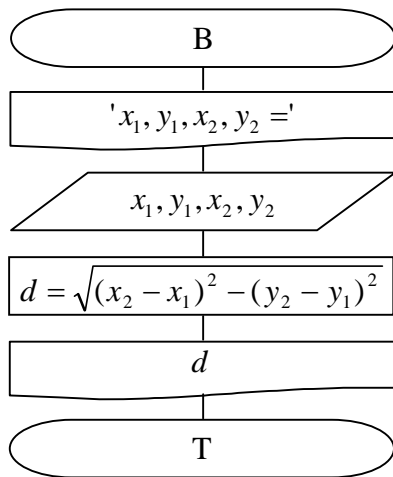
Задача 1: Дано две точки с координатами $M1(x_1, y_1)$ и $M2(x_2, y_2)$. Найти расстояния между ними.

Математическое моделирование:

1. Математический модель: Используем формулу нахождения расстояния между точек $M1(x_1, y_1)$ и $M2(x_2, y_2)$. Формула является следующим:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

2. Построим алгоритм соответствующей формуле:



Задания:

Простые задания:

1. Дано кубик размером x . Найти объем.
2. Дано 3 круга, радиус которых r_1, r_2, r_3 . Найти площадь кругов.
3. Дано треугольник, площадь которого s и высота h . Найти его основу.
4. Найти площадь шара, радиус которого r .
5. Дано треугольник со сторонами a, b и c . Найти периметр треугольника.
6. Найти площадь параллелепипеда, с основаниями a, b и c высотой h .
7. Чему равен объем Конуса, если высота h и радиус r ?
8. За какое время пройдет автомобиль расстояния s , если его скорость равен v ?
9. Брошено тело с высоты h . Найти время падение тела?
10. Если в 1 с капает 1 мл воды, то сколько литров натечет за x год.
11. Даны числа от 1 до n . Найти сумму всех чисел.
12. Найти тяжесть тела с массой m .
13. Какая сила воздействует на тело, если его масса m , а ускорение a .
14. Напряжение цепи U и сопротивление R . Найти силу тока, проходящую в цепи.
15. В цепи сопротивления R_1, R_2 и R_3 соединены параллельно. Найти общую сопротивление в цепи.

Задачи повышенной сложности:

$$1. \quad f_1 = \frac{2 \operatorname{tg}(x + \frac{\pi}{6})}{\frac{1}{3} + \cos^2(y + x^2)} + \log_2^{(x^2+2)} \quad x, y - \text{вещественный}$$

$$2. \quad f_2 = \frac{1/(x + 2/x^2 + 3/x^3) + e^{x^2+3x}}{\operatorname{arctg}(x + y) + |5 + x|^2} - \cos^2(y^2 + x^2/2);$$

$$3. \quad z = \ln \left| (x + y)^2 + \sqrt{|y| + 2} - \left(x - \frac{xy}{x^2/2 - 5} \right) \right| + \frac{\cos^2(x + y)}{(x + y)^{1/3}}; \quad x, y - \text{целый}$$

4. $T_{11} = \frac{x^2 + 1}{x^2 + \frac{xy + y^2}{y^2 + \frac{y + xy}{|xy| + 5}}} + \frac{1}{1 + \cos x + \frac{1}{\sin|x|}}$ x, y – вещественные числа.
5. $T = \sqrt[5]{a} + \sqrt[4]{b \cdot \frac{ax^2 + b}{2 \cdot b + a \cdot b}} \cdot (a^2 + x^2 + b^2 + 2)$ a, b – вещественный, x – целый
6. $F = \frac{|\sin^2|cx_2^3 + dx_1^3 - cd|}{\sqrt{(cx_1^2 + dx_2^2 + 5) + 2}} + \operatorname{tg}(x_1 \cdot x_2^2 + d^3)$; x_1, x_2 – вещественный, c, d – целые.
7. $y^2 = \frac{ax^2 + bx + c}{xa^3 + a^2 + a^{b-c}} + \cos\left|\frac{ax + b}{cx + d + 2^c}\right|$ a, b, c, d – целые, x – вещественный.
8. $W_1 = 0.75 + \frac{8.2x^2 + \sqrt{|x^3 + 3x|} + \cos(x - 2)}{a/4 + b/3 + c/2 + 1}$ a, b, c – целые, x – вещественный.
9. $TT = \frac{\sqrt{x-1} + \sqrt{x+2} + \lg(\sqrt{ax^2} + 2)}{\sqrt[2]{\sqrt{x+2} + \sqrt{x+24} + x^5}}$ x – вещественный, a – целый.
10. $W_2 = \sqrt{e^{-xy} - x \cdot \sin(ax) - \frac{x^2 + 2}{|x| + 5}} + \sqrt{\ln(x^2 + 2) + 5}$ a – целый, x, y – вещественные.
11. $AA = \sqrt{\frac{2\operatorname{tg}(x+2) - \cos(x+2^x)}{1 + \cos^2(x+2)}} + \frac{\sin x^2}{x^2 + 3}$ x – вещественное число.
12. $BB_1 = x \cdot \sin(x/2 + x/3 + x/4) + \frac{\lg(x^2 - 2) + 3^a}{\cos(x+3) \cdot \sin(x+3) + 8}$ a – целый, x – вещественный.
13. $TT = \sqrt{y^2 + e^x} + \sqrt{e^x + \frac{a}{x^2 + 2}} + \frac{\cos^2 x}{\sin x^2} + \cos^3 x$; a – целый, x – вещественный.
14. $AF = 2^{-x} \cdot \sqrt{x + \sqrt[4]{|y| + 2}} \cdot \sqrt[3]{e^{x-1} / \sin(z+2) + 2}$; x – целый, y, z – вещественный.

Лабораторная работа №4.2

Составления алгоритмов разветвляющейся структуры

Цель работы: Использование условно и безусловна проходящих операторов, операторов выбора и научиться составлять программы, используя оператор выбора.

Теоретическая часть :

Разветвляющийся алгоритм (ветвление) обеспечивает выбор между двумя альтернативами.

Выполняется проверка, а затем выбирается один из путей.

Подобная структура называется также «ЕСЛИ – ТО – ИНАЧЕ», или «развилка». Каждый из путей (ТО или ИНАЧЕ) ведет к общей точке слияния, так что выполнение программы продолжается независимо от того, какой путь был выбран.

Может оказаться, что для одного из результатов проверки ничего предпринимать не надо. В этом случае можно применять только один обрабатывающий блок.



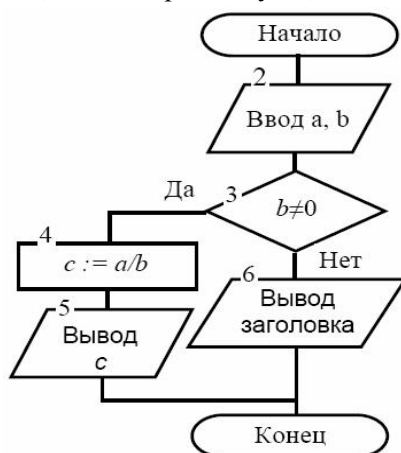
Рис. 1 Разветвляющаяся структура алгоритма



Рис. 2 Структура «неполное ветвление»

Пример 2. Составить алгоритм и написать программу вычисления частного двух чисел.

Программа должна проверять правильность введенных пользователем данных и, если они неверные (делитель равен нулю), выдавать сообщение об ошибке.



