

Томский государственный университет систем
управления и радиоэлектроники (ТУСУР)

Факультет вычислительных систем (ФВС)
Кафедра Моделирования и системного анализа (МиСА)

Е.В. Истигечева, А.В. Мельников

Программирование и основы алгоритмизации

Лабораторный практикум

Томск 2015

Е.В. Истигчева, А.В. Мельников

Программирование и основы алгоритмизации / Методические указания по лабораторным работам – Томск: Томский государственный университет систем управления и радиоэлектроники. Факультет вычислительных систем, 2015. – 31 с.

© Е.В. Истигчева, А.В. Мельников, 2015.

© Факультет вычислительных систем, кафедра моделирования и системного анализа, 2015.

СОДЕРЖАНИЕ

Лабораторная работа 1. Знакомство со средой разработки Borland Turbo C.....	4
Лабораторная работа 2. Программная реализация линейных алгоритмов	6
Лабораторная работа 3. Программная реализация разветвленных алгоритмов.....	11
Лабораторная работа 4. Программная реализация циклических алгоритмов	15
Лабораторная работа 5. Программная реализация алгоритмов с массивами	19
Лабораторная работа 6. Программная реализация алгоритмов с использованием функций и указателей.....	23
Лабораторная работа 7. Программная реализация строковых типов данных.....	27

Данный практикум призван оказать помощь преподавателям в подборе задач для проведения лабораторного практикума, помочь учащимся ВУЗа в получении навыков разработки алгоритмов и программ. Автор старался дать минимум теоретических сведений об алгоритмизации и программировании на Си (этот материал дается на лекциях), а больше внимания уделить решению примеров. В качестве примеров выбирались наиболее типичные или интересные. Решение каждого примера сопровождается краткими пояснениями и комментариями. Как правило, приводится несколько вариантов решения каждого примера.

Лабораторная работа 1. Знакомство со средой разработки Borland Turbo C

Цель: освоение среды разработки Borland Turbo C.

Что нужно знать для выполнения работы:

Общие сведения

Горячие клавиши

Для выполнения часто используемых операций используются так называемые горячие клавиши и комбинации клавиш.

Клавиши общего назначения:

F1 – вызвать подсказку;

F2 – сохранить файл;

F3 – открыть файл;

F4 – выполнить программу до строки, на которой установлен курсор;

F5 – развернуть/свернуть активное окно;

F6 – перейти к следующему окну;

F7 – выполнение программы в режиме отладки с трассировкой;

F8 – выполнение программы в режиме отладки;

F9 – откомпилировать программу и создать исполняемый файл;

F10 – перейти в меню.

Клавиши активации меню:

Alt+пробел – системное меню;

Alt+C – меню компиляции;

Alt+D – меню отладки;

Alt+E – меню редактирования;

Alt+F – меню операций с файлами;

Alt+H – меню подсказки;

Alt+O – меню настроек;

Alt+P – меню проекта;

Alt+R – меню запуска;

Alt+S – меню поиска;

Alt+W – меню выбора окна;

Alt+X – выход из системы.

Клавиши режима редактирования:

Ctrl+Del – удалить выделенный текст;

Ctrl+Ins – скопировать выделенный текст в буфер;

Shift+Del – скопировать выделенный текст в буфер и удалить его;

Shift+Ins – вставить текст из буфера в файл;

Alt+Bkspc – возвращает текст в предыдущее состояние.

Клавиши управления окнами:

Alt+F3 – закрыть активное окно;

Alt+F4 – открыть окно проверки;

Alt+F5 – отобразить экран пользователя.

Клавиши активации режима подсказки:

F1 – выдать инструкцию по использованию подсказки;

Shift+F1 – вызвать тематический указатель по системе подсказки.

Ctrl+F1 – вызывать контекстную подсказку.

Клавиши режима выполнения и отладки:

Alt+F4 – открыть окно проверки;

Alt+F7 – возврат к предыдущей ошибке;

Alt+F8 – перейти к следующей ошибке;

Alt+F9 – создать объектный файл (компиляция без компоновки);
Ctrl+F2 – реинициализация выполняющейся программы;
Ctrl+F3 – вывести на экран окно со стеком вызовов функций;
Ctrl+F4 – вычислить выражение;
Ctrl+F7 – добавить просматриваемое выражение;
Ctrl+F8 – установить или отменить точку останова;
Ctrl+F9 – выполнить программу.

Контрольные вопросы:

1. Перечислить клавиши общего назначения?
2. Перечислить клавиши активации меню?
3. Перечислить клавиши режима редактирования?
4. Перечислить клавиши управления окнами?
5. Перечислить клавиши активации режима подсказки?
6. Перечислить клавиши режима выполнения и отладки?

Задания.

1. Запустить среду разработки Borland Turbo C;
2. Написать в окно программы следующий код:

```
#include <stdio.h>
main(){
printf(“Здравствуйте!\n”);
}
```
3. Откомпилировать и запустить программу, используя горячие клавиши;
4. Изучить, все горячие клавиши перечисленные выше.

Лабораторная работа 2. Программная реализация линейных алгоритмов

Цель: изучить операторы ввода и вывода, форматы, используемые в этих операторах.

Что нужно знать для выполнения работы:

Общие сведения

Для ввода данных используется функция:

```
scanf(«управляющая строка», аргумент1, аргумент2,...);
```

Если в качестве аргумента используется переменная, то перед ее именем записывается символ &.

Для вывода данных используется функция:

```
printf("управляющая строка", аргумент1, аргумент2, ... );
```

Управляющая строка содержит объекты трех типов: обычные символы, которые просто выводятся на экран дисплея, спецификации преобразования, каждая из которых вызывает вывод на экран значения очередного аргумента из последующего списка и управляющие символы-константы.

Примеры выполнения задания лабораторной работы:

Пример1. Напишите программу вычисления суммы четырех чисел.

Этапы решения задачи:

- Будем считывать четыре числа a,b,c,d. Далее вычислим сумму этих чисел и выведем результат на экран.
- Составим блок - схему программы.

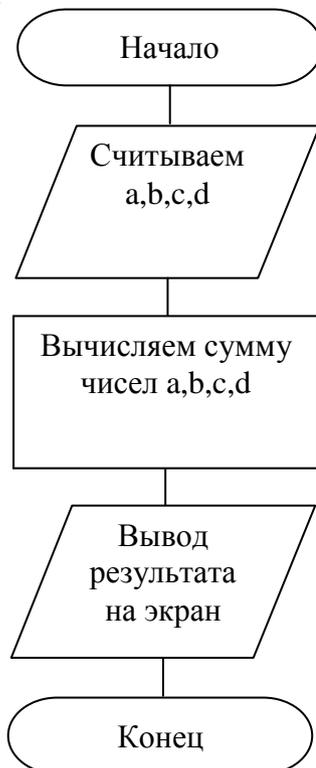


Рисунок 1 – Блок-схема

Листинг программы:

```
#include <stdio.h>
void main()
{
int a,b,c,d,p;
scanf("%d%d%d%d",&a,&b,&c,&d);
p=a+b+c+d;
printf("%d\n",p);
}
// Вызываем библиотеку stdio.h
// Объявляем главную функцию программы
// Начало тела программы
// Объявляем переменные
// Считываем a,b,c,d
// Вычисляем сумму чисел a,b,c,d
// Выводим результат на экран
// Конец тела программы
```

Контрольные вопросы:

7. Функция ввода данных?
8. Функция вывода данных?
9. Что в себе содержит управляющая строка?
10. Что такое спецификация преобразования?
11. Назовите управляющие символы-константы?

