МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ХАРЬКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам и расчетно-графическому заданию по дисциплине «Информатика», Часть 2 для студентов, обучающихся по направлениям 6.050801 «Микро- и наноэлектроника», 6.050802 «Электронные устройства и системы» дневной и заочной форм обучения

Утверждено редакционно-издательским советом университета, протокол № 2 от 23.06.2016 г.

Харьков НТУ «ХПИ» 2016 **Методические указания** к лабораторным работам и расчетнографическому заданию по дисциплине «Информатика», Часть 2 / Сост. Король Е.И., Вержановская М. Р., Махонин Н. В. – Харьков: НТУ «ХПИ», 2016.-68 с.

Составители: Е.И. Король

М. Р. Вержановская

Н. В. Махонин

Рецензент В.В. Замаруев

Кафедра промышленной и биомедицинской электроники

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА №1 ИСПОЛЬЗОВАНИЕ ОДНОМЕРНЫХ	
МАССИВОВ	5
1.1 Одномерные статические массивы	5
1.2 Пример выполнения работы	5
1.3 Индивидуальные задания	6
ЛАБОРАТОРНАЯ РАБОТА №2 ЗАПОЛНЕНИЕ МАССИВОВ	
СЛУЧАЙНЫМИ ЧИСЛАМИ	8
2.1 Генератор случайных чисел	8
2.2 Пример выполнения работы	9
2.3 Индивидуальные задания	11
ЛАБОРАТОРНАЯ РАБОТА №3 РАБОТА СО СТРОКОВЫМИ	
ПЕРЕМЕННЫМИ	13
3.1 Объявление строк	13
3.2. Функции для работы со строками	13
3.3 Пример выполнения работы	14
3.4 Индивидуальные задания	15
ЛАБОРАТОРНАЯ РАБОТА №4 ПРОГРАММИРОВАНИЕ С	
ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ	17
4.1 Объявление функции	17
4.2. Передача параметров	18
4.3. Перегрузка функций и указатель на функцию	18
4.4 Пример выполнения работы	19
4.5 Индивидуальные задания	19
ЛАБОРАТОРНАЯ РАБОТА №5 РАБОТА С БИТАМИ ЦЕЛОГО ЧИС	ЛА22
5.1 Побитовые логические операции	22
5.2 Пример выполнения работы	23
5.3 Индивидуальные задания	25
ЛАБОРАТОРНАЯ РАБОТА №6 ПРОГРАММИРОВАНИЕ С	
ИСПОЛЬЗОВАНИЕМ СТРУКТУР	31
6.1 Объявление структур	31
6.2 Пример выполнения работы	32
6.3 Инливилуальные залания	33

36
36
37
40
41
43
43
44
45
46
47
47
48
49
Z S
53
53
54
55
58
66
67

ЛАБОРАТОРНАЯ РАБОТА №1ИСПОЛЬЗОВАНИЕ ОДНОМЕРНЫХ МАССИВОВ

1.1 Одномерные статические массивы

В программе одномерный массив объявляется следующим образом:

тип имя массива [размер];

Пример объявления 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, поэтому при обращении к элементам за пределами массива вы обратитесь к случайным данным, расположенным в памяти, не относящейся к массиву.

Элементам массива могут быть присвоены начальные значения при описании. Для этого они указываются после описания через знак присвоения в операторных скобках через запятую.

int $mas[4]={3, 6, 10, 2};$

Значений указывать больше размера массива нельзя. При указании меньшего количества начальных констант, остальные элементы получат значения по правилам инициализации переменных.

1.2 Пример выполнения работы

Условие: Удалить из одномерного массива целых чисел все отрицательные элементы.

Текст программы:

```
#include <stdio.h>
#include <math.h>
#include <locale.h>
int main(void) {
   int mas[100];
   int i;
   int k;
   int j;

   setlocale(LC_ALL, "Russian");

// Ввод элементов массива
```

```
printf("Введите размер массива(<100):");
    scanf("%i",&k);
    for (i=0;i<k;i++){
        printf("Введите %i-й элемент массива:",i);
        scanf("%i",&mas[i]);
  // Удаление отрицательных элементов
    for (i=0, j=0; i< k; i++)
        if (mas[i]>0) {
                mas[j]=mas[i];
                j++;
        }
  // Вывод элементов, полученного массива, по 5 элементов в
строку
    printf("Получен массив:");
    for (i=0;i< j;i++)
        printf("%c %4i",(i%5)?' ':'\n',mas[i]);
    return 0;
}
```

1.3 Индивидуальные задания

Выполнить задание в соответствии с выданным вариантом.

- 1. Задан массив из k символов. Преобразовать массив следующим образом: сначала должны стоять цифры, входящие в массив, а затем все остальные символы. Взаимное расположение символов в каждой группе не должно изменяться.
- 2. Дан массив целых чисел. Определить количество четных элементов и количество элементов, оканчивающихся на цифру 5.
- 3. Задан массив из k символов. Преобразовать массив следующим образом: расположить символы в обратном порядке.
- 4. В массиве записаны результаты 20 игр футбольной команды (если игра окончилась выигрышем данной команды, то записано число 3, проигрышем— 2, если игра окончилась вничью— 1). Определить количество выигрышей, количество ничьих и количество проигрышей данной команды.
- 5. Задан массив из k чисел. Найти число, наиболее часто встречаюшееся в этом массиве.
- 6. В массиве записаны оценки по иностранному языку каждого из 22 учеников класса. Определить количество пятерок, количество четверок, количество троек и количество двоек.

- 7. Задан массив из k чисел. Отсортировать элементы массива по возрастанию.
- 8. Найти число пар соседних элементов массива, являющихся четными числами.
- 9. Задан массив из k чисел. Найти числа, входящие в массив только один раз.
- 10. Задан массив из k чисел. Сдвинуть элементы массива циклически на n позиций влево.
- 11. Найти число элементов массива, которые больше своих "соседей", т. е. предшествующего и последующего.
- 12. Задан массив из k чисел. Сдвинуть элементы массива циклически на n позиций вправо.
- 13. Известны данные о количестве осадков, выпавших за каждый день марта. Верно ли, что осадков не было 10 дней в месяц?
- 14. Задан массив из k чисел. Преобразовать массив следующим образом: все отрицательные элементы массива перенести в начало, а все остальные в конец, сохранив исходное взаимное расположение, как среди отрицательных, так и среди положительных элементов.
- 15. Найти среднее арифметическое элементов массива, больших числа 10.
- 16. Задан массив из k символов. Создать два новых массива: в первый перенести все цифры из исходного массива, во второй все остальные символы.
- 17. Найти среднее арифметическое элементов массива, меньших некоторого числа е.
- 18. Задан массив из k символов. Определить, симметричен ли он, т. е. читается ли он одинаково слева направо и справа налево .
- 19. Найти число пар соседних элементов массива, оканчивающихся нулем.
- 20. Задано два массива. Найти наименьшие среди элементов первого массива, которые не входят во второй массив.
- 21. Найти средние арифметические положительных и отрицательных элементов массива.