Задача

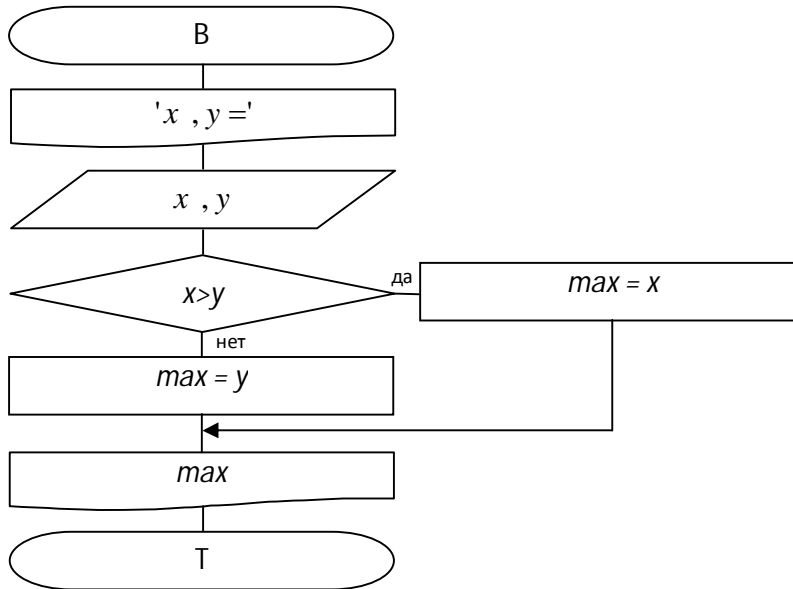
Задача1: Дано две вещественные числа x и y . Составьте программу, которая выводит на экран большее из них (если они равны, то выводите любое из них).

Математическое моделирование :

1. Математическая модель:

$$\max = \begin{cases} x, & \text{если } x > y \\ y, & \text{если } x \leq y \end{cases}$$

2. Составляем алгоритм, соответствующей формуле.



Задача2: Вычислите данное выражение. x – вещественный, a , b и c целые.

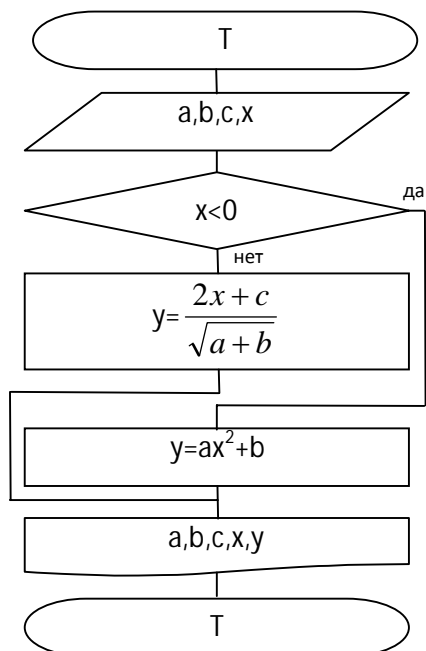
$$y = \begin{cases} ax^2 + b, & \text{если } x < 0 \\ \frac{(2x + c)}{\sqrt{a + b}}, & \text{если } x \geq 0 \end{cases}$$

Математическое моделирование:

1. Математическая модель:

$$y = \begin{cases} ax^2 + b, & \text{если } x < 0 \\ \frac{(2x + c)}{\sqrt{a + b}}, & \text{если } x \geq 0 \end{cases}$$

2. Составляем алгоритм, соответствующей формуле.



Задания:

Простые задания:

1. Дано вещественные числа x и y . Вычислите: $\max(x, y)$ и $\min(x, y)$
2. Дано вещественные числа x, y и z . Вычислите: $\max(x, y, z)$ и $\min(x, y, z)$
3. Дано вещественные числа x, y и z . Вычислите: $\max(x+y+z, x, y, z)$ и $\min^2(x+y/2, x, y, z)$
4. Дано вещественные числа a, b и c . Проверьте: выполняются ли неравенство $a < b < c$?
5. Дано вещественные числа a, b и c . Если выполняется неравенство $a \geq b \geq c$, то удвойте вещественные числа, в ином случае поменяйте их с модулями.
6. Дано два вещественных числа. Выводите первое число, если оно больше второго, в ином случае выводите их двоих.
7. Дано два вещественных числа. Поменяйте первое число с нулем, если оно меньше или равно второму, в ином случае оставьте без изменений.
8. Дано три вещественных числа. Выбирайте из них числа, которые относятся к интервалу $[1, 3]$.
9. Дано вещественные числа x, y (x и y не равны). Поменяйте меньшее из них с половиной их суммы, а большее с удвоенным произведением.
10. Дано три вещественных числа. Поменяйте положительных с их квадратами.
11. Если неравные вещественные числа x, y, z меньше одного, то минимального из этих трех чисел поменяйте половиной суммы других двух чисел. В ином случае оставьте без изменений.
12. Даны вещественные числа a, b, c и d . Если выполняется неравенство $a \leq b \leq c \leq d$, то каждого из них поменяйте самым большим.
13. Дано вещественные числа x, y . Если x и y отрицательны, поменяйте их модулями; если только один отрицательный, то каждого производить 0.5.
14. Даны вещественные положительные числа x, y, z . Существует ли треугольник со сторонами длиной x, y, z ?
15. Дано вещественные числа a, b, c ($a \neq 0$). Определите, имеет ли решение квадратное уравнение $ax^2 + bx + c = 0$. Если существуют настоящие решения, то найти их. В ином случае сообщите о том, что не существует решений.

Задачи повышенной сложности:

$1 \quad Q = \begin{cases} \frac{ax^2 + \sqrt[3]{bc}}{2ab}, & \text{если } ab \neq 0 \\ \frac{\sin^2 x + b^3}{\ln ax^2} & \text{если } a > 0 \end{cases}$	$2 \quad y = \begin{cases} \frac{\sin^2 ax + \cos^3 ax^2}{(a+b)^2 + c}, & \text{если } a < 0 \\ \frac{\sqrt{a^2 - b^3} + 2ac}{\sin^2 a + bx^2}, & \text{если } a^2 \geq b^3 \end{cases}$	$3 \quad y = \begin{cases} \frac{a + b^2 + 2abx}{\sqrt[3]{c^2 + b^3}}, & \text{если } x > 0 \\ \frac{a^2 + 2ab + c^2}{7a^2 + 8b^3}, & \text{если } a \neq 0 \end{cases}$
$4 \quad y = \begin{cases} \frac{a \sin x + b \cos x^2}{a^2 + b^2}, & \text{если } ab \neq 0 \\ \frac{a^3 + 2ab^2 + c^3}{a + b^2}, & \text{если } b > 0 \end{cases}$	$5 \quad Z = \begin{cases} \frac{2x^3 + ax^2 + c^3}{a^2 - c^2}, & \text{если } a > c \\ \frac{2ab + \sin ax^2}{\sqrt{a^2 + 2q^2}}, & \text{если } q \neq 0 \end{cases}$	$6 \quad y = \begin{cases} \frac{2c^2 + ab \cos x^2}{a \sin x^2 + b^3}, & \text{если } a > 0 \\ \frac{a^2 + 2bc^3}{\sqrt{a^3 + 2c^2 + d^3}}, & \text{если } d \neq 0 \end{cases}$
$7 \quad Q = \begin{cases} \frac{a + b^2 + 2c^3}{a \cdot \sin^2 x^2}, & \text{если } a > b \\ \frac{a^2 + 2ab + c^2}{a + 2c^2 + xy}, & \text{если } xy > a \end{cases}$	$8 \quad Z = \begin{cases} \sqrt[3]{cy^2 + a^3} - \ln x, & \text{если } x > 0 \\ \frac{a^2 + bx + c^2}{\sqrt{a^3 - b^3}}, & \text{если } a^3 > b^3 \end{cases}$	$9 \quad y = \begin{cases} \frac{ax^2 + cx^3}{\sqrt{a^2 - b} + c}, & \text{если } a^2 > b \\ \frac{\sin^2 x^3}{\cos x^2 + ab}, & \text{если } x \neq 0 \end{cases}$
$10 \quad Q = \begin{cases} \frac{2x + ax^2 + c^3}{a^2 + b^2}, & \text{если } x > c \\ \frac{a(x^2 + y) + 2xy^2}{a^2 - b^2}, & \text{если } a \neq b \end{cases}$	$11 \quad y = \begin{cases} \frac{ax^2 + b + 2c^3}{x^2 + g^2}, & \text{если } x \neq g \\ \frac{2c^2 + \sqrt[5]{a^2}}{c^3 + ax^2 + d}, & \text{если } x < 0 \end{cases}$	$12 \quad Z = \begin{cases} \frac{2x^2 + ab^3 + c^2}{2a + c^2}, & \text{если } x \neq 0 \\ \frac{2a^2 + 7b^3 + 12c}{2x + 7ax^2 + c^3}, & \text{если } c > 0 \end{cases}$
$13 \quad U = \begin{cases} \frac{a^2 + 4abc + c^2}{a - b^2}, & \text{если } a > b^2 \\ \sqrt[3]{\frac{2a}{c^2}} + \sqrt[5]{\frac{c^2}{x^2}}, & \text{если } x \neq 0 \end{cases}$		

Лабораторная работа №4.3

Составления алгоритмов циклические структуры

Цель работы: Создание повторяющегося процесса на языке Паскаль:

А) «while» – Создание повторяющегося процесса для поставленных ранее задач;

В) «repeat until» – Создание повторяющегося процесса для поставленных в конце задач;

С) Научиться создавать с помощью параметрического цикла повторяющегося процесса и его использование.