Задания

Написать программу и составить блок-схему к ней согласно данному варианту.

Вариант 1.

Напишите программу вычисления суммы и произведения двух чисел.

Вариант 2.

Напишите программу нахождения гипотенузы и площади прямоугольного треугольника по двум катетам.

Вариант 3.

Напишите программу вычисления среднего арифметического двух чисел.

Вариант 4.

Напишите программу вычисления суммы четырёх слагаемых.

Вариант 5.

Напишите программу вычисления периметра и площади прямоугольного треугольника по двум катетам.

Вариант 6.

Программа запрашивает у пользователя a, b, c , вычислить значение выражения $a^2 - (b - c)^2 + b^2 + 1$.

Вариант 7.

Программа запрашивает три числа a, b, c . Вывести на экран значения выражения $a + \sqrt{bc} + 5c$

Вариант 8.

Программа запрашивает три числа a, b, c . Вывести на экран значения выражения $a + b + 2\sqrt{c}$.

Вариант 9.

Программа запрашивает три числа a, b, c . Вывести на экран значения выражения $a^2 + b + 1$.

Вариант 10.

Найти площадь поверхности параллелепипеда по формуле:

$$S = 2 (A B + B C + A C).$$

Вариант 11.

Найти площадь поверхности и объем куба по формулам:

$$S = 6 A^2 \quad V = A^3$$

Вариант 12.

Найти объем ртути в сосуде по формуле:

$$V = \frac{m}{\rho};$$

Вариант 13.

Вычислить значение функции

$$y = \frac{\sqrt{5x^6} + 2x}{x + 8x}$$

Вариант 14.

На обучение одного ученика государство тратит в год 350 тыся рублей. Произвести расчёт затрат государства на обучение одного класса и всей школы, если в классах по 28 учащихся, а в школе 26 классов.

Вариант 15.

Вычислить длину вектора по формуле:

$$D = \sqrt{A^2 + B^2 + C^2}$$

Вариант 16.

Вычислить вес человека, используя формулу:

$$\text{вес} = \frac{\text{рост} \times \text{окружность грудной клатки}}{240}$$

Вариант 17.

Заработок рабочих на фабрике составил 1854 условные единицы. Этот заработок нужно разделить поровну между рабочими, количество которых N. Найти заработок одного рабочего.

Вариант 18.

Магазин продаёт автомашины по цене 25 тысяч условных единиц за каждую. Найти общую выручку от продажи, если количество машин – 45 штук.

Вариант 19.

На каждом из пяти одинаковых по длине участков дороги автомобиль шел с различной скоростью. Определить среднее время прохождения каждого из участков дороги.

Вариант 20.

Дана последовательность из шести чисел (они вводятся с клавиатуры). Найти сумму этих чисел и их среднее арифметическое. Вывести на экран полученные величины и сами числа на разных строках.

Вариант 21.

Вычислить стоимость дачного гарнитура, состоящего из четырёх стульев, двух кресел и стола. Стоимость каждого предмета вводится с клавиатуры

Вариант 22.

Рассчитать семейный бюджет на одного человека. Семья состоит из четырёх человек: отца, матери, сына и дочери. Отец и мать работают (т.е. приносят в семью деньги), сын и дочь – школьники (денег в семью не приносят). Зарплаты отца и матери вводятся с клавиатуры.

Вариант 23.

Найти площадь поверхности цилиндра и его объём по формулам:

$$S = \frac{3,14 \cdot D^2}{2} + 3,14 \cdot D \cdot H$$

$$V = \frac{3,14 \cdot D^2}{4} \cdot H$$

Вариант 24.

Камень бросили в колодец и через T секунд послышался плеск воды. На какой глубине от сруба находится вода в колодце?

Высота определяется по формуле:

$$H = \frac{gT^2}{2}$$

Вариант 25.

Найти массу круглого цилиндрического стержня $d=3,5$ см и длиной $l=11,56$ см, сделанного из алюминия, плотность которого $\rho=2,7$ г/см. Формулы для расчёта:

$$M = \nu \rho$$

$$\nu = ld$$

Лабораторная работа 3. Программная реализация разветвленных алгоритмов

Цель: Формирование первичных умений в разработке разветвляющихся программ

Что нужно знать для выполнения работы

1. Символы, используемые в схемах программ и правила их применения.
2. Структура программы.
3. Оператор присваивания, математические и логические операции, их приоритет.
4. Операторы ввода-вывода `scanf()`, `printf()`.
6. Условный оператор `if (условие) оператор_1 else оператор_2`.

Пример выполнения задания лабораторной работы

Условие задачи. Даны два числа a и b . Получите число $\max a, b$, равное наибольшему из исходных. Если исходные числа равны, то $\max a, b$ положить равным любому из них.

Решение. Для решения задачи нужно ввести значения переменных a и b , сравнить их значения и в зависимости от результата сравнения $\max a, b$ положить равным a или b .

Обратите внимание на то, что вместе с результатом выводятся и исходные данные. Это позволит упростить тестирование и повысить наглядность вывода. Во втором варианте решения данной задачи, не сравнивая, присваиваем переменной-результату $\max a, b$ первое «попавшееся» значение, а затем сравниваем это значение со вторым числом.

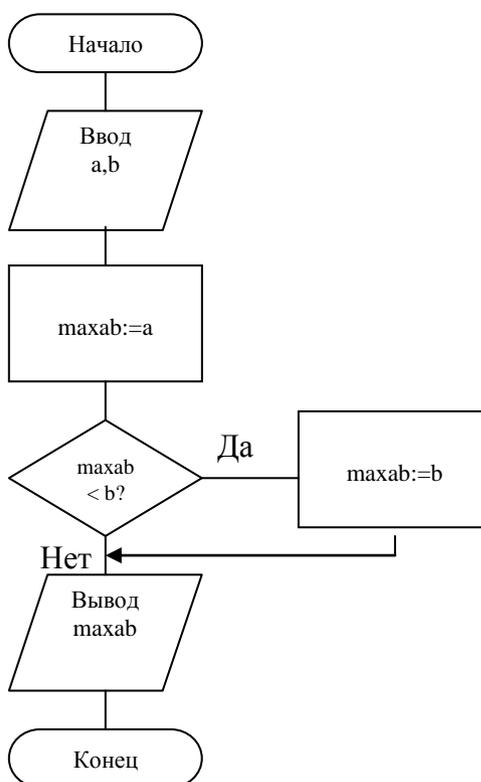


Рисунок 2 - Схема алгоритма программы

Контрольные вопросы:

1. Функция ввода-вывода данных?
2. Структура программы?
3. Оператор условия if-else?