- 22. Задан массив из k чисел. Определить количество инверсий в массиве (т. е. таких пар элементов, в которых большее число находится слева от меньшего).
- 23. Задан массив из k символов. Удалить из него повторные вхождения каждого символа.
- 24. Задан массив из k символов. Определить количество различных элементов в массиве.
- 25. Задан массив из k символов латинского алфавита. Вывести на экран в алфавитном порядке все символы, которые входят в этот массив по одному разу.

ЛАБОРАТОРНАЯ РАБОТА №2 ЗАПОЛНЕНИЕ МАССИВОВ СЛУЧАЙНЫМИ ЧИСЛАМИ

2.1 Генератор случайных чисел

Не всегда надо заполнять числовые одномерные и двумерные массивы порядковыми номерами или конкретными значениями. Возможно заполнить элементы массива случайными числами. В языке С для этого есть специальные функции rand() и srand(). Они находятся в библиотечном файле cstdlib, поэтому чтобы их применять в программе, необходимо подключить этот библиотечный файл: #include <stdlib.h>.

Если воспользоваться только функцией rand() — будут получаться одинаковые "случайные числа" от запуска к запуску. Получается, что числа генерируются не совсем случайные. Чтобы добиться "настоящей" случайности чисел при повторных запусках программы, необходимо применить функцию srand() до функции rand(). При этом надо передать ей в виде параметра функцию time() с параметром NULL: srand(time(NULL));. Таким образом srand() получает в виде параметра текущее системное время, которое при каждом запуске программы будет разным. Это позволит функции rand() каждый раз генерировать именно случайные числа. Для использования time() необходимо подключить библиотечный файл time.h: #include <time.h>.

При использовании команды rand() будут генерироваться числа в диаппазоне — от 0 до 32767. Возможно понадобится заполнить массив числами от 200 до 300, от 0.1 до 1, от -20 до 20. Такую генерацию случайных чисел возможно реализовать с использованием арифметических команд следующим образом:

rand() % 7 — rand() генерирует число и далее вычисляется остаток от деления на 7 от этого числа. Понятно, что это могут быть числа только от 0 до 6. Например генерируется 50 — остаток от деления на 7 будет равен 1, генерируется 49 — остаток от деления на 7 будет равен 0.

1 + rand() % 7 — очень похоже на предыдущий случай, 0-я уже не будет, а вот 7 появится в диапазоне. Например генерируется 49 — остаток от деления на 7 равен 0 и к нему добавляется единица, генерируется 6 — остаток от деления на 7 равен 6 и опять же добавляется единица.

200 + rand() % 101 - даст число от 200 до 300. Например генерируется 100 - остаток от деления на 101 равен 100 и добавляется 200. Получается число 300. Генерируется 202: 200 + (202 % 101) = 200 + 0 = 200.

rand() % 41 - 20 — от — 20 до 20. Например генерируется 1: (1 % 40) — 20 = 1 - 20 = -19; генерируется 30: 30 - 20 = 10.

0.01* (rand() % 101) — от 0.01 до 1. Например генерируется 55: 0.01* 55 = 0.55.

Т.о. можно получить требуемый диапазон изменения случайных чисел.

2.2 Пример выполнения работы

Условие: заполнить массив целых чисел из 100 элементов в диапазоне от - 50 до 50. Во всех последовательностях отрицательных чисел ограничить значения тех элементов, абсолютное значение которых превышает абсолютное среднее для этой последовательности

Текст программы:

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <locale.h>
int Ar[100]; /*массив, который обрабатывается */
```

```
int main(void) {
 int i, j; /* индексы в массиве */
 int av; /* среднее значение */
 int nn; /*количество эл-тов в последовательности*/
 int ib; /* индекс начала последовательности */
    setlocale(LC_ALL, "Russian");
    srand(time(NULL)); /* инициализация rand */
  /* заполнение массива случайными числами */
    for (i=0; i<100; Ar[i++]=rand()%101-50);
  /* вывол начального массива */
    printf("Начальный массив:");
    for (i=0; i<100; i++)
printf("%c %3d",i%10?' ':'\n',Ar[i]);
    printf("\n\n");
    for (nn=i=0; i<100; i++) {/*перебор массива */
      if (Ar[i]<0)
      /* обработка отрицательного элемента */
        if (!nn) {
        /* начало последовательности */
        /* запомнить индекс начала,
           начальное значение накопителя суммы
           и счетчика элементов */
            ib=i; av=Ar[i]; nn=1;
        else {
        /* накопление суммы,
           подсчет количества */
            av+=Ar[i]; nn++;
        }
      /*конец обработки отрицательного элемента */
      else /* обработка положительного элемента */
        if (nn) {
        /* если есть необработанная
           отрицательная последовательность */
            av/=nn; /* усреднение */
        /* перебор всей последовательности
           с ограничением */
            for (j=ib; j<i; j++)
                if (Ar[j]>av) Ar[j]=av;
                    nn=0;
/* последовательность обработана */
       } /* конец если есть необработанная... */
    } /* конец перебор массива */
    if (nn) /* если не обработана последняя
             отрицательная последовательность */
        for (av/=nn, j=ib; j<i; j++)
            if (Ar[j]>av) Ar[j]=av;
  /* вывод результатов */
    printf("Массив-результат:");
```