Теоретическая часть:

Циклический алгоритм (Цикл) содержит некоторую последовательность операций, выполняемую многократно. Основной блок цикла, тело цикла, производит требуемые вычисления. Остальные блоки организуют циклический процесс: устанавливают начальные и новые значения данных, проверяют условия окончания или продолжения циклического процесса.

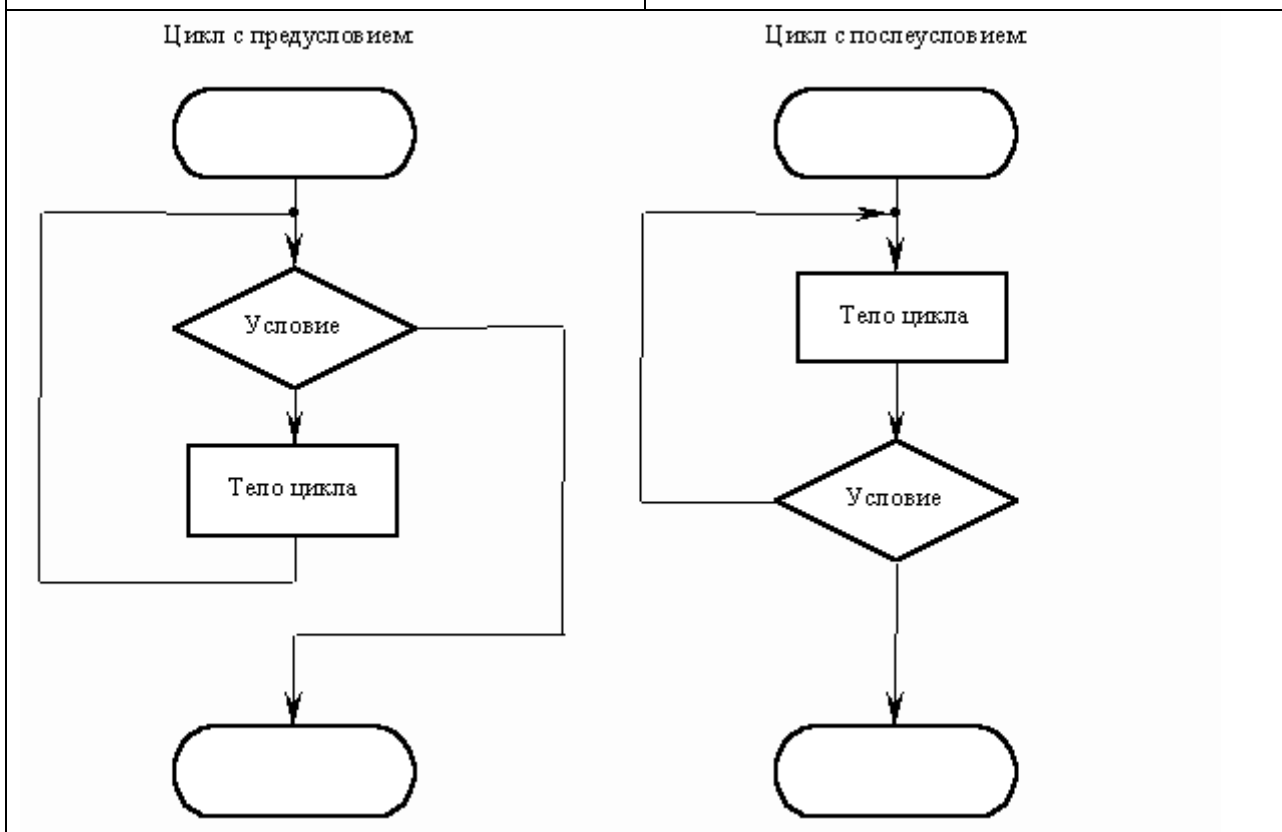
Различают два типа структур цикла: *цикл с параметром* или *с повторением* и *цикл с условием*. Циклический алгоритм позволяет компактно описать большое число одинаковых вычислений над разными данными для получения необходимого результата.

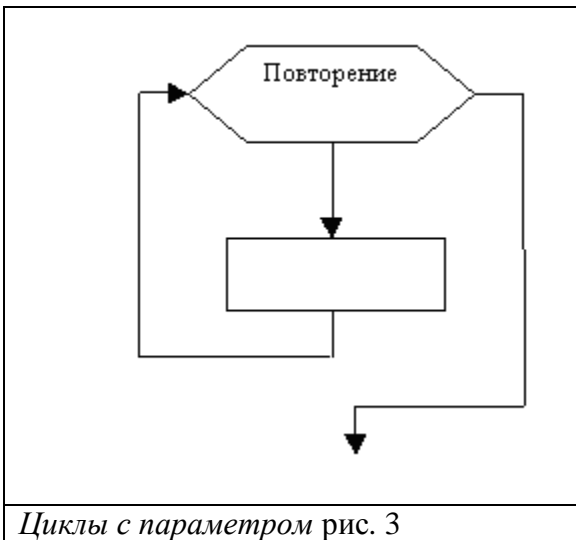
Циклы с параметром используют тогда, когда количество повторов тела цикла заранее известно. Схематично такой цикл изображен на рис. 3.

Циклы с условием используются тогда, когда число повторений заранее неизвестно, но задано условие окончания цикла. Причем, если условие окончания цикла проверяется перед выполнением тела цикла, то такие циклические структуры называют циклами *с предусловием* («Выполнять пока» рис. 1), а если проверка условия происходит после выполнения тела цикла – циклами *с постусловием* («Выполнять до тех пор пока не» рис.2).

«Выполнять пока» рис. 1

«Выполнять до тех пор пока не» рис.2





Пример

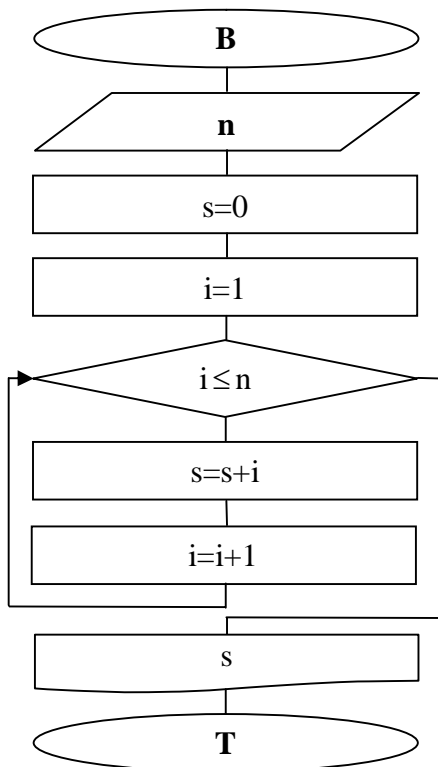
Пример1: найти сумму чисел от 1 до n.