Листинг программы

```
#include <stdio.h>                                     // Вызываем библиотеку stdio.h
void main()                                             // Объявляем главную функцию программы
{                                                       // Начало тела программы
  int a,b,maxab;                                       // Объявляем переменные
  scanf("%d",&a,&b);                                    // Считываем a,b
  maxab=a;                                             // Переменной maxab присваиваем значение a
  if(maxab<b)                                          // Если maxab меньше b, то
  maxab=b;                                             // Переменной maxab присваиваем значение b
  printf("%d\n",maxab);                                // Выводим результат на экран
}                                                       // Конец тела программы
```

Задание:

Написать программу и нарисовать блок-схему к ней согласно данному варианту.

Вариант 1.

Программа вводит два числа a и b , если $a > b$, то программа вводит третье число c и находит сумму $b+c$. Если $a=b$, то программа печатает слово «Конец». Если $a < b$, то программа вводит третье число c , находит сумму $a+b+c$, выводит на экран сумму $a+b+c$ и слова «Новый год!».

Вариант 2.

Программа вводит два числа a и b . Если $a > b$, то программа вводит третье число c , находит сумму $a+b+c$ и выводит ее на экран. Если $a < b$, то находит разность $a-b$, если разность четная, то вводит третье число c и выводит его на экран. Во всех остальных случаях программа печатает слово «Конец».

Вариант 3.

Перераспределить значения переменных X и Y так, чтобы в X оказалось меньшее из этих значений, а в Y — большее.

Вариант 4.

Программа вводит два числа a и b , если $a > b$, то программа вводит третье число c . Находит сумму $a+b+c$ и выводит ее на экран. Если $a < b$, то находит их разность $b-c$. Если разность нечетная, то ввести третье число и вывести на экран $a+b$. Во всех остальных случаях напечатать слово "Конец".

Вариант 5.

Программа просит пользователя ввести a и b , если $a > b > 5$, то программа выводит на экран слово "осень", если $a=b$, то выводит значение выражения

$\frac{a+b}{a^2+b^2+1}$, во всех остальных случаях ввести c и вывести значение выражения $\sqrt{c} * \sin(c^2 + a^2b) - \frac{1}{7}$.

Вариант 6.

Программа просит пользователя ввести a и b и, если $a < b$ то просит ввести $c > 0$ и находит значение выражения $2a^2b^3c - \frac{1}{3}\sqrt{c}$, в остальных случаях выводит на экран «Конец работы».

Вариант 7.

Даны действительные числа x, y . Если x, y отрицательны, то каждое значение заменить его модулем; если отрицательное только одно из них, то оба значения увеличить на 0.5; если оба значения не отрицательны и ни одно из них не принадлежит отрезку $[0.5, 2.0]$, то оба значения уменьшить в 10 раз; в остальных случаях x, y оставить без изменения.

Вариант 8.

Программа вводит 2 числа a и b . Если $a=b$, то программа выводит на экран значение выражения $\sin(a^2+b^2)$, если $a < b$, то – выводит на экран слово 'осень'. Если $a < 0$ и $b < 0$, то вводит третье число c и выводит на экран слова «конец программы» и значение выражения $a+b-c$.

Вариант 9.

Дано число: если оно кратно 4, то печатать «нужное число»; если оно кратно 3 и 5, то также печатать «нужное число», если число четное, то печатать "постороннее число", если четное число не кратно 6, то вывести на экран сообщение « четное число не кратно 6».

Вариант 10.

Программа вводит два числа a и b , если $a > b$, то программа вводит третье число c . Если $a=b$, то программа печатает слово «Конец». Если $a < b$, то программа вводит третье число c , если $c < 8$, то программа находит значение выражения $\ln(a^2+b^2)+\exp(c)$, выводит его и слова «Новый год» на экран.

Вариант 11.

Программа вводит целое число i , если оно > 0 и чётное, программа вводит второе число. Если оба числа чётные, то программа выводит на экран сумму 2-х чисел, во всех чисел остальных случаях программа выводит на экран сообщение «конец работы».

Вариант 12.

Программа вводит три числа a, b, c . Если $a-b > c$, то вывести слово «зима», в остальных случаях вывести на экран «конец работы».

Вариант 13.

Программа вводит два числа a и b , если $a > b$, то программа вводит c и выводит на экран значение выражения $a^7 + \frac{ab}{19} + c^{\frac{1}{5}}$, иначе - выводит разность $a-b$ и слова «конец работы».

Вариант 14.

Программа вводит целое число a , если $a > 5$, то вывести на экран выражение $a+b$, в противном случае вывести на экран слова «конец работы!».

Вариант 15.

Программа вводит два числа a и b , если $a=b$ то вывести на экран слова «моя любимая футбольная команда», если $a > b$, то ввести c и вывести на экран значение выражения $a^2 + \frac{b}{a^2 + 1} + \sqrt{c}$, если $a < b$, то вывести на экран слова «конец работы».

Вариант 16.

Программа вводит два числа a и b , если $a=b$, то вывести на экран слова «скоро Новый Год!», в противном случае ввести третье число c и вывести на экран 3 строки :

-значение выражения $a+b+c$

-значение выражения $a^2 + b^2$

-«До свиданья»

Вариант 17.

Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу $(1, 3)$.

Вариант 18.

Программа вводит 2 числа a и b . Если $a < b$, то программа вводит третье число c и выводит значение выражения $a+b+c^2$. Во всех остальных случаях программа выводит на экран слова «конец программы».

Вариант 19.

Ввести два числа. Меньшее заменить суммой, а большее – удвоенным произведением.

Вариант 20.

Написать программу для вывода дней недели по их номеру:

1 - понедельник

2 - вторник

...

7 - воскресенье

С клавиатуры вводится номер дня недели.

Вариант 21.

Программа вводит два числа a и b , если $a > b$, то программа вводит третье число c . Если $a=b$, то программа печатает слово "Конец". Если, $a < b$, то программа вводит третье число c , находит разность $a-b-c$, выводит на экран разность и слова "Новый год".

Вариант 22.

Даны два действительных числа. Выбрать из них те, которые принадлежат интервалу $(5, 7)$.

Вариант 23.

Даны три переменные: X, Y, Z . Если их значения упорядочены по убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.

Вариант 24.

Найти минимум из трех чисел.

Вариант 25.