```
for (i=0; i<100; i++)
printf("%c %3d",i%10?' ':'\n',Ar[i]);
    return 0;
}</pre>
```

2.3 Индивидуальные задания

Объявить массив целых чисел и заполнить его случайными значениями. Размер массива и диапазон значений его элементов заданы в Вашем варианте индивидуального задания. В индивидуальных заданиях указано также, какую обработку массива следует произвести.

Для всех вариантов задания следует иметь в виду следующее:

- 0 считается положительным числом, если в задании не оговорен какой-то другой его статус;
- когда речь идет о какой-то последовательности чисел, имеется в виду последовательность с длиной, большей 1;
- в тех случаях, когда задание требует выполнения каких-то вычислений, разрешается выполнять их с той точностью, которую обеспечивают операции целочисленной арифметики.

Таблица 2.1 – Индивидуальные задания

№ вар.		Диапазон значений	Что нужно сделать
1	2	3	4
1	50	_ 1 ()() _ 1 ()()	Заменить все элементы с отрицательным значением на значение минимального не равного 0 положительного элемента
2	200	-50 - 50	Подсчитать количество пар соседних элементов с одинаковыми значениями
3	200	0 - 100	Подсчитать количество участков, которые образуют непрерывные последовательности чисел с неуменьшающимися значениями
4	200	-50 - 50	Подсчитать количество пар соседних элементов, которые имеют противоположные знаки
5	100		Вывести начальные индексы всех непрерывных последовательностей неотрицательных чисел, длина которых больше 5
6	100		Найти ту непрерывную последовательность положительных чисел, сумма элементов в которой максимальная

Продолжение табл. 2.1

1	2	3	4
7	100	-100 - 100	Разместить все элементы с положительными значениями в левой части массива, элементы с отрицательными значениями - в правой, а нули - между ними
8	100	-100 - 100	Заменить все элементы с отрицательными значениями средним арифметическим значением всех положительных элементов
9	200	0 - 100	Найти непрерывный участок из 10 элементов, сумма которых максимальна
10	200	0 - 100	Найти значение 3-го по величине элемента и значения всех элементов массива, которые его превышают, заменить на найденное значение
11	100	-50 - 50	Найти количество пар соседних элементов, которые имеют одинаковые абсолютные значения, но противоположные знаки
12	200	-100 - 100	Во всех последовательностях положительных чисел заменить значения элементов с максимальным и минимальным значением на среднее для этой последовательности
13	200	0 - 100	Найти непрерывный участок из 10 элементов, который имеет наибольшее среднее значение элементов
14	100	-50 - 50	Во всех последовательностях положительных чисел изменить порядок элементов на противоположный
15	200	-100 - 100	Во всех последовательностях положительных чисел ограничить снизу значения тех элементов, значения которых меньше среднего для этой последовательности
16	100	-50 - 50	Заменить все элементы с положительными значениями абсолютным значением отрицательного элемента с максимальным абсолютным значением
17	200	-50 - 50	Вывести начальные индексы всех непрерывных последовательностей чисел с неувеличивающимися значениями
18	100	-50 - 50	Вывести начальные индексы всех непрерывных знакопеременных последовательностей чисел
19	200	-50 - 50	Подсчитать количество всех, непрерывных последовательностей положительных чисел, длина которых больше 7
20	200	-50 - 50	Найти из непрерывных последовательностей отрицательных чисел ту, которая имеет наибольшее абсолютное значение среднего арифметического ее элементов
21	100	-100 - 100	Разместить все элементы с нулевыми значениями в левой части массива, элементы с отрицательными значениями - за ними, а за ними - элементы с положительными значениями
22	100	-100 - 100	Ограничить значения всех элементов, абсолютные значения которых превышают среднее для положительных и отрицательных элементов

Окончание табл. 2.1

1	2	3	4
23	100		Заменить каждый элемент на среднее арифметическое его и его соседей слева и справа.
24	200	-50 - 50	Найти непрерывные участки, на которых сумма элементов равна 0
25	200		Для каждого положительного элемента определить, есть ли в массиве отрицательный элемент с противоположным значением; если да, заменить эти элементы на 0

ЛАБОРАТОРНАЯ РАБОТА №3 РАБОТА СО СТРОКОВЫМИ ПЕРЕМЕННЫМИ

3.1 Объявление строк

Объявление строки аналогично объявлению массива:

char имя строки [размер]

Важной особенностью является то, что строка должна обязательно заканчиваться нулевым символом $'\ 0' - ($ нуль - терминатор). Длина строки равна количеству символов плюс нулевой символ.

3.2. Функции для работы со строками

Функции для работы со строками содержатся в библиотеке string.h (под -ключение: #include string.h).

Наиболее часто применяются следующие функции:

char *strcpy(st1, st2) – копирует содержимое строки st2, включая нулевой символ, в строку st1.

char *strcat(st1, st2) – добавляет справа к строке st1 содержимое строки st2.

int strcmp(st1, st2) – сравнивает содержимое строк st2 и st1. Если st1 < st2, то результат равен -1, если st1 = st2 – результат равен нулю, если st1 > st2 – результат равен 1.

char ***strstr**(**st1**, **st2**) – возвращает указатель на первое появление подстроки st2 в строке st1.

char ***strchr**(**st**, **sh**) - возвращает указатель на первое появление символа **ch** в строке **st**.

char ***strtok**(**st1**, **st2**) – возвращает указатель на лексему, находящуюся в строке st1. При первом вызове функция возвращает указатель на первый символ в st1, а после первой лексемы устанавливает нулевой символ. При последующих вызовах функции со значением NULL в качестве первого аргумента указатель аналогичным образом переходит к следующим лексемам. После того как закончились все лексемы , указатель устанавливается в NULL.

int strlen(st) — возвращает длину строки st.

char *strrev(st) – изменяет порядок следования символов в строке на противоположный.

char *strdup(st); – дублирует строку st.

char *strlwr(st) — конвертирует символы строки st к нижнему регистру.

char *strupr(st) – конвертирует символы строки st к верхнему регистру.

int atoi(st) – преобразует строку st в число целого типа (int).

double atof(\mathbf{st}) – преобразует строку \mathbf{st} в число действительного типа.

char *itoa(a, st, base) – преобразует число целого типа а в строку st (base – основание системы счисления).

char *gcvt(a, dec, st) – преобразует число действительного типа а в строку st. Значение dec указывает на число десятичных разрядов (не более 18).

Кроме того применяются следующие функции из библиотеки stdio.h.

char * **gets**(**char** * **st**) — считывает символы в строку st из стандартного потока ввода до символа новой строки '\n'.

puts(**const char** * **st**) — выводит строку типа на которую указывает параметр st в стандартный поток вывод и добавляет символ новой строки \n' .

3.3 Пример выполнения работы

Условие: Выделить и вывести на экран все слова произвольной строки. Слова отделяются друг от друга одним или несколькими пробелами.

Текст программы:

#include <stdio.h>