Математический просчёт:

1. Математическая модель:

$$s = \sum_{i=1}^n i$$

2. По формуле строим алгоритм:



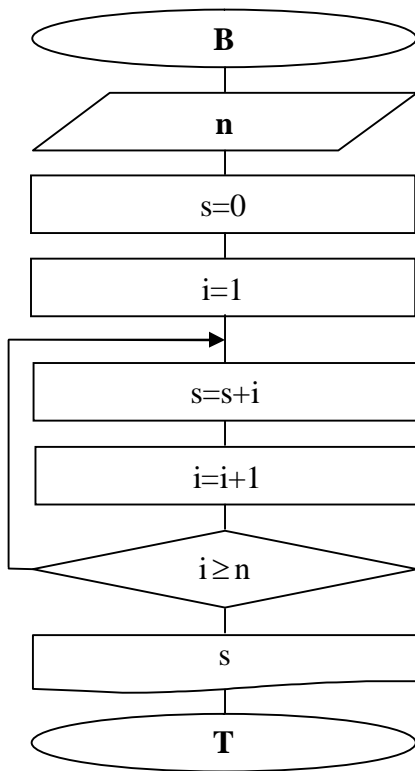
Пример2: найти сумму чисел от 1 до n.

Математический просчёт:

1. Математическая модель:

$$s = \sum_{i=1}^n i$$

2. По формуле строим алгоритм:



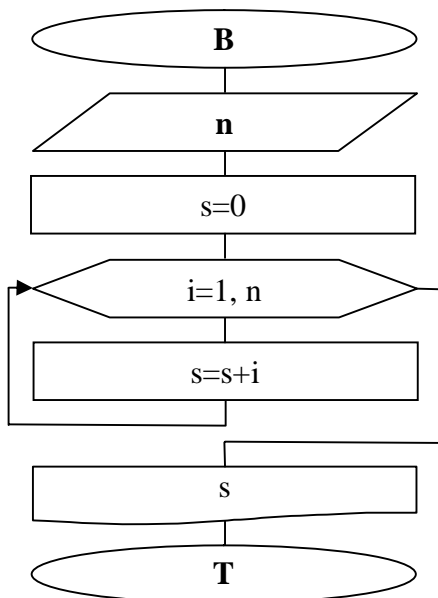
Пример2: найти сумму чисел от 1 до n.

Математический просчёт:

1. Математическая модель:

$$s = \sum_{i=1}^n i$$

2. По формуле строим алгоритм:



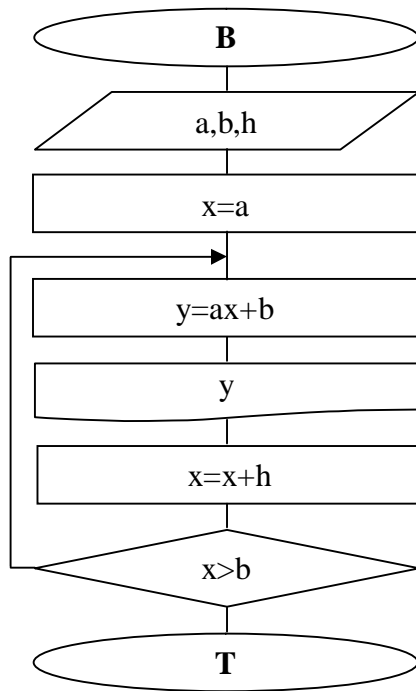
Пример4: $y = ax + b$, $a \leq x \leq b$, h - Дано натуральное число.

Математический просчёт:

1. Математическая модель:

$$y = ax + b, \quad a \leq x \leq b, \quad h$$

2. По формуле строим алгоритм:



Задания:

Простые задания:

1.
$$S = \frac{x-1}{1} - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \dots + \frac{(-1)^{n-1}(x-1)^n}{n}$$
2.
$$P = 1 - \frac{x^2}{1} + \frac{x^4}{2} - \frac{x^6}{3} + \dots + \frac{(-1)^n x^{2n}}{n}$$
3.
$$S = 1 - x^2 + x^4 - x^6 + \dots + (-1)^n \cdot x^{2n}$$
4.
$$SS = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + \frac{(-1)^{n-1} x^{2n-1}}{n}$$
5.
$$S = x + \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{x^{2n-1}}{n}$$
6.
$$P = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + \frac{(-1)^{n+1} x^{2n-1}}{2n-1}$$
7.
$$S = x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n-1}}{(2n-1)}$$
8.
$$PP = 1 + \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$
9.
$$S = k + \frac{k^2}{2} + \frac{k^3}{3} + \dots + \frac{k^n}{n}$$
10.
$$S = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$
11.
$$S = \frac{x}{1!} + \frac{x}{2!} + \frac{x}{3!} + \dots + \frac{x}{n!}$$
12.
$$S = \frac{\pi}{10} + \frac{\pi^3}{10^3} + \frac{\pi^5}{10^5} + \dots + \frac{\pi^{2n+1}}{10^{2n+1}}$$
13.
$$S = \frac{\pi}{10} + \frac{\pi}{10^3} + \frac{\pi}{10^5} + \dots + \frac{\pi}{10^{2n+1}}$$

Задачи повышенной сложности:

1. $y = \sqrt[3]{\frac{ax+b}{b^2 + \cos^2 x}} - \frac{\sin x^2}{ab}$, $a \leq x \leq c, h = 0.3$
2. $y = \sqrt[3]{\frac{\sin ax + b^{2c}}{b^2 + \cos^2 x}} - \frac{\sin x^2}{ab}$, $c \leq x \leq d, n = 25$
3. $y = \frac{a^b + b^x + c^a}{2x^2 + 3a^{x+c}}$, $3 \leq x \leq 5, h = 0.2$
4. $y = \sqrt[3]{a^a} + x^2 \cos ax$, $-\frac{\pi}{2} \leq x \leq \pi; h = \frac{\pi}{19}, a = 3.26$
5. $y = a \cos x - \sin x^2$, $0 \leq x \leq 10, h = 0.5$
6. $y = a^2 + \sqrt[5]{\frac{b + \sin x}{a^3 + \cos^2 x^3}}$, $1 \leq a \leq 12; n = 20$
7. $y = \frac{ax^2}{b} + \frac{x}{c}$, $1 \leq x \leq 10, h = 8, a = 3, b = 12, c = 6$
8. $y = \frac{a^2 + bx + x^c}{a^2 + b^2 + x^2}$, $5 \leq x \leq 10, h = 0.4$
9. $y = \sqrt[3]{\frac{\sin ax + b^{2c}}{b^2 + \cos^2 x}} - \frac{\sin x^2}{ab}$, $-1 < x < 1, n = 50$
10. $y = \frac{ax^2 + bx + 4}{a^2 + b^2 + x^2}$, $1 \leq x \leq 20, n = 100$
11. $y = a^2 \cos x + \frac{\sin x}{2} + bx^2$, $c \leq x \leq e, h = 0.2$
12. $y = 2\sqrt[3]{a^{\sin 2x}} + x^2 \cos ax$, $-\frac{\pi}{2} \leq x \leq \pi; n = 10$
13. $y = \sqrt[k]{\frac{ax+b}{b^2 + \cos^2 x}} - \frac{\sin x^2}{ab}$, $d \leq x \leq c, n = 15$
14. $y = \sqrt{\frac{\sin ax + b^{2c}}{b^2 + \cos^2 x}} - \frac{\sin x^2}{ab}$, $0 \leq x \leq 1, n = 50$
15. $y = \frac{\log a^{2\sin x} + e^{2x}}{\arctan x + 2}$, $-\pi \leq x \leq \pi, n = 10$

ЛАБОРАТОРНАЯ РАБОТА №5

СРЕДА ПРОГРАММИРОВАНИЯ VISUAL C++.

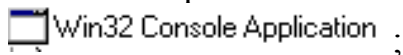
ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ


1.1. Консольный режим работы среды Visual C++ 6.0

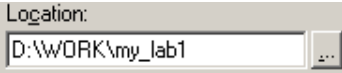

Программа, создаваемая в среде Visual C++, всегда оформляется в виде отдельного проекта. Проект (*project*) – это набор взаимосвязанных исходных файлов, предназначенных для решения определенной задачи, компиляция и компоновка которых позволяет получить выполняемую программу. В проект входят как файлы, непосредственно создаваемые программистом, так и файлы, которые автоматически создает и редактирует среда программирования.