Найти максимум из двух чисел.

Лабораторная работа 4. Программная реализация циклических алгоритмов

Цель: Формирование первичных умений в разработке простых циклических программ с заданным числом повторений.

Что нужно знать для выполнения работы

1. Структура цикла, назначение и порядок следования его отдельных частей.
2. Структуры: «цикл-пока» и «цикл-повторить».
3. Операторы цикла «while()», «for ()».

Пример выполнения задания лабораторной работы

Условие задачи: Составьте программу, находящую сумму значений функции $\sin(x)$ для x , изменяющегося от 0 до 10π с шагом $\pi/3$.

Схема алгоритма

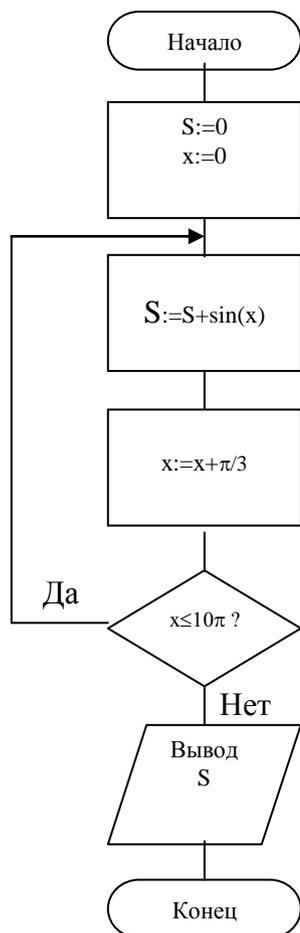


Рисунок 3 - Схема алгоритма программы

Контрольные вопросы:

1. Структура цикла «цикл-пока»?
2. Структура цикла «цикл- повторить»?
3. Оператор цикла while()?
4. Оператор цикла for()?

Листинг программы

```
#include <stdio.h>           // Вызываем библиотеку stdio.h
#include <math.h>            // Вызываем библиотеку math.h
void main()                 // Объявляем главную функцию программы
{                             // Начало тела программы
    float x,S;              // Объявляем переменные
    S=0;                    // Обнуляем переменную S
    for(x=0;x<10*pi;x=x+pi/3){ // Задаем цикл по x
        S=S+sin(x);        // Находим сумму значений функции sin(x)
    }                       // Выводим результат на экран
    printf(“%F\n”,S);      // Конец тела программы
}
```

Задание:

Написать программу и нарисовать блок-схему к ней согласно данному варианту.

Вариант 1.

Дано n чисел. Найти среднее арифметическое этих чисел.

Вариант 2.

Программа запрашивает, сколько чисел хочет сложить пользователь. Программа просит ввести эти числа, а затем выводит полученную сумму.

Вариант 3.

Программа запрашивает количество чисел и вводит числа в соответствии с этим количеством. Программа определяет количество отрицательных чисел, сумму всех положительных.

Вариант 4.

Программа запрашивает количество чисел, вводит целые числа в соответствии с заявленным количеством, определяет, сколько чисел больше 15. Результат вывести на экран.

Вариант 5.

Программа запрашивает количество чисел, затем вводит целые числа в соответствии с указанным количеством. Программа находит сумму отрицательных и сумму нечетных чисел.

Вариант 6.

Программа выводит на экран в столбик числа от 0 до 100.

Вариант 7.

Программа вводит натуральное n , затем вводит числа в соответствии с этим количеством. Найти сумму тех чисел, порядковый номер которых кратен 3.

Вариант 8.

Программа вводит натуральное n , затем вводит числа в соответствии с этим количеством. Найти количество тех чисел, которые больше 10

Вариант 9.

Программа запрашивает количество чисел, затем вводит целые числа в соответствии с указанным количеством, находит среднее арифметическое и сумму отрицательных чисел, которые < -20 .

Вариант 10.

Программа запрашивает количество чисел, затем вводит целые числа в соответствии с указанным количеством. Определяет количество четных чисел.

Вариант 11.

Программа запрашивает количество чисел, затем вводит целые числа в соответствии с указанным количеством. Программа находит сумму отрицательных чисел и сумму нечетных. Программа выводит ту сумму, которая больше.

Вариант 12.

Программа запрашивает количество и вводит целые числа в соответствии с этим количеством. Программа находит сумму всех положительных чисел, количество отрицательных чисел, а так же среднее арифметическое.

Вариант 13.

Дано n целых чисел. Найти максимум среди этих чисел.

Вариант 14.

Дано n целых чисел. Найти минимум среди этих чисел.

Вариант 15.

Программа запрашивает количество чисел, затем вводит целые числа в соответствии с указанным количеством. Программа находит сумму отрицательных, сумму нечетных. Программа выводит ту сумму, которая меньше.

Вариант 16.

Программа запрашивает у пользователя количество чисел, вводит целые числа в соответствии с заданным количеством. Найти количество нечетных чисел.

Вариант 17.

Дано N целых чисел. Найти минимум среди положительных чисел и максимум среди отрицательных.

Вариант 18.

Программа запрашивает у пользователя количество чисел, вводит целые числа в соответствии с заданным количеством. Если числа больше 50, то посчитать их сумму, если меньше, то определить количество таких чисел.

Вариант 19.

Напишите программу, которая из N чисел определяет максимум и среднее арифметическое.

Вариант 20.

Дано 100 целых чисел. Вычислить разность между максимальным и минимальным элементами.

Вариант 21.

Программа запрашивает у пользователя количество чисел, вводит целые числа в соответствии с заданным количеством. Найти максимальный элемент среди чисел, введенных под номерами с 5 по 10.

Вариант 22.

Программа запрашивает у пользователя количество чисел, вводит вещественные числа. Найти максимальный и минимальный элементы и вывести их произведение.

Вариант 23.

Программа запрашивает у пользователя количество чисел, вводит целые числа в соответствии с заданным количеством. Программа определяет количество отрицательных целых чисел, кратных 2.

Вариант 24.

Программа запрашивает у пользователя количество чисел. Программа вводит числа в соответствии с заданным количеством и выводит на экран количество таких чисел, которые кратны числу, заданному пользователем.

Вариант 25.

Программа запрашивает у пользователя количество чисел, вводит целые числа в соответствии с заданным количеством. Найти минимальный элемент среди чисел, введенных под номерами с 7 по 15.

Лабораторная работа 5. Программная реализация алгоритмов с массивами

Цель: Формирование практических умений и навыков составления блок-схем и записи на языке программирования Си алгоритмов заполнения, обработки, и сортировки массивов.

Что нужно знать для выполнения работы

Массив – это тип данных, представляющий собой ограниченный набор упорядоченных элементов одного и того же типа, имеющих одно и то же имя.

Как и другие переменные, массив должен быть объявлен. Существует несколько способов объявления массива.