```
#include <string.h>
#include <locale.h>
char st[100], sl[100];
int main(void) {
     setlocale(LC_ALL, "Russian");
     int k=0, i;
     gets(st);
     strcat(st," ");
     int n=strlen(st);
     if (n<2) return 1;
     s1[0]='\0';
     for (i=0; i<n; i++)
          if (st[i] != ' ') {
               sl[k]=st[i];
               s1[k+1] = ' \ 0';
               k++;
          }
          else
               if (strlen(sl)>0) puts(sl);
               sl[0] = ' \ 0';
               k=0;
     return 0;
}
```

3.4 Индивидуальные задания

- 1. Дана строка, состоящая из групп нулей и единиц. Каждая группа отделяется от другой одним или несколькими пробелами . Найти количество групп с пятью символами.
 - 2. Дано предложение. Определить число пробелов в нем.
- 3. Дана строка, состоящая из групп нулей и единиц. Найти и вывести на экран самую короткую группу.
- 4. Дано предложение. Определить число вхождений в него некоторого символа.
- 5. Дана строка, состоящая из групп нулей и единиц. Подсчитать количество символов в самой длинной группе.
 - 6. Дано предложение. Определить долю (в %) букв 'а' в нем.
- 7. Дана строка, состоящая из групп нулей и единиц. Найти и вывести на экран группы с четным количеством символов.

- 8. Дано предложение. Определить долю (в %) букв 'о' в нем.
- 9. Дана строка, состоящая из групп нулей и единиц. Подсчитать количество единиц в группах с нечетным количеством символов.
- 10. Дано предложение. Определить, сколько в нем одинаковых соседних букв.
- 11. Дана строка, состоящая из букв, цифр, запятых, точек, знаков «+» и «-». Выделить подстроку, которая соответствует записи целого числа.
- 12. Дано предложение. Определить число вхождений в него некоторого буквосочетания.
- 13. Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков «+» и «-». Выделить подстроку, которая соответствует записи вещественного числа с фиксированной точкой.
- 14. Дано предложение. В нем слова разделены одним пробелом (начальные и конечные пробелы и символ в предложении отсутствуют). Определить количество слов в предложении.
- 15. Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков «+» и «-». Выделить подстроку, которая соответствует записи вещественного числа с плавающей точкой.
 - 16. Дан текст. Определить, сколько в нем предложений.
- 17. Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Вывести на экран числа этой строки в порядке возрастания их значений.
 - 18. Дано предложение. Определить, сколько в нем гласных букв.
- 19. Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Вывести четные числа этой строки.
- 20. Дано предложение. Определить, каких букв в нем больше: 'м' или 'н'.
- 21. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Вывести на экран слова этого текста в порядке, соответствующем латинскому алфавиту.
- 22. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Вывести на экран порядковый номер слова, накры-

вающего k- ю позицию (если на k- ю позицию попадает пробел, то номер предыдущего слова).

- 23. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Разбить исходную строку на две подстроки, причем первая длиной k символов (если на k ю позицию попадает слово, то его следует отнести ко второй строке).
- 24. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Вывести на экран порядковый номер слова максимальной длины и номер позиции в строке, с которой оно начинается.
- 25. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Вывести на экран порядковый номер слова минимальной длины и количество символов в этом слове.

ЛАБОРАТОРНАЯ РАБОТА №4 ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ

4.1 Объявление функции

Функция — это последовательность операторов , оформленная таким образом , что ее можно вызвать по имени из любого места программы . Функция описывается следующим образом:

```
тип возвращаемого значения имя функции ( список параметров ) {
тело функции
}
```

Первая строка описания называется заголовком функции. Тип возвращаемого значения может быть любым, кроме массива или функции. Допустимо не возвращать никакого значения (тип void).

В С не допускается вложение функций друг в друга. Выход из функции осуществляется следующими способами:

1. Если необходимо вернуть определенное значение, то выход осуществляется оператором

return выражение;

2. Если нет необходимости возвращать вычисленное значение, то выход осуществляется по достижении закрывающей операторной скобки или при выполнении оператора return без выражения.

4.2. Передача параметров

При работе важно соблюдать следующее правило: при объявлении и вызове функции параметры должны соответствовать по количеству, порядку следования и типам. Функция может не иметь параметров, в этом случае после имени функции обязательно ставятся круглые скобки. Существует три основных способа передачи параметров: передача по значению, ссылке или указателю.

1. Передача параметров по значению

В момент обращения к функции в памяти создаются временные переменные с именами, указанными в списке параметров. Во временные переменные копируются значения фактических параметров:

print (x, y).

2. Передача параметров по ссылке

При передаче параметров по ссылке передается не значение соответствующей переменной, а ее адрес. Для указания на данный способ передачи после имени параметра ставится символ «&»:

3. Передача параметров по указателю

В отличие от передачи по ссылке адрес переменной передается в функцию не с использованием операции разадресации (&), а операцией косвенной адресации (*):

где px, py – указатели на переменные x и y, соответственно.

4.3. Перегрузка функций и указатель на функцию

В языке С допустимо использование нескольких функций с одинаковым именем, но различным числом или типами параметров. Такое свойство называется перегрузкой функций. Перегруженные функции различаются компилятором по типам и числу параметров.

Так как имя функции является указателем на начало функции в оперативной памяти, то можно объявлять указатели на функции для последующего их использования в программе.

При объявлении указатель должен возвращать тот же тип и иметь такие же аргументы, как и функция, на которую он будет указывать. Например, указатель на функцию

double y(double x, int n);

будет иметь вид:

double (*fun)(double, int).

При этом свое значение указатель на функцию получает как:

fun = y;

4.4 Пример выполнения работы

Условие: даны два предложения. Найти общее количество букв 'н' в них. (Определить функцию для расчета количества букв 'н' в предложении.)

Текст программы:

```
#include <stdio.h>
#include <locale.h>
char st[100];
int count_n(char s[]){
    int j=0,i;
    for(i=0;s[i]!='\0';i++)
        if (s[i]=='H')j++;
    return j;
}
int main(void) {
     setlocale(LC_ALL, "Russian");
     gets(st);
     printf("Количество букв 'н' в предложе-
нии:%i",count n(st));
     return 0;
}
```

4.5 Индивидуальные задания

Выполнить задание в соответствии с выданным вариантом с использованием функций.

1. Даны вещественные числа x_1 , y_1 x_2 , y_2 , ... x_5 , y_5 . Найти площадь пятиугольника, вершины которого имеют координаты (x_1, y_1) - (x_5, y_5) . Для это-

го разбить его на треугольники. (Определить функцию для расчета площади треугольника по координатам его вершин.)

- 2. Даны две последовательности целых чисел: a_1 , a_2 , ... a_8 и b_1 , b_2 , ..., b_8 . Найти количество четных чисел в первой из них и количество нечетных во второй. (Определить функцию, позволяющую распознавать четные числа.)
- 3. Упорядочить массив целых чисел, методом пузырька, используя функцию перестановки значений двух элементов массива.
- 4. Даны натуральное число n и целые числа a_1 , ..., a_n . Найти количество чисел, являющихся степенями пятерки. (Определить функцию, позволяющую распознавать степени пятерки.)
- 5. Найти все трехзначные простые числа. (Определить функцию, позволяющую распознавать простые числа.)
- 6. Составить процедуру, "рисующую" на экране прямоугольник из символов "*" с заданными размерами сторон.
- 7. Упорядочить массив целых чисел, используя функцию поиска номера минимального элемента массива с указанной позиции.
- 8. Даны два натуральных числа. Выяснить, в каком из них больше цифр. (Определить функцию для расчета количества цифр натурального числа.)
- 9. Получить все шестизначные счастливые номера. Счастливым называют такое шестизначное число, в котором сумма его первых трех цифр равна сумме его последних трех цифр. (Определить функцию для расчета суммы цифр трехзначного числа.)
- 10. Даны два натуральных числа. Выяснить, является ли хоть одно из них палиндромом ("перевертышем"), т. е. таким числом, десятичная запись которого читается одинаково слева направо и справа налево. (Определить функцию, позволяющую распознавать числа-палиндромы.)
- 11. Даны шесть различных чисел. Определить максимальное из них. (Определить функцию, находящую максимум из двух различных чисел.)
- 12. Составить функцию, которая печатает таблицу умножения до NxN в формате:

1x1=1	•••	•••	Nx1=N

- 1xN=N ... NxN=??
- 13. Даны натуральные числа а и b. Найти их наименьшее общее кратное. (Определить функцию для расчета наибольшего общего делителя двух натуральных чисел, используя алгоритм Евклида.)
- 14. Даны натуральные числа а и b, обозначающие соответственно числитель и знаменатель дроби. Сократить дробь, т. е. найти такие натуральные числа р и q, не имеющие общих делителей, что p/q = a/b. (Определить функцию для расчета наибольшего общего делителя двух натуральных чисел, используя алгоритм Евклида.)
- 15. Даны два предложения. Найти общее количество букв и в них. (Определить функцию для расчета количества букв в предложении.)
- 16. Составить программу для нахождения общего количества заданной буквы в трех заданных предложениях. (Определить функцию для расчета количества некоторой буквы в предложении.)
- 17. Составить функцию, которая печатает таблицу умножения до NxN в формате:

- 18. Упорядочить массив целых чисел, используя функцию поиска номера минимального элемента массива с указанной позиции.
- 19. Даны натуральное число n и целые числа a_1 , ..., a_n . Найти количество чисел являющихся полными квадратами. (Определить функцию, позволяющую распознавать полные квадраты.)
- 20. Даны стороны двух треугольников. Найти сумму их периметров и сумму их площадей. (Определить процедуру для расчета периметра и площади треугольника по его сторонам)
- 21. Даны два натуральных числа. Выяснить, в каком из них сумма цифр больше. (Определить функцию для расчета суммы цифр натурального числа.)
- 22. Составить процедуру, "рисующую" на экране горизонтальную линию из любого числа символов "*".

- 23. Составить процедуру, "рисующую" на экране вертикальную линию из любого числа символов "*"
- 24. Два простых числа (см. предыдущую задачу) называются "близнецами", если они отличаются друг от друга на 2 (таковы, например, числа 41 и 43). Напечатать все пары чисел-"близнецов", не превышающих число 200. (Определить функцию, позволяющую распознавать простые числа.)
- 25. Даны основания и высоты двух равнобедренных трапеций. Найти сумму их периметров и сумму их площадей. (Определить процедуру для расчета периметра и площади равнобедренной трапеции по ее основаниям и высоте.)

ЛАБОРАТОРНАЯ РАБОТА №5 РАБОТА С БИТАМИ ЦЕЛОГО ЧИСЛА

5.1 Побитовые логические операции

В языке предусмотрен ряд операций для работы с битами; эти операции нельзя применять к переменным типа float или double.

- **&** побитовое AND
- побитовое включающее OR
- ^ побитовое исключающее OR
- << сдвиг влево
- >> сдвиг вправо
- ~ дополнение (унарная операция)

Понять действие указанных операторов помогает следующая таблица истинностей

Таблица 6.1 – Таблица истинностей

X	Y	X & Y	$X \mid Y$	X ^ Y	~ X	X << 2
0	0	0	0	0	1	000
0	1	0	1	1	1	000
1	0	0	1	1	0	100
1	1	1	1	0	0	100

Побитовая операция AND часто используется для маскирования некоторого множества битов; например, оператор

 $\mathbf{c} = \mathbf{n} \& \mathbf{0177}$ (выделяется 7 бит: 0177 = 0b0111 1111)

передает в c семь младших битов n, устанавливая остальные в ноль.

Операция | побитового OR используется для включения битов:

$c = x \mid mask$

устанавливает на единицу те биты в x, которые равны единице в mask.

Следует быть внимательным и отличать побитовые операции & и | от логических операций && и ||, которые подразумевают вычисление значения истинности слева направо. Например, если x = 1, а y = 2, то значение (x & y) равно нулю, в то время как значение (x & y) равно единице.

Операции сдвига << и >> осуществляют соответственно сдвиг влево и вправо своего левого операнда на число битовых позиций, задаваемых правым операндом. Таким образом, (x << 2) сдвигает х влево на две позиции, заполняя освобождающиеся биты нулями, что эквивалентно умножению на 4. Сдвиг вправо величины без знака заполняет освобождающиеся биты знаковым битом ("арифметический сдвиг"), либо нулем ("логический сдвиг").

Унарная операция \sim дает дополнение к целому; это означает, что каждый бит со значением 1 получает значение 0 и наоборот. Эта операция обычно оказывается полезной в выражениях типа

х & ~037 (сбрасывается 5 бит: $037 = 0b0001\ 1111$, а $\sim 037 = 0b1110\ 0000$) где последние шесть битов x маскируются нулем.

5.2 Пример выполнения работы

Условие: составить программу, которая сначала вводит составные части 16-ти битной структуры данных, приведенной в Вашем варианте индивидуального задания, и формирует из них заданную упакованную структуру. Затем вводит упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей. Ввод и вывод упакованного значения осуществляется в шестнадцатеричном виде.

Параметры структуры данных:

Слово состояния устройства в системе ввода-вывода представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	C	C	C	C	C	0	F	В	N	N	N	N	N	N	N	N

где:

```
С..С - код состояния
```

F - признак ошибки (1/0)

В - признак занятости (0/1)

N.. N - количество байт, переданных в последней операции

Во время отладки программы пронаблюдать последовательные изменения переменной tmp в процессе формирования слова состояния. Результат записать как четыре последовательных значения.