Для создания нового проекта необходимо:

- выбрать в главном меню **File – New**;
- в открывшемся окне (закладка **Projects**) выбрать тип создаваемого проекта

 Win32 Console Application ;

- в поле **Project Name** ввести имя проекта  ;
- в поле **Location** ввести имя папки, в которой будет размещен проект, и

полный путь к ней  . Папку также можно выбрать, используя диалоговое окно **Choose Directory**, для чего надо щелкнуть мышью по кнопке  ;

- щелкнуть мышью по кнопке **OK**;
- в открывшемся окне мастера приложений **Win32 Console Application – Step 1 of 1** выбрать  **An empty project.** (пустой проект) и щелкнуть по кнопке

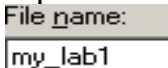
 ;

• в открывшемся окне **New Project Information** (информация о новом проекте) щелкнуть мышью по кнопке **OK**.

Для работы с консольным приложением необходимо создать **новый** или добавить **существующий** файл с кодом программы.

Для создания нового файла необходимо:



- выбрать **File – New**;
- в открывшемся окне (закладка **Files**) выбрать тип файла  ;

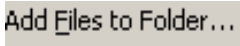
• в поле **File name** ввести имя файла:  (желательно, чтобы вводимое имя совпадало с именем проекта);

- щелкнуть мышью по кнопке **OK**.

Для добавления в проект существующего файла с кодом программы необходимо:

- скопировать имеющийся файл (расширение **cpp**) в рабочую папку проекта;

- в окне **Workspace** (закладка ) щелкнуть правой кнопкой мыши по папке  Source Files ;

- в раскрывшемся меню выбрать пункт , в диалоговом окне **Insert Files into Project** указать добавляемый файл и щелкнуть мышью по кнопке **ОК**.

В папке проекта, как правило, размещено несколько файлов и одна вложенная папка. Файлы имеют следующее назначение:

- файл с расширением ***.dsw** (например, *my_lab1.dsw*) – файл проекта, который объединяет все входящие в проект файлы;
- файл с расширением ***.cpp** (например, *my_lab1.cpp*) – файл кода программы.

1.2. Функции библиотеки *math.lib*

Функции для расчета математических выражений находятся в библиотеке **math.lib** (подключение библиотеки: **#include <math.h>**). Все аргументы в тригонометрических функциях задаются в *радианах*. Параметры и аргументы всех остальных функций имеют тип **double** (кроме **abs(x)**).

Математическая функция	Функция библиотеки math.lib	Описание
$ x $	abs(x)	Вычисление абсолютного значения (только для целых чисел!)
$ x $	fabs(x)	Вычисление абсолютного значения x
\sqrt{x}	sqrt(x)	Вычисление квадратного корня x
x^y	pow(x, y)	Возведение x в степень y
$\sin(x)$	sin(x)	Вычисление синуса x
$\text{sh}(x) = (e^x - e^{-x})/2$	sinh(x)	Вычисление синуса гиперболического x
$\cos(x)$	cos(x)	Вычисление косинуса x
$\text{ch}(x) = (e^x + e^{-x})/2$	cosh(x)	Вычисление косинуса гиперболического x
$\text{tg}(x)$	tan(x)	Вычисление тангенса x
$\text{tgh}(x)$	tanh(x)	Вычисление тангенса гиперболического x
$\arccos(x)$	acos(x)	Вычисление значения арккосинуса x
$\arctg(x)$	atan(x)	Вычисление значения арктангенса x
$\arctg(x/y)$	atan2(x,y)	Вычисление значения арктангенса двух аргументов x и y
e^x	exp(x)	Вычисление экспоненты числа x
$\ln(x)$	log(x)	Вычисление натурального логарифма x
$\lg_{10}(x)$	log10(x)	Вычисление десятичного логарифма x

Математическая функция	Функция библиотеки math.lib	Описание
Округление к большему	ceil(x)	Функция возвращает действительное значение, соответствующее наименьшему целому числу, которое больше или равно x
Округление к меньшему	floor(x)	Функция возвращает действительное значение, соответствующее наибольшему целому числу, которое меньше или равно x
Остаток от деления x на y	fmod(x,y)	Функция возвращает действительное значение, соответствующее остатку от целочисленного деления x на y

Например:

```
double x, y,
double z = exp(1)+exp(x*x)+exp(2*pow(x,3));    // z = e + e^{x^2} + e^{2x^3}
double z1 = pow (x, pow (y, 4+pow (x, 1/4.)));  // z1 = x^{y^{4+4x}}
double z2 = fmod( 5, 2);                       // или z2 = 5%2; => z2 = {5/2} = 1
double z3 = ceil (5.6);                         // или z3 = ceil (5.1); => res3 = 6
double z4 = ceil (-5.1);                       // или z4 = ceil (-5.8); => res4 = -5
double z5 = floor (5.6);                       // или z5 = floor (5.1); => res5 = 5
double z6 = floor (-5.6);                      // или z6 = floor (-5.1); => res6 = -6
```

1.3. Пример выполнения работы

Написать программу вычисления линейного арифметического выражения

$$h = \frac{x^{2y} + e^{y-1}}{1 + x|y - \operatorname{tg}z|} + 10 \cdot \sqrt[3]{x} - \ln(z).$$

При $x = 2,45$; $y = -0,423 \cdot 10^{-2}$; $z = 1,232 \cdot 10^3$ **ответ:** $h = 6,9465$.

Код программы

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

int main ()
{
    double x, y, z, a, b, c, h;           // Начало главной функции
                                         // Объявление переменных

    cout << "\Vvedite x: ";
    cin >> x;
    cout << "\Vvedite y: ";
    cin >> y;
    cout << "\Vvedite z: ";
    cin >> z;
```

```

a = pow(x, 2*y) + exp(y-1);           // Вычисление выражения
b = 1+x * fabs(y - tan(z));
c = 10 * pow(x, 1/3.) - log(z);
h = a / b + c;

cout << "Result h= " << h << endl;    // Вывод результата

return 0;                               // Завершение выполнения программы
}                                         // Конец главной функции

```

Замечания:

1. Для ввода значений переменных **x**, **y** и **z** необходимо набрать с клавиатуры:

```

2.45      (x = 2,45)
-0.423e-2 (y = -0,423·10-2)
1.232e3   (z = 1,232·103)

```

2. В языке C++ при вычислении арифметических выражений происходит *автоматическое приведение типов*, следовательно, при делении целого значения на целое, результат будет *целым числом*. Например, при вычислении “**1/3**” результат будет равен нулю, так как целая часть вычисленного выражения равна нулю. Для получения результата, имеющего дробную часть, необходимо, чтобы один из операндов имел действительный тип. Для этого можно использовать функцию *явного приведения типа*, а для констант достаточно *поставить точку после числа*, например: “**1/3.**”, или “**1./3.**”, или “**1./3**”.

Например:

```

int s, n;
double sr = static_cast<double> (s) / n;    // явное приведение типа
double y = pow (x, 3/4.);                  // y =  $\sqrt[4]{x^3} = x^{\frac{3}{4}}$ 

```

3. Язык C чувствителен к регистру букв, т.е. прописные и строчные буквы воспринимаются как *разные* символы.

Например: count, Count, COUNT – разные идентификаторы.

4. При выводе информации для перехода на новую строку применяется манипулятор (функция управления выводом) **endl** или ‘\n’, для выравнивания выводимой информации – ‘\t’ (вставляет символ табуляции).

5. Главная функция **int main ()** автоматически вызывается при запуске программы и возвращает операционной системе по окончании значение **0 (return 0;)**.

Для компиляции, компоновки и запуска программы на выполнение используются следующие пункты подменю **Build**:

Compile (Ctrl+F7) – *компиляция* выбранного файла. Результаты компиляции выводятся в окно **Output**.

Build (F7) – *компоновка* проекта. Компилируются все файлы, в которых произошли изменения с момента последней компоновки.

Rebuild All – *перекомпоновка* проекта. Компилируются все файлы проекта независимо от того, были ли в них произведены изменения или нет.

Execute (Ctrl+F5) – *выполнение* исполняемого файла, созданного в результате компоновки проекта. Для файлов, в которые были внесены изменения, выполняется перекомпилирование и перекомпоновка.

Если в процессе компиляции были обнаружены синтаксические ошибки, то выводится соответствующее сообщение. В этом случае необходимо последовательно исправить все ошибки и компилировать проект снова. Если синтаксических ошибок нет, но результат выполнения программы неверный, то необходимо искать логические ошибки. Для этого следует использовать встроенный в систему отладчик (см. лабораторную работу №2).

Для открытия сохраненного ранее проекта необходимо выбрать в меню **File – Open Workspace...** В открывшемся диалоговом окне выбрать папку проекта и открыть в ней файл с расширением **dsw**.

1.4. Индивидуальные задания

Написать программу вычисления значения выражения при заданных исходных данных. Сравнить полученное значение с указанным правильным результатом.

$$1. s = \frac{2 \cos\left(x - \frac{2}{3}\right)}{\frac{1}{2} + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2/5}\right)$$

при $x = 14,26$; $y = -1,22$; $z = 3,5 \cdot 10^{-2}$. **Ответ s = 0,749155.**

$$2. s = \frac{\sqrt[3]{9 + (x - y)^2}}{x^2 + y^2 + 2} - e^{|x-y|} \operatorname{tg}^3 z$$

при $x = -4,5$; $y = 0,75 \cdot 10^{-4}$; $z = -0,845 \cdot 10^2$. **Ответ s = -3,23765.**

$$3. s = \frac{1 + \sin^2(x + y)}{\left|x - \frac{2y}{1 + x^2 y^2}\right|} x^{|y|} + \cos^2\left(\operatorname{arctg} \frac{1}{z}\right)$$

при $x = 3,74 \cdot 10^{-2}$; $y = -0,825$; $z = 0,16 \cdot 10^2$. **Ответ s = 1,05534.**

$$4. s = |\cos x - \cos y|^{(1+2\sin^2 y)} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right)$$

при $x = 0,4 \cdot 10^4$; $y = -0,875$; $z = -0,475 \cdot 10^{-3}$. **Ответ s = 1,98727.**

$$5. s = \ln\left(y^{-\sqrt{|x|}}\right) \left(x - \frac{y}{2}\right) + \sin^2(\operatorname{arctg}(z))$$

при $x = -15,246$; $y = 4,642 \cdot 10^{-2}$; $z = 21$. **Ответ s = -182,038.**

$$6. s = \sqrt{10(\sqrt[3]{x} + x^{y+2})}(\arcsin^2 z - |x - y|)$$

при $x = 16,55 \cdot 10^{-3}$; $y = -2,75$; $z = 0,15$.

Ответ s = -40,6307.

$$7. s = 5 \operatorname{arctg}(x) - \frac{1}{4} \arccos(x) \frac{x + 3|x - y| + x^2}{|x - y|z + x^2}$$

при $x = 0,1722$; $y = 6,33$; $z = 3,25 \cdot 10^{-4}$.

Ответ s = -205,306.

$$8. s = \frac{e^{|x-y|} |x-y|^{x+y}}{\operatorname{arctg}(x) + \operatorname{arctg}(z)} + \sqrt[3]{x^6 + \ln^2 y}$$

при $x = -2,235 \cdot 10^{-2}$; $y = 2,23$; $z = 15,221$.

Ответ s = 39,3741.

$$9. s = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y - x) \frac{\cos y - \frac{z}{(y-x)}}{1 + (y-x)^2}$$

при $x = 1,825 \cdot 10^2$; $y = 18,225$; $z = -3,298 \cdot 10^{-2}$.

Ответ s = 1,21308.

$$10. s = 2^{-x} \sqrt{x + \sqrt[4]{|y|}} \sqrt[3]{e^{x-1/\sin z}}$$

при $x = 3,981 \cdot 10^{-2}$; $y = -1,625 \cdot 10^3$; $z = 0,512$.

Ответ s = 1,26185.

$$11. s = y^{\sqrt[3]{|x|}} + \frac{\cos^3(y)}{e^{|x-y|} + \frac{x}{2}} \cdot |x - y| \left(1 + \frac{\sin^2 z}{\sqrt{x+y}} \right)$$

при $x = 6,251$; $y = 0,827$; $z = 25,001$.

Ответ s = 0,712122.

$$12. s = 2^{(y^x)} + (3^x)^y - \frac{y \left(\operatorname{arctgz} - \frac{1}{3} \right)}{|x| + \frac{1}{y^2 + 1}}$$

при $x = 3,251$; $y = 0,325$; $z = 0,466 \cdot 10^{-4}$.

Ответ s = 4,23655.

$$13. s = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{|x - y|(\sin^2 z + \operatorname{tg} z)}$$

при $x = 17,421$; $y = 10,365 \cdot 10^{-3}$; $z = 0,828 \cdot 10^5$.

Ответ s = 0,330564.

$$14. s = \frac{y^{x+1}}{\sqrt[3]{|y-2|} + 3} + \frac{x + \frac{y}{2}}{2|x+y|} (x+1)^{-1/\sin z}$$

при $x = 12,3 \cdot 10^{-1}$; $y = 15,4$; $z = 0,252 \cdot 10^3$.

Ответ s = 82,8256.

$$15. s = \frac{x^{y+1} + e^{y-1}}{1 + x|y - \operatorname{tg} z|} \left(1 + |y - x| \right) + \frac{|y - x|^2}{2} - \frac{|y - x|^3}{3}$$

при $x = 2,444$; $y = 0,869 \cdot 10^{-2}$; $z = -0,13 \cdot 10^3$.

Ответ s = -0,498707.

ЛАБОРАТОРНАЯ РАБОТА №6

ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ

2.1. Логические операции и операции сравнения

Операции сравнения применяются при работе с двумя операндами и возвращают **true** (1), если результат сравнения – истина, и **false** (0), если результат сравнения – ложь. В языке C определены следующие операции сравнения: **<** (меньше), **<=** (меньше или равно), **>** (больше), **>=** (больше или равно), **!=** (не равно), **==** (равно).

Логические операции работают с операндами скалярных типов и возвращают результат булева типа. Существует три логические операции:

! – отрицание или логическое **НЕ**;

&& – логическое **И**;

|| – логическое **ИЛИ**.

2.2. Оператор условной передачи управления *if*

Форматы оператора **if**:

1. Полная форма: **if** (логическое_выражение) оператор_1;
else оператор_2;

Если логическое выражение истинно, то выполняется оператор_1, иначе – оператор_2.

2. Сокращенная форма: **if** (логическое_выражение) оператор_1;
Если логическое_выражение истинно, то выполняется оператор_1.

3. Вложенная форма:

if (логическое_выражение_1) оператор_1;
else if (логическое_выражение_2) оператор_2;
else оператор_3;

Если логическое_выражение_1 истинно, то выполняется оператор_1, иначе, если логическое_выражение_2 истинно, то выполняется оператор_2, иначе выполняется оператор_3.

2.3. Оператор множественного выбора *switch*

Общая форма оператора:

```
switch (переменная_выбора)
{
    case const_1: операторы_1; break;
    ...
    case const_N: операторы_N; break;
    default: операторы_N+1;
}
```

переменная_выбора, **const_1**, ..., **const_N** – константа, переменная или выражение целого, символьного или логического типа.

При использовании оператора **switch** сначала анализируется *переменная_выбора* и проверяется, совпадает ли её значение со значением одной из констант **const_1**, ..., **const_N**. При совпадении выполняются операторы этого **case**. Конструкция **default** (может отсутствовать) выполняется, если результат выражения не совпал ни с одной из констант.

2.4. Пример выполнения работы

Написать программу вычисления выражения $s = \begin{cases} |\cos(x)| + \ln(y), & |xy| > 10 \\ e^{2x+y}, & 3 < |xy| \leq 10 \\ \sqrt{|x|} + 2\text{tg}(y), & \text{иначе} \end{cases}$

Предусмотреть вывод информации о выбранной ветви вычислений.

Код программы

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

int main()
{
    double x, y, s;
    cout << "\Vvedite x: ";
    cin >> x;
    cout << "\Vvedite y: ";
    cin >> y;

    double f_xy = fabs(x*y);
    if (f_xy > 10) { // |xy| > 10
        s = fabs(cos(x)) + log(y);
        cout << "1 vetv. Result = " << s << endl;
    }
    else if (f_xy > 3 && f_xy <= 10) // 3 < |xy| ≤ 10
        { s = exp(2*x+y);
        cout << "2 vetv. Result = " << s << endl;
        }
    else { // иначе
        s = sqrt(fabs(x)) + 2*tan(y);
        cout << "3 vetv. Result = " << s << endl;
    }

    return 0;
}
```

2.5. Индивидуальные задания

Составить программу вычисления выражения согласно указанному варианту. Предусмотреть вывод информации о выбранной ветви вычислений.