Общая форма объявления одномерного массива имеет следующий вид:

тип имя [размер]

где тип – базовый тип элемента массива;

имя – идентификатор массива;

размер – количество элементов в массиве.

Доступ к элементу массива осуществляется с помощью имени массива и индекса. Индекс элемента массива помещается в квадратных скобках после имени. Нижнее значение индекса всегда ноль.

Таким образом, элементами массива, состоящего из N элементов, являются переменные с индексами

$a[0], a[1], \dots, a[N-1]$.

В качестве N в описании должна стоять целая положительная константа.

Контрольные вопросы:

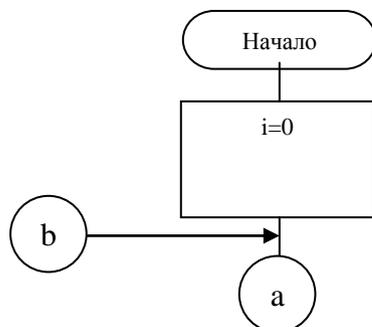
1. Что такое массив?
2. Структура массива?
3. Виды массивов?
4. Что такое индекс?

Пример выполнения задания лабораторной работы

$$S = \sum_{i=1}^N x_i,$$

Условие задачи: Найти $N = 15$

Схема алгоритма



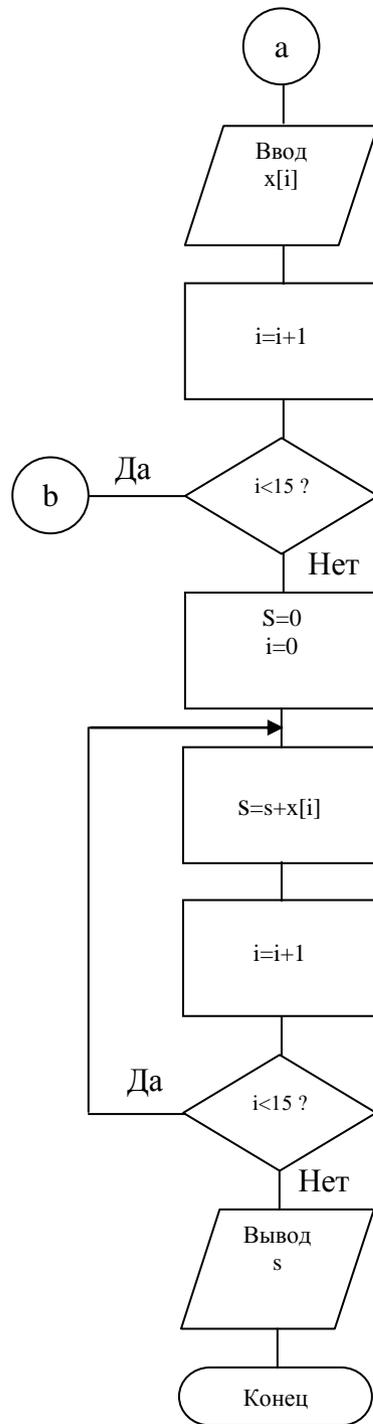


Рисунок 4 - Схема алгоритма программы

Листинг программы:

```
# include <stdio.h>
void main()
{
float x[15],s;
int i;
for (i=0;i<15;i++)
scanf("%f",&x[i]);
s=0;
for (i=0;i<15;i++)
s=s+x[i];
printf("\n S=%f",s);
}
```

\\ Вызываем библиотеку stdio.h
\\ Объявляем главную функцию программы
\\ Начало тела программы
\\ Объявляем переменные

\\ Задаем цикл по i
\\ Считываем массив x
\\ Обнуляем переменную s
\\ Задаем цикл по i
\\ Находим сумму элементов массива x
\\ Выводим результат на экран
\\ Конец тела программы

Задание:

Написать программу и нарисовать блок-схему к ней согласно данному варианту.

Вариант 1.

Программа заполняет одномерный массив из 10 целых чисел числами, считанными с клавиатуры. Программа определяет сумму положительных чисел и среднее арифметическое всех чисел массива.

Вариант 2.

Программа заполняет массив из 10 целых чисел случайными числами в диапазоне от -3 до 8. Программа находит максимальное число.

Вариант 3.

Одномерный массив чисел X[10] вывести пары отрицательных чисел, стоящих рядом.

Вариант 4.

Программа сортирует массив из 5 целых чисел «пузырьком», массив из 10 - вставкой и сливает их в массив из 15.

Вариант 5.

Одномерный массив из 10-и целых чисел заполнить с клавиатуры, определить сумму тех чисел, которые >5.

Вариант 6.

Дан одномерный массив из вещественных чисел. Ввести число c и найти значение

выражения $c + \frac{c}{a_1} + \frac{c}{a_2} + \dots + \frac{c}{a_n}$, где a – элементы массива.

Вариант 7.

Напечатать массив из 10 вещественных элементов в обратном порядке.

Вариант 8.

Дан массив целых чисел. Найти максимальный элемент массива и его порядковый номер.

Вариант 9.

Дан массив целых чисел. Найти минимальный элемент массива и его порядковый номер.

Вариант 10.

В массиве целых чисел поменять местами максимальный и минимальный элементы.

Вариант 11.

Упорядочить массив целых чисел по возрастанию.

Вариант 12.

Упорядочить массив целых чисел по убыванию.

Вариант 13.

Дан массив целых чисел. Найти сумму элементов с четными номерами и произведение элементов с нечетными номерами.

Вариант 14.

В массиве действительных чисел все нулевые элементы заменить на среднее арифметическое всех элементов массива.

Вариант 15.

Дан массив целых чисел. Переписать все положительные элементы во второй массив, а остальные - в третий.

Вариант 16.

Дан массив целых чисел. Найти максимальный элемент массива и количество таких элементов в массиве

Вариант 17.

Переставить в одномерном массиве второй элемент и максимальный.

Вариант 18.

Дан массив целых чисел. Найти минимальный элемент массива и количество таких элементов в массиве

Вариант 19.

Дан одномерный массив из 8 элементов. Заменить все элементы массива меньше 15 их удвоенными значениями. Вывести на экран монитора преобразованный массив.

Вариант 20.

Дан одномерный массив целого типа. Получить другой массив, состоящий только из нечетных чисел исходного массива или сообщить, что таких чисел нет. Полученный массив вывести в порядке убывания элементов.

Вариант 21.

Дан одномерный массив действительного типа. Получить другой массив, состоящий только из неотрицательных чисел исходного массива или сообщить, что таких чисел нет. Полученный массив вывести в порядке возрастания элементов.

Вариант 22.

Дан одномерный массив из 8 элементов. Элементам массива меньше 10 присвоить нулевые значения, а элементам больше 20 присвоить 1. Вывести на экран монитора первоначальный и преобразованный массивы в строку.

Вариант 23.