```
Текст программы:
```

```
#include <stdio.h>
#include <locale.h>
int main(void) {
   int c; /* код состояния */
   int f; /* признак ошибки */
   int b; /* признак занятости */
   unsigned int n; /* количество байт */
   unsigned int tmp; /* слово состояния */
   setlocale(LC_ALL, "Russian");
//*********************
  /* ввод составных частей */
   printf("Введите код состояния (0 - 31) >");
   scanf("%i",&c);
   printf("Введите признак ошибки (0 / 1) >");
   scanf("%i",&f);
   printf("Введите признак занятости (0 / 1) >");
   scanf("%i",&b);
   printf("Введите количество переданных байт (0 - 255) > ");
   scanf("%u",&n);
  /* формирование упакованного кода */
   tmp=((unsigned int)c&0x1F)<<11; // Фиксируем значение 1
   tmp = ((unsigned int) f & 1) << 9;
                                     // Фиксируем значение 2
   tmp = ((unsigned int)b&1) << 8;
                                     // Фиксируем значение 3
                                     // Фиксируем значение 4
   tmp | = n\&0xFF;
  /* вывод результата */
   printf("\nСлово состояния устройства = %04x\n", tmp);
//***************
  /* ввод слова состояния устройства */
   printf("Введите слово состояния устройства n");
   printf("(16-ричное число от 0 до 0xFFFF) >");
   scanf("%x",&tmp);
  /* Выделение составных частей */
   c = (tmp >> 11) & 0x1F;
   f = (tmp >> 9) &1;
```

```
b=(tmp>>8)&1;
    n=tmp&0xFF;

/* вывод результатов */
    puts("\n");
    printf("Код состояния = %i\n",c);
    printf("Признак ошибки = %i\n",f);
    printf("Признак занятости = %i\n",b);
    printf("Количество переданных байт = %i\n",n);
    return 0;
}
```

5.3 Индивидуальные задания

Произвести упаковку и распаковку следующих данных. Зафиксировать промежуточные значения слова состояния в процессе упаковки.

1. Физический адрес на диске представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	Н	Н	T	T	T	T	T	T	T	T	T	S	S	S	S	S

где:

НН - номер головки

Т..Т - номер дорожки

S..S - номер сектора

2. Слово состояния программы в вычислительной системе представляется в виде:

≟ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	C	C	C	C	E	I	P	0	0	0	K	K	K	0	0	0

где:

С..С - маски каналов 0 - 3

Е - маска внешнего прерывания

I - маска внутреннего прерывания

Р - маска программного прерывания

ККК - ключ зашиты памяти

3. Слово состояния канала в вычислительной системе представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	N	N	N	N	E	C	C	C	C	C	C	C	C	C	0	P

где:

N.. N - номер канала

Е - признак ошибки

С..С - код причини прерывания

- Р признак завершения программы в канале
- 4. Формат команды загрузки/сохранения в вычислительной системе имеет вид:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	1	1	1	0	0	0	1	D	R	R	R	R	A	A	A	A

где:

- D направление передачи в регистр(0)/в память(1)
- R..R 1-й операнд регистр
- А..А регистр адреса 2-го операнда
- 5. Слово состояния оборудования для вычислительной системы представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	P	P	P	0	F	F	0	Н	Н	Н	Н	0	0	V	V	V

где:

РРР - количество принтеров

FF - количество гибких дисков

Н..Н - тип жесткого диска

VVV - тип видеоадаптера

6. Формат команды сдвига в вычислительной системе имеет вид:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	1	1	0	1	T	T	D	N	N	N	N	N	N	N	N	N

где:

- ТТ тип сдвига
- D направление сдвига
- N..N количество разрядов сдвига
- 7. Блок управления буфером кеша в вычислительной системе представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	S	S	S	0	D	0	F	0	В	В	В	В	В	В	В	В

где:

- SSS код системной области или 000
- D признак "грязного" буфера
- F признак свободного буфера
- В..В номер блока, который отображен в буфере

8. Элемент списка безопасности объекта в вычислительной системе представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	U	U	U	U	U	U	U	U	R	W	Е	0	0	0	0	X

где:

- U..U идентификатор пользователя
- R право читать
- W право писать
- Е право выполнять программный код
- Х явный запрет доступа
- 9. Формат команды канала в система ввода-вывода имеет вид:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	O	O	O	O	O	C	D	N	N	N	N	N	N	N	N	N

где:

- О..О код операции
- С признак цепочки команд
- D признак цепочки данных
- N.. N количество байтов для передачи
- 10. Элемент профиля пользователя в вычислительной системе представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	G	G	G	G	G	G	G	G	S	S	S	S	P	0	0	0

где:

- G..G идентификатор группы, к которой притнадлежит пользователь
- S..S код системы, которая загружается для пользователя
- Р признак привилегированного пользователя
- 11. Информация о состоянии устройства в системе ввода-вывода представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	C	C	C	U	U	U	U	U	E	E	Е	Е	Е	E	0	В

где:

- ССС номер канала
- U..U номер устройства в канале
- Е..Е код состояния
- В признак занятости устройства

12. Формат команды сложения в вычислительной системе имеет вид:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	1	1	0	0	0	1	1	0	D	A	A	A	В	В	В	0

где:

D - сложение байтов/слов - 0/1

ААА - регистр - 1-й операнд

ВВВ - регистр - 2-й операнд

13. Формат представления текущей даты в некоторых системных структурах имеет вид:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	D	D	D	D	D	M	M	M	M	M	Y	Y	Y	Y	Y	Y

где:

D..D - день

М..М - месяц

Ү.. Ү - год после 1980

14. Формат представления текущего времени в некоторых системных структурах имеет вид:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	S	S	S	S	S	M	M	M	M	M	M	Н	Н	Н	Н	Н

где:

S..S - секунды/2

М..М - минуты

Н..Н - часы

15. Дескриптор сегмента для системы виртуальной памяти представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	F	F	F	F	F	F	F	F	R	W	L	L	L	L	L	L

где:

F..F - номер блока, с которого начинается сегмент

R - доступ для чтения

W - доступ для записи

L..L - размер сегмента в блоках

16. Атрибут файла в файловой системе представляется в виде:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	T	T	T	T	A	A	A	0	I	I	I	I	I	I	I	I

где:

Т..Т - код типа файла

ААА - код доступа

І..І - номер файлового индекса

17. Блок управления памятью в операционной системе имеет формат:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	O	O	O	O	O	O	O	O	0	P	L	L	L	L	L	L

где:

- О..О идентификатор владельца блока
- Р признак программного блока
- L..L размер блока
- 18. Точка изображения на 16-цветном дисплее с размером экрана 64х64 описывается в формате:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	X	X	X	X	X	X	Y	Y	Y	Y	Y	Y	C	C	C	C

где:

- Х..Х горизонтальная координата
- Ү.. Ү вертикальная координата
- С..С цвет
- 19. Управляющее слово программыруемого таймера имеет формат:

≟ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	C	C	F	F	D	D	D	D	D	D	D	D	D	D	D	D

где:

- СС номер канала таймера
- FF форма сигнала
- D..D коэффициент деления опорной частоты
- 20. Информация о критической ошибке на диске представляется в ви-

де:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	O	A	A	R	R	R	0	1	D	D	D	D	D	D	D	D

где:

- O тип операции чтение/запись (0/1)
- АА код области диска
- RRR возможные реакции на ошибку
- D..D номер диска

21. Дескриптор сообщения в системе телекоммуникаций имеет формат:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	N	N	N	N	0	P	P	P	U	U	U	U	U	U	U	U

где:

N..N - номер узла, из которого пришло сообщение

РРР - приоритет сообщения

U..U - идентификатор пользователя - автора сообщения

22. Дескриптор семафора, который защищает пул ресурсов, имеет формат:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	N	N	N	M	M	M	0	L	W	W	W	W	W	W	W	W

где:

N..N - текущее количество свободных единиц ресурса

М..М - общее количество единиц ресурса

L - признак блокирования/разблокирования семафора (1/0)

W..W - количество процессов, которые ожидают доступа к ресурсу

23. Заголовок кадра в системе передачи данных имеет формат:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	S	S	S	S	S	S	D	D	D	D	D	D	0	C	C	$ \mathbf{C} $

где:

S..S - адрес источника

D..D - адрес приемника

ССС - управляющий код

24. Заголовок пакета в системе передачи данных имеет формат:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	T	T	T	0	S	S	S	S	L	L	L	L	L	L	L	L

где:

ТТТ - тип пакета

S..S - идентификатор источника

L..L - длина пакета

25. Поле управления диспетчером кадра в протоколе управления логическим каналом имеет формат:

⊥ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	1	0	$\overline{\mathbf{C}}$	C	0	0	0	0	P	N	N	N	N	N	N	N

где:

СС - управляющий код

Р - признак запроса или завершения (0/1)

N.. N - номер последовательности

ЛАБОРАТОРНАЯ РАБОТА №6 ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРУКТУР

6.1 Объявление структур

Структура — это составной тип данных, в котором под одним именем объединены данные различных типов. Отдельные данные структуры называются полями. Объявление структуры осуществляется с помощью ключевого слова struct, за которым идет ее имя и далее список элементов, заключенных в фигурные скобки:

```
struct имя_структуры
{
    тип_элемента_1 имя_поля_1;
    ...
    тип_элемента_п имя_поля_п;
} имя переменной 1, имя переменной 2,...
```

Язык С позволяет задать битовые поля структуры, при этом размер этого поля указываются в битах через двоеточие за его именем:

```
тип элемента 1 имя поля 1: кол бит;
```

Правила работы с полями структуры идентичны работе с переменными соответствующих типов. К полям структуры можно обращаться через составное имя. Формат обращения:

```
имя_переменной.имя_поля
```

или

```
указатель_на_структуру- >имя_поля.
```

В языке С существует ряд ограничений на использование структур. Обязательные правила заключаются в том, что единственные операции, которые вы можете проводить со структурами, состоят в определении ее адреса с помощью операции & и доступе к одному из ее членов. Это влечет за

собой то, что структуры нельзя присваивать или копировать как целое, и что они не могут быть переданы функциям или возвращены ими. На указатели структур эти ограничения однако не накладываются, так что структуры и функции все же могут с удобством работать совместно. И наконец, автоматические структуры, как и автоматические массивы, не могут быть инициализированы; инициализация возможна только в случае внешних или статических структур.

6.2 Пример выполнения работы

Условие: создать массив структур, содержащий информацию о студентах: ФИО, номер группы, оценки за последнюю сессию. Вывести информацию о студентах гр. EMS42 в порядке убывания среднего балла.

Текст программы:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
    struct strc{
       char fio[40];
        char ngr[7];
        int otc[4];
        double sb;
   } mstud[100],stemp;
int main(void) {
    int nst, i, j;
    printf("Vvedite kol-vo studentov:");
    scanf("%i",&nst);
// Ввод данных
    for (i=0; i < nst; i++) {
      printf("Vvedite FIO:");
      scanf("%s",mstud[i].fio);
      printf("Vvedite nomer gr.:");
      scanf("%s",mstud[i].ngr);
      printf("Vvedite 4 otcenki:");
      mstud[i].sb = 0;
          (j=0; j<4; j++) {
            scanf("%i",&mstud[i].otc[j]);
            mstud[i].sb += mstud[i].otc[j]/4.;
// Упорядочивание массива
```

6.3 Индивидуальные задания

Выполнить задание в соответствии с выданным вариантом с использованием структур.

- 1. В магазине сформирован список постоянных клиентов, который включает ФИО, домашний адрес покупателя и размер предоставляемой скидки. Удалить из этого списка все повторяющиеся записи, проверив ФИО и домашний адрес покупателя.
- 2. Известны возраст и пол каждого из 20-ти человек. Найти общую массу мужчин.
- 3. Список товаров, имеющихся на складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести в алфавитном порядке список товаров, хранящихся больше месяца, стоимость которых превышает 100 000 р.
- 4. Известны данные о количестве учащихся в каждом из 15-ти учебных заведений и о типе этого заведения (школа, техникум или училище). Найти общее число учащихся школ.
- 5. Для получения места в общежитии формируется список студентов, который включает ФИО студента, группу, средний балл, доход на члена семьи. Общежитие в первую очередь предоставляется тем, у кого доход на члена семьи меньше двух минимальных зарплат, затем остальным в порядке

уменьшения среднего балла. Вывести список очередности предоставления мест в общежитии.