1. $s = \begin{cases} (x+y)^2 - \sqrt[3]{|x|}, & xy > 0 \\ (x+y)^2 + \sin(x), & xy < 0 \\ (x+y)^2 + y^3, & \text{иначе} \end{cases}$
2. $s = \begin{cases} \ln(x) + \sqrt[3]{|y|}, & x/y > 0 \\ \ln|x/y| \cdot (x+y)^3, & x/y < 0 \\ (x^2+y)^3, & \text{иначе} \end{cases}$
3. $s = \begin{cases} x^2 + \sqrt[3]{y} + \sin(y), & x-y = 0 \\ (x-y)^2 + \ln(|x|), & x-y > 0 \\ (y-x)^2 + \operatorname{tg}(y), & \text{иначе} \end{cases}$
4. $s = \begin{cases} \sqrt[3]{|x-y|} + \operatorname{tg}(x), & x > y \\ (y-x)^3 + \cos(x), & x < y \\ (y+x)^2 + x^3, & \text{иначе} \end{cases}$
5. $s = \begin{cases} y\sqrt{|x|} + 3\sin(x), & x > y \\ x\sqrt{|x|}, & x < y \\ \sqrt[3]{|x|} + x^3/y, & \text{иначе} \end{cases}$
6. $s = \begin{cases} e^{x-|y|}, & 0,5 < xy < 10 \\ \sqrt[3]{|x+y|}, & 0,1 < xy < 0,5 \\ 2x^2, & \text{иначе} \end{cases}$
7. $s = \begin{cases} e^{-x}, & 1 < xb < 10 \\ \sqrt[3]{|x+4y|}, & 12 < xb < 40 \\ y \cdot x^2, & \text{иначе} \end{cases}$
8. $s = \begin{cases} (x^2+y)^3, & x/y < 0 \\ \ln|x/y| + x/y, & x/y > 0 \\ \sqrt[3]{|\sin(y)|}, & \text{иначе} \end{cases}$
9. $s = \begin{cases} 2x^3 + 3y^2, & x > |y| \\ |x-y|, & 3 < x < |y| \\ \sqrt[3]{|x-y|}, & \text{иначе} \end{cases}$
10. $s = \begin{cases} \ln(|x|+|y|), & |xy| > 10 \\ e^{x+y}, & |xy| < 10 \\ \sqrt[3]{|x|} + y, & \text{иначе} \end{cases}$
11. $s = \begin{cases} \operatorname{tg}(x) + \frac{x}{\sqrt[3]{y}}, & xy > 0 \\ \ln|x^2 \cdot y|, & xy < 0 \\ x^3 + \sin^2(y), & \text{иначе} \end{cases}$
12. $s = \begin{cases} \operatorname{tg}(x) + x^2, & y > 2x \\ |x+y|^3, & y < 2x \\ \sqrt[3]{x} \cdot \sin(x), & \text{иначе} \end{cases}$
13. $s = \begin{cases} (x + \ln(|y|))^3, & x/y > 0 \\ 2/3 + \ln(|\sin(y)|), & x/y < 0 \\ \sqrt[3]{x^2} + y, & \text{иначе} \end{cases}$
14. $s = \begin{cases} \ln(x)^3, & x^3 > 0 \\ \operatorname{tg}(x^3) + y \cdot x, & x^3 < 0 \\ \sqrt[3]{|y^3 - x^2|}, & \text{иначе} \end{cases}$
15. $s = \begin{cases} (x^2 + y^3)/y, & x > 0 \\ \ln|x^3| + \cos(y), & x < 0 \\ \sqrt[3]{\sin^2(y)}, & \text{иначе} \end{cases}$

ЛАБОРАТОРНАЯ РАБОТА №7

ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ

3.1. Оператор цикла с параметром *for*

Общий вид оператора:

```
for (инициализирующее_выражение; условие;  
      инкрементирующее_выражение)  
{  
    тело цикла;  
}
```

Инициализирующее_выражение выполняется только один раз в начале выполнения цикла и, как правило, инициализирует счетчик цикла.

Условие содержит операцию отношения, которая выполняется в начале каждого цикла. Если условие равно **true (1)**, то цикл повторяется, иначе выполняется следующий за телом цикла оператор.

Инкрементирующее_выражение, как правило, предназначено для изменения значения счетчика цикла. Модификация счетчика происходит после каждого выполнения тела цикла.

3.2. Оператор цикла с предусловием *while*

Общий вид оператора:

```
while (условие)  
{  
    тело цикла;  
}
```

Операторы *тела цикла* повторяются до тех пор, пока *условие* истинно.

3.3. Оператор цикла с постусловием *do*

Общий вид оператора:

```
do {  
    тело цикла;  
}  
while (условие);
```

Операторы *тела цикла* повторяются до тех пор, пока *условие* истинно.

3.4. Операторы перехода

Оператор **break** прекращает выполнение ближайшего к нему цикла или оператора **switch**.

Оператор **continue** передает управление на проверку условия циклов **while** и **do while**, либо на *инкрементирующее_выражение* цикла **for**.

3.5. Отладка программы

Для поиска логических ошибок используется встроенный отладчик.

Для пошагового выполнения программы необходимо нажимать клавишу **F10**. При каждом нажатии выполняется текущая строка. Если необходимо пошагово проверить код вызываемой функции, то следует нажать **F11**. Для досрочного выхода из функции нажать **Shift+F11**. Если необходимо начать отладку с определенного места программы, то надо установить курсор в соответствующую строку программы и нажать **Ctrl+F10**.

Другим способом отладки является установка *точек прерывания* программы. Для этого надо поместить курсор в нужную строку и нажать **F9**. Точка прерывания обозначается красным кружком на специальном поле, расположенном слева от окна кода программы. Для удаления точки прерывания следует в необходимой строке повторно нажать **F9**. Количество точек прерывания в программе может быть любым.

Для выполнения программы до точки прерывания необходимо нажать **F5**. Для продолжения отладки применяется клавиша **F5** (выполнение программы до следующей точки прерывания) или используются клавиши для пошаговой отладки.

Желтая стрелка на поле слева от окна кода программы указывает на строку, которая будет выполнена на следующем шаге отладки.

Для контроля за значениями переменных удобно использовать следующий способ: подвести указатель мыши к интересующей переменной и задержать его на несколько секунд. На экране рядом с именем переменной появится окно, содержащее текущее значение этой переменной. Кроме этого, значения переменных будут отображаться в окнах, расположенных снизу. В левом нижнем окне отображаются значения последних использованных программой переменных. В правом нижнем окне (**Watch**) можно задать имена переменных, значения которых необходимо контролировать.

3.6. Пример выполнения работы

Написать программу вывода на экран таблицы значений функции $\sum_{k=0}^{20} \frac{2x^k}{\cos^n(x)}$ для x , изменяющегося от $a = 0,1$ до $b = 1$ с шагом $h = 0,1$.