Дан одномерный массив целого типа. Получить другой массив, состоящий только из четных чисел исходного массива, меньше 10, или сообщить, что таких чисел нет. Полученный массив вывести в порядке возрастания элементов.

Вариант 24.

Дан массив целых чисел. Найти сумму элементов массива с четными номерами.

Вариант 25.

Дан массив целых чисел. Найти произведение элементов массива с нечетными номерами.

Лабораторная работа 6. Программная реализация алгоритмов с использованием функций и указателей

Цель: Научится использовать функции для выполнения однотипных действий над различными данными, правильно задавать параметры функций, передавать указатели на функции в качестве параметров.

Что нужно знать для выполнения работы

Часто в программе требуется повторить определенную последовательность операторов в разных частях программы. Для того, чтобы описывать эту последовательность один раз, а применять многократно, в языках программирования применяются подпрограммы. Подпрограмма - автономная часть программы, выполняющая определенный алгоритм и допускающая обращение к ней из различных частей общей программы.

В языке Си существует один вид подпрограмм - функции. Каждая программа в своем составе должна иметь главную функцию `main()`, служащую точкой входа в программу. Кроме функции `main()`, в программу может входить произвольное число функций, выполнение которых инициализируется либо прямо, либо вызовами из функции `main()`. Каждая функция по отношению к другой является внешней. Для того, чтобы функция была доступной, необходимо, чтобы до ее вызова о ней было известно компилятору. Форма записи функции: **<тип > <имя функции>(<формальные параметры>){ <тело функции > }**

Если тип возвращаемого функцией значения не указан, то подразумевается `int`. Если с именем функции не связан результат, то нужно указать тип функции `void`. Параметры, записываемые в обращении к функции, называются фактическими; параметры, указанные в описании функции - формальными. Фактические параметры должны соответствовать формальным по количеству, порядку следования и типу. Объекты, объявленные вне функции, действуют в любой функции и называются глобальными. Объекты, объявленные в функции, действуют только в ней и называются локальными. В теле функции обычно присутствует оператор `return <выражение>`, определяющий возвращаемое функцией значение.

Все параметры функции, кроме массивов, передаются по значению, т.е. внутри функции создаются локальные копии параметров. Если необходимо передать саму переменную, а не её копию, то в функцию передаётся адрес этой переменной. Таким образом, через параметры можно передавать результат выполнения функции. То есть, параметры, с помощью которых результаты должны передаваться из функции в точку вызова, описываются как указатели. Вызов функции может быть оформлен в виде оператора, если с именем функции не связано возвращаемое значение, или в виде выражения, если возвращаемое значение связано с именем функции.

Прототип функции может указываться до вызова функции вместо описания функции для того, чтобы компилятор мог выполнить проверку соответствия типов аргументов и параметров. Прототип функции по форме такой же, как и заголовок функции. В конце него ставится «;».

Функции можно подключать с помощью директивы `#include <имя файла>`. Такие файлы с функциями удобно использовать в диалоговых программах с пользовательским меню, позволяющих выбрать один из режимов.

Контрольные вопросы:

1. Структура функций?

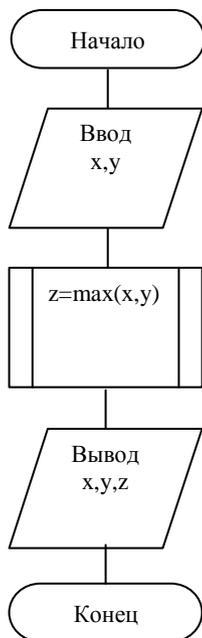
2. Оператор Return?
3. Локальные и глобальные переменные?
4. Перегрузка функций?

Пример выполнения задания лабораторной работы

Условие задачи: Найти максимальное значение среди 2-х чисел.

Схема алгоритма

Функция: main()



Функция: max()

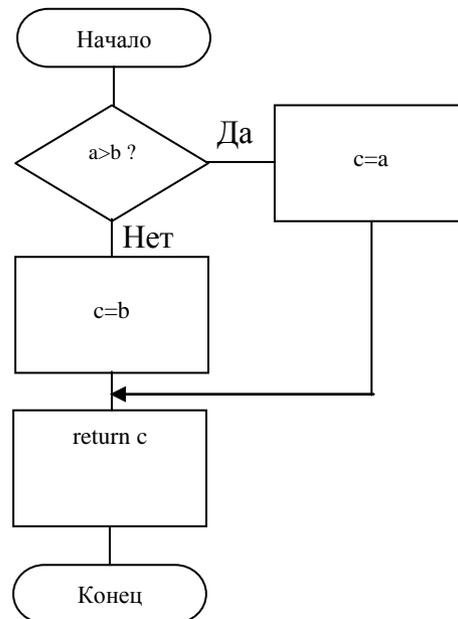


Рисунок 5 - Схема алгоритма программы

Листинг программы:

```
#include <stdio.h> // Вызываем библиотеку stdio.h
int max(int,int); // Прототип функции
void main() // Объявляем главную функцию программы
{ // Начало тела программы
  int x,y,z; // Объявляем переменные
  printf(" input x,y "); // Вывод сообщения на экран
  scanf("%d %d",&x,&y); // Считываем x,y
  z=max(x,y); // Вызов функции с фактическими параметрами
  printf("x=%d y=%d max=%d",x,y,z); // Выводим результат на экран
  getch(); // Задержка
} // Конец тела программы
int max(int a ,int b) // Заголовок функции с формальными параметрами
{ // Начало тела функции
  int c; // Объявляем переменные
  if (a>b) // Если a больше b, то
  c=a; // Переменной c присваиваем a
  else // Иначе
  c=b; // Переменной c присваиваем b
  return c; // Возвращаем переменную c
} // Конец тела функции
```

Задание:

Написать программу, максимально разбив ее на функции и нарисовать блок-схему к ней согласно данному варианту.

Вариант 1.

Даны две матрицы A(3x4) и B(3x3). Программа находит и выводит на экран сумму элементов последнего столбца матрицы A и сумму элементов 2-й строки матрицы B.

Вариант 2.

Даны матрицы A(3x4). Программа определяет, сколько положительных элементов в матрице.

Вариант 3.

Даны две матрицы A(3x3) и B(3x3). Программа находит и распечатывает сумму элементов 5-го столбца матрицы A и сумму элементов последней строки матрицы B.

Вариант 4.

Дана матрица A(3x4). Программа находит максимальный элемент I-ой строки и выводит его на экран.

Вариант 5.

Даны матрицы A[3][3] и B[3][3]. Программа выводит матрицу C=A+B

Вариант 6.

Программа вводит целую матрицу A(3x4) и выводит транспонированную к ней матрицу A', все элементы которой уменьшены в 2 раза.

Вариант 7.

Написать программу, которая вводит целочисленную матрицу A(4x4), находит сумму элементов главной диагонали и количество четных элементов матрицы.

Вариант 8.

Дана матрица A(4x3). Программа находит и распечатывает сумму элементов 2-го столбца матрицы A.

Вариант 9.

Программа находит среднее арифметическое вещественной матрицы $A(2 \times 2)$.

Вариант 10.

Дана матрица $A(3 \times 4)$. Программа находит максимальный элемент 2-го столбца.

Вариант 11.

Даны матрицы $A(4 \times 3)$ и $B(2 \times 2)$. Программа определяет, сколько нулей в матрице A и сколько единиц в матрице B , результат выводит на экран.

Вариант 12.

Дана матрица $A(3 \times 4)$. Определить количество положительных элементов каждого столбца матрицы A и сформировать из них одномерный массив $B(4)$.

Вариант 13.

Найти минимальный элемент матрицы $X(4 \times 4)$ и вывести на экран номер строки и столбца, в которых он находится.

Вариант 14.

Даны матрицы $A[3 \times 3]$ и $B[2 \times 2]$. Сформировать одномерный массив C , который содержит все элементы двумерного массива A , меньшие максимального элемента матрицы B .

Вариант 15.

Даны матрицы $A[4 \times 4]$, $B[4 \times 4]$ и q . Вывести на экран все элементы массива A , которые меньше, чем максимальный элемент столбца матрицы B , номер которого задается значением q .

Вариант 16.

Вычислить сумму чисел в каждой строке матрицы $A(3 \times 4)$.

Вариант 17.

Вычислить произведение чисел в каждом столбце матрицы $A(3 \times 4)$.

Вариант 18.

Вычислить количество положительных чисел в каждом столбце матрицы $A(3 \times 4)$.

Вариант 19.

Дана матрица $A(4 \times 4)$. Вычислить сумму чисел для каждого столбца, удовлетворяющих условию $a[i, j] > x$. Здесь x - произвольная величина.

Вариант 20.

Дана матрица $A(4 \times 4)$. Вычислить значение наибольшего элемента для каждой строки массива.

Вариант 21.

Дана матрица $A(3 \times 4)$. Вычислить значение наименьшего элемента для каждого столбца массива.

Вариант 22.

Дана матрица $A(3 \times 4)$. Вычислить значение наибольшего элемента и его индекс для каждого столбца массива

Вариант 23.

Дана матрица $A(3 \times 4)$. Вычислить сумму элементов для каждого столбца. Поменять местами столбцы с максимальным и минимальным значением суммы.

Вариант 24.

Дана матрица $A(3 \times 4)$. Вычислить сумму отрицательных чисел в каждой строке.

Вариант 25.

Дана матрица $A(3 \times 4)$. Вычислить среднее значение чисел в каждой строке массива.

Лабораторная работа 7. Программная реализация строковых типов данных

Цель: Изучение правил описания, ввода-вывода и основных функций обработки символьных (строковых) данных..

Что нужно знать для выполнения работы

Для работы со строками в языке C есть специальные функции, которые описаны в библиотечном файле `string.h`. Наиболее часто используются такие функции как:

`strcpy()`, `strcat()`, `strcmp()`.

Рассмотрим их поподробнее. Вызов функции `strcpy()` имеет вид:

```
strcpy(s1, s2);
```

Эта функция позволяет копировать содержимое строки `s2` в строку `s1`. Массив `s1` должен быть достаточно вместительным, чтобы в него поместилась строка `s2`. Если нужного места не хватает, то компилятор не выдает сообщение об ошибке или хотя бы предупреждение. Программа не прервет свою работу. Однако же все это вполне может привести к порче данных или самой программы. В результате программа может работать, но неправильно. И никто об этом не узнает.

Функция `strcat()` имеет вызов:

```
strcat(s1, s2);
```

Функция `strcat()` присоединяет строку `s2` к строке `s1` и помещает новую строку в массив, где находилась строка `s1`. Строка `s2` не изменяется. При этом, как и в предшествующем случае, нет никакого контроля. Нулевой байт строки `s1` заменяется первым символом строки `s2`. Новая строка автоматически завершится нулевым байтом.

Функция `strcmp()` сравнивает строку `s2` со строкой `s1`. Если строка `s1` лексикографически меньше строки `s2`, возвращает отрицательное число.

```
if (strcmp(s1,s2) == -1)
```

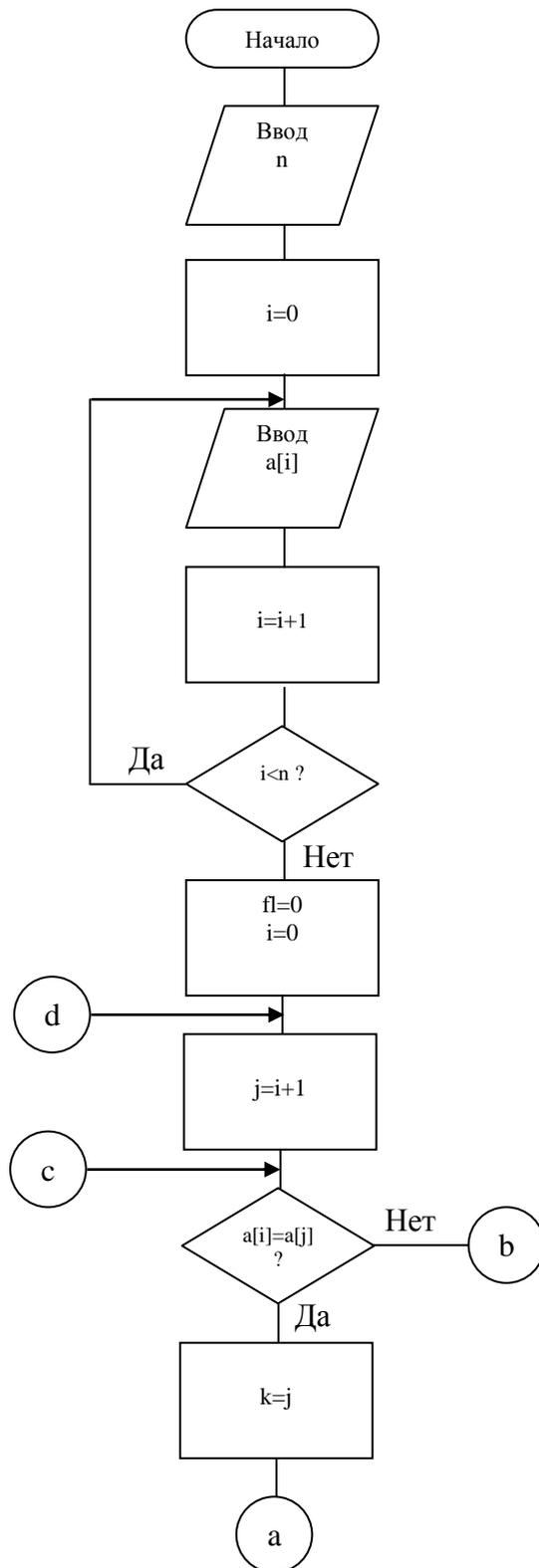
Контрольные вопросы:

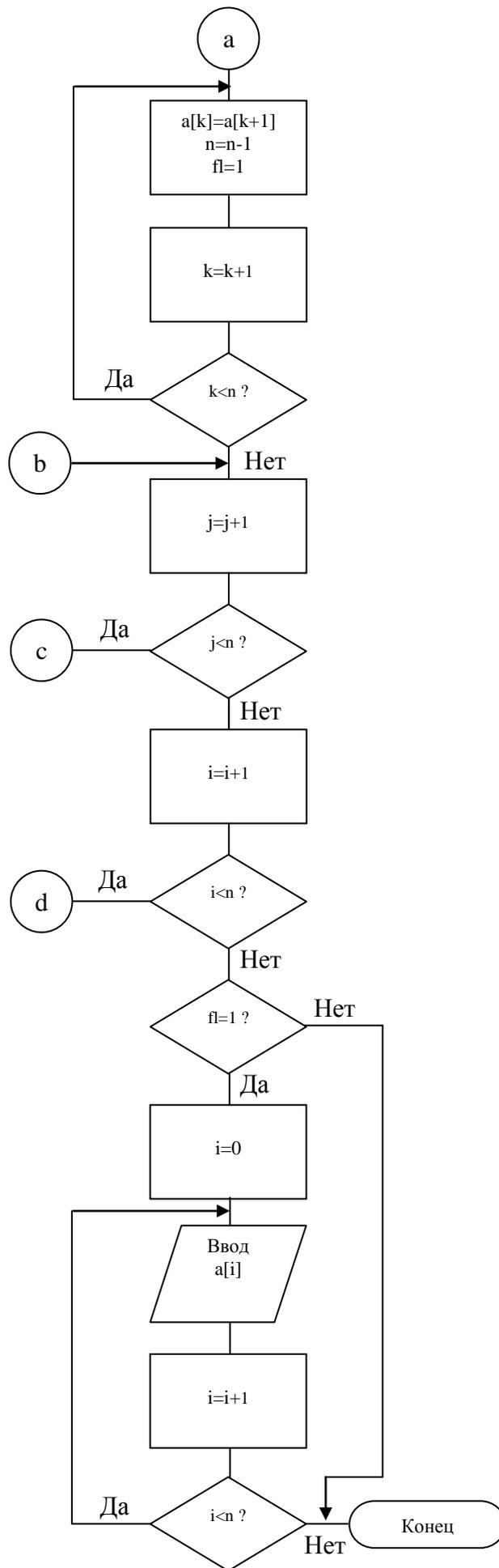
1. Строковые данные?
2. Строковые переменные?
3. Строки изменяемой и фиксированной длины?
4. Функции обработки строк?

Примеры выполнения задания лабораторной работы

Условие задачи: Удалить из строки все двойные символы. Пример: “asddewwf” → “asdewf”

Схема алгоритма





Листинг программы:

```
#include <stdio.h>
#include <string.h>
void main ()
{
    char a[255];
    int i, j, k, fl, n;
    scanf("%d", &n);
    for(i=0; i<n; i++)
        scanf("%c", &a[i]);
    fl=0;
    for(i=0; i<n; i++) {
        for(j=i+1; j<n; j++) {
            if(a[i]==a[j]) {
                for(k=j; k<n; k++)
                    a[k]=a[k+1];
                n--;
                fl=1;
            }
        }
    }
    if(fl==1) {
        for(i=0; i<n; i++)
            printf("%c ", a[i]);
    }
}
```

```
// Вызываем библиотеку stdio.h
// Вызываем библиотеку string.h
// Объявляем главную функцию программы
// Начало тела программы
// Объявляем массив
// Объявляем переменные
// Считываем количество символов
// Задаем цикл по i
// Считываем массив
// Обнуляем переменную fl
// Задаем цикл по i, начало тела цикла
// Задаем цикл по j, начало тела цикла
// Если текущий символ равен последующему, то
// Задаем цикл по k
// На место текущего ставил последующий символ
// Уменьшаем n на единицу
// Переменной fl присваиваем единицу
// Конец тела цикла j
// Конец тела цикла i
// Если fl равна единицы, то
// Задаем цикл по i, начало тела цикла
// Выводим результат на экран
// Конец тела цикла i
// Конец тела программы
```

Задание:

Написать программу и нарисовать блок-схему к ней согласно данному варианту.

Вариант 1.

В произвольном тексте найти и отпечатать все слова, начинающиеся с буквы а.

Вариант 2.

В произвольном тексте найти и отпечатать все слова, заканчивающиеся буквой г.

Вариант 3.

В произвольном тексте найти самое короткое слово.

Вариант 4.

В произвольном тексте найти и отпечатать слова, содержащие букву е, но не содержащие букву w.

Вариант 5.

Отпечатать второе и четвертое слова произвольного текста.

Вариант 6.

В 2-х предложениях найти и распечатать совпадающие слова.

Вариант 7.

Во введенном предложении расположите слова в обратной последовательности.

Вариант 8.

В произвольном тексте найти и отпечатать все слова, начинающиеся с буквы u.

Вариант 9.

В произвольном тексте найти и отпечатать все слова, заканчивающиеся буквой m.

Вариант 10.

В произвольном тексте найти самое длинное слово.

Вариант 11.

В произвольном тексте найти и отпечатать слова, содержащие букву t, но не содержащие букву q.

Вариант 12.

Отпечатать первое и четвертое слова произвольного текста.

Вариант 13.

В 3-х предложениях найти и распечатать совпадающие слова.

Вариант 14.

Отпечатать первое и третье слова произвольного текста.

Вариант 15.

В произвольном тексте найти и отпечатать все слова, начинающиеся с буквы y.

Вариант 16.

В произвольном тексте найти и отпечатать все слова, заканчивающиеся буквой w.

Вариант 17.

В произвольном тексте найти и отпечатать слова, содержащие букву i, но не содержащие букву m.

Вариант 18.

Отпечатать второе и пятое слова произвольного текста.

Вариант 19.

В 4-х предложениях найти и распечатать совпадающие слова.

Вариант 20.

Отпечатать второе и третье слова произвольного текста.

Вариант 21.

В произвольном тексте найти и отпечатать все слова, начинающиеся с буквы r.

Вариант 22.

В произвольном тексте найти и отпечатать все слова, заканчивающиеся буквой k.

Вариант 23.

В произвольном тексте найти и отпечатать слова, содержащие букву f, но не содержащие букву n.

Вариант 24.

Отпечатать первое и второе слова произвольного текста.

Вариант 25.

В произвольном тексте найти и отпечатать слова, содержащие букву a, но не содержащие букву h.