Код программы

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

int main()
{
    double s, x, a, b, h;
    int k, i;

    cout << "Vvedite a, b, h, k:" << endl;
    cin >> a >> b >> h >> k; // Ввод значений: 0.1 1 0.1 20

    x = a;
```

```

do                                     // Начало цикла по x
{
  s = 2;                               // Начальное значение при k=0
  for (i = 1; i <= k; i++)             // Вычисление суммы  $\sum_{k=1}^{20} \frac{2x^k}{\cos^n(x)}$ 
    s += 2*pow(x, i) / pow(cos(x), i);
  cout << setw(15) << x << setw(15) << s << endl; // Вывод таблицы
  x += h;                               // Изменение значения x на величину шага h
}
while (x <= b + h/2);                 // Проверка условия продолжения цикла по x
cout << endl;                          // Переход на новую строку
return 0;
}

```

3.6. Индивидуальные задания

Вывести на экран таблицу значений функции $y(x)$ для x , изменяющегося от $a = 0,1$ до $b = 1,2$ с шагом $h = 0,1$.

$$1. \quad y = \sum_{n=1}^{20} \frac{x^{n-1}}{2n+1}.$$

$$2. \quad y = \sum_{n=0}^{20} \frac{(2x)^n}{n+1}.$$

$$3. \quad y = \sum_{n=1}^{20} \frac{x^{n-1}}{\sin(nx)}.$$

$$4. \quad y = \sum_{n=1}^{20} \frac{n^2+1}{n} \left(\frac{x}{2}\right)^n.$$

$$5. \quad y = \sum_{n=1}^{20} \frac{\cos\left(n \cdot \frac{\pi}{4}\right)}{n+1} x^n.$$

$$6. \quad y = \sum_{n=1}^{20} \frac{x^{2n-2}}{2n+1}.$$

$$7. \quad y = \sum_{n=0}^{20} \frac{x^{2n}}{\cos(nx)}.$$

$$8. \quad y = \sum_{n=1}^{20} \frac{2n^2+1}{2n} x^{2n-2}.$$

$$9. \quad y = \sum_{n=1}^{20} \frac{2n+1}{\sin(nx)} x^{n-1}.$$

$$10. \quad y = \sum_{n=0}^{20} \frac{\cos^n(x)}{2n+1}.$$

$$11. \quad y = \sum_{n=1}^{20} \frac{n \cdot x^{n-1}}{\sin(2n+x)}.$$

$$12. \quad y = \sum_{n=1}^{20} \frac{(1+x)^{n-1}}{|\sin^n(x)|}.$$

$$13. \quad y = \sum_{n=1}^{20} \frac{x^{2n-2}}{4\cos(nx^2)}.$$

$$14. \quad y = \sum_{n=1}^{20} \frac{n^2}{(2n+1)} x^{n-1}.$$

$$15. \quad y = \sum_{n=0}^{20} \frac{\cos\left(n \cdot \frac{\pi}{4}\right)}{n^2} x^n.$$

ЛАБОРАТОРНАЯ РАБОТА №8

ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ОДНОМЕРНЫХ МАССИВОВ

4.1. Одномерные статические массивы

Массив – это набор данных одинакового типа, расположенных в непрерывной области памяти таким образом, чтобы по индексу элемента можно было вычислить адрес его значения: $\text{адрес}(a[i]) = \text{адрес}(a[0]) + i * k$,

где k – количество байт, отводимых под элемент массива;

i – индекс элемента массива.

Для доступа к элементу массива необходимо указать его имя и индекс (порядковый номер элемента в массиве):

имя_массива [индекс]

В программе одномерный массив объявляется следующим образом:

тип имя_массива [размер];

Пример декларации массива:

```
int mas[4];
```

Индексы в массиве начинаются с **0**, т. е. массив, приведенный в примере, будет содержать следующие элементы: $mas[0]$, $mas[1]$, $mas[2]$ и $mas[3]$. Выход индекса за пределы массива не проверяется.

Пример 1. Упорядочить элементы массива по возрастанию их значений, т.е. для всех элементов массива должно выполняться условие: $a_i < a_{i+1}$.

```
...
for (k=1; k<n; k++)           // k – номер просмотра массива
    for (i=0; i<n-k; i++)     // Посмотр элементов массива
        if (a[i] > a[i+1])    // Сравнение элементов массива
            { temp=a[i];      // Перестановка элементов ai и ai+1,
              a[i]=a[i+1];    // если они стоят неправильно
              a[i+1]=temp;
            }
```

Пример 2. Удалить из одномерного массива все отрицательные элементы
Для решения данной задачи необходимо выполнить следующие действия:

```
...
for (i=0; i<n; i++)
    if (a[i]<0)                // Если найден отрицательный элемент, то
        {
            for (j=i+1; j<n; j++) // сдвинуть все элементы, стоящие
                a[j-1]=a[j];     // после удаляемого на одну позицию
            n--;                // Уменьшение размера массива
            i--;                // Возврат к предыдущему индексу
        }
```

Пример 3. Даны одномерные упорядоченные по возрастанию массивы: X размером n элементов и Y размером m элементов. Объединить элементы этих массивов в массив Z так, чтобы он оказался упорядоченным по возрастанию.

```

...
k=i=j=0;
while(i<n && j<m) {
    if (a[i]<b[j]) { c[k]=a[i];
                    i++;
                }
    else { c[k]=b[j];
           j++;
        }
    k++;
}
while(i<n) {
    c[k]=a[i];
    i++;
    k++;
}
while(j<m) {
    c[k]=b[j];
    j++;
    k++;
}

```

4.2. Пример выполнения работы

Составить программу поиска минимального и максимального элементов одномерного массива и их индексов.

```

#include <iostream.h>
#include <math.h>
int main()
{
    int a[10], i, n, min, imin, max, imax;
    cout << "Vvedite razmer: ";           // Ввод одномерного массива
    cin >> n;
    for (i=0; i<n; i++)
    {
        cout << "Vvedite a[" << i << "]= ";
        cin >> a[i];
    }
    cout << "Massiv a:" << endl;         // Вывод одномерного массива
    for (i=0; i<n; i++)
        cout << a[i] << " ";
}

```

```

cout << endl;
min=max=a[0];
imin=imax=0;
for (i=1; i<n; i++)
    if (a[i]<min) { min=a[i];
                    imin = i;
                }
            else
                if (a[i]>max) { max=a[i];
                                imax = i;
                            }
cout << "Max = " << max << "   i=" << imax << endl;
cout << "Min = " << min << "   i=" << imin << endl;
return 0;
}

```

4.3. Индивидуальные задания

Ввести одномерный статический массив из k чисел. Выполнить в соответствии с номером варианта индивидуальное задание и вывести на экран исходные данные и полученный результат.

1. Преобразовать массив следующим образом: все отрицательные элементы массива перенести в начало, сохранив исходное взаимное расположение, как среди отрицательных, так и среди остальных элементов массива.
2. Расположить элементы массива в обратном порядке.
3. Найти и поменять местами элементы, имеющие минимальное и максимальное значения в массиве.
4. Определить, упорядочены ли элементы массива по убыванию.
5. Вывести все неповторяющиеся элементы массива.
6. Сдвинуть элементы массива циклически на n позиций *влево*.
7. Сдвинуть элементы массива циклически на n позиций *вправо*.
8. Удалить минимальный и максимальный элементы массива.
9. Сформировать два новых массива: в первый записать отрицательные элементы исходного массива, во второй – все остальные.
10. Определить, симметричен ли массив, т.е. читается ли он одинаково слева направо и справа налево.
11. Найти количество элементов массива, отличающихся от среднего значения элементов массива не более чем на 3.
12. Определить количество инверсий в массиве (таких пар элементов, в которых большее значение находится слева от меньшего).
13. Определить количество элементов, значение которых больше среднего значения всех элементов массива.
14. Удалить элементы, значение которых меньше среднего значения всех элементов массива.
15. Удалить из массива повторяющиеся элементы.

Список литературы

1. Навроцкий А.А. Основы алгоритмизации и программирования в среде Visual C++: лаб. Практикум по курсу “Основы алгоритмизации и программирования” для студ. 1-2-го курсов.-Минск: БГУИР, 2008.-48с.: ил.
2. Джесс Либерты, Дэвид Хорват. Освой самостоятельно C++ за 24 часа. 4-е издание, Пер.с.анг.-М.: Издательский дом «Вильямс», 2007. – 448с.:ил.-Парал.тит.англ.
3. Аширова А., Сапаев У., Алламов О. Методическое указание для выполнении лабораторных работ по дисциплине информатика. Ургенч-2009.-46с.