- 6. Известны данные о цене и тираже каждого из 15 журналов. Найти среднюю стоимость журналов, тираж которых меньше 10 000 экземпляров.
- 7. В справочной автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, тип автобуса, пункт назначения, время отправления и прибытия. Вывести информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени.
- 8. Известны стоимость и "возраст" каждой из 20-ти моделей легковых автомобилей. Найти среднюю стоимость автомобилей, "возраст" которых превышает 6 лет.
- 9. На междугородной АТС информация о разговорах содержит дату разговора, код и название города, время разговора, тариф, номер телефона в этом городе и номер телефона абонента. Вывести для заданного города общее время разговоров с ним и сумму.
- 10. Известны рост и пол каждого из 22-х человек. Найти средний рост мужчин.
- 11. Информация о сотрудниках фирмы включает ФИО, табельный номер, количество отработанных часов за месяц, почасовой тариф. Рабочее время свыше 144 часов считается сверхурочным и оплачивается в двойном размере. Вывести размер заработной платы каждого сотрудника фирмы за вычетом подоходного налога, который составляет 12 % от суммы заработка.
- 12. Известны данные о стоимости каждой из 15-ти моделей автомобилей и об их типе (легковой или грузовой). Найти среднюю стоимость легковых автомобилей.
- 13. Информация об участниках спортивных соревнований содержит наименование страны, название команды, ФИО игрока, игровой номер, возраст, рост и вес. Вывести информацию о самой молодой команде.
- 14. Известны оценки каждого из 20-ти учеников класса по двенадцати предметам. Определить среднюю оценку каждого ученика и всего класса. Вывести фамилии учеников, у которых средняя оценка выше средней по классу.

- 15. Для книг, хранящихся в библиотеке, задаются регистрационный номер книги, автор, название, год издания, издательство, количество страниц. Вывести список книг с фамилиями авторов в алфавитном порядке, изданных после заданного года.
- 16. Известны данные о массе и объеме 30-ти предметов, изготовленных из различных материалов. Определить максимальную плотность материала.
- 17. Различные цехи завода выпускают продукцию нескольких наименований. Сведения о выпущенной продукции включают наименование, количество, номер цеха. Для заданного цеха необходимо вывести количество выпущенных изделий по каждому наименованию.
- 18. Известны данные о массе и объеме 30-ти предметов, изготовленных из различных материалов. Определить материал с минимальной плотностью.
- 19. Информация о сотрудниках предприятия содержит ФИО, номер отдела, должность, дату начала работы. Вывести список сотрудников заданного отдела в порядке убывания стажа.
- 20. Известны данные о численности населения (в миллионах жителей) и площади (в тысячах квадратных километров) 28-ми государств. Определить название государства с минимальной плотностью населения.
- 21. Ведомость абитуриентов, сдавших вступительные экзамены в университет, содержит ФИО, адрес, оценки. Определить количество абитуриентов, проживающих в г. Минске и сдавших экзамены со средним баллом не ниже 4.5, вывести их фамилии в алфавитном порядке.
- 22. В справочной аэропорта хранится расписание вылета самолетов на следующие сутки. Для каждого рейса указаны номер рейса, тип самолета, пункт назначения, время вылета. Вывести все номера рейсов, типы самолетов и время вылета в заданный пункт назначения в порядке возрастания времени вылета.
- 23. У администратора железнодорожных касс хранится информация о свободных местах в поездах дальнего следования на ближайшую неделю в следующем виде: дата выезда, пункт назначения, время отправления, число свободных мест. Оргкомитет международной конференции обращается к

администратору с просьбой зарезервировать m мест до города N на k- е число со временем отправления поезда не позднее t часов вечера. Вывести время отправления или сообщение о невозможности выполнить заказ в полном объеме.

- 24. Ведомость абитуриентов, сдавших вступительные экзамены в университет, содержит ФИО абитуриента, оценки. Определить средний балл по университету и вывести список абитуриентов, средний балл которых выше среднего балла по университету. Первыми в списке должны идти студенты, сдавшие все экзамены на 10.
- 25. В радиоателье хранятся квитанции о сданной в ремонт радиоаппаратуре. Каждая квитанция содержит следующую информацию: наименование группы изделий (телевизоры, радиоприемники и т. п.), марку изделия, дату приемки в ремонт, состояние готовности заказа (выполнен, не выполнен). Вывести информацию о состоянии заказов на текущие сутки для заданной группы изделий.

ЛАБОРАТОРНАЯ РАБОТА №7 ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ФАЙЛОВ

7.1 Организация работы с файлами

Различают два вида файлов: текстовые и двоичные.

Текстовые файлы хранят информацию в виде последовательности символов. В текстовом режиме каждый разделительный символ строки автоматически преобразуется в пару (возврат каретки – переход на новую строку).

Бинарные (или двоичные) файлы предназначены для хранения только числовых значений данных. Структура такого файла определяется программно.

Функции для работы с файлами размещены в библиотеке stdio.h (#include <stdio.h>). Каждый файл должен быть связан с некоторым указателем. Этот указатель имеет тип FILE и используется во всех операциях с файлами.

Формат объявления указателя на файл следующий :

FILE * указатель на файл;

Maкрос NULL определяет пустой указатель.

Макрос **EOF**, часто определяемый как -1, является значением, возвращаемым тогда, когда функция ввода пытается выполнить чтение после конца файла.

Макрос **FOPEN_MAX** определяет целое значение, равное максимальному числу одновременно открытых файлов.

7.2 Функции для работы с файлами

Функция

FILE *fopen (const char * имя_файла, const char * режим_открытия);

открывает файл и связывает его с потоком. Возвращает указатель на открытый файл. **Имя_файла** — это указатель на строку символов, в которой хранится имя файла и путь к нему. **Режим_открытия** — это указатель на строку символов, в которой указывается режим открытия файла. Допустимые режимы:

r – открытие текстового файла для чтения;

w – с оздатие текстового файла для записи;

а – добавление информации в конец текстового файла.

При работе с текстовыми файлами к символу, указывающему режим открытия, добавляется символ « t » (по умолчанию), а при работе с бинарными – «b». Если необходимо и читать и записывать в файл, то добавляется символ «+». Так, для открытия бинарного файла на чтение и изменение надо:

fopen ("data.bin", "rb+");

При возникновении ошибки во время открытия файла, функция fopen возвращает значение NULL.

Функция

int fclose (FILE *указатель_ на _ файл);

закрывает поток, который был открыт с помощью вызова fopen() и записывает в файл все данные, которые еще оставались в дисковом буфере. Доступ к файлу после выполнения функции будет запрещен.

Возвращение нуля означает успешную операцию закрытия. В случае же ошибки возвращается EOF.

Функция

int fcloseall(void);

закрывает все открытые файлы . Возвращает количество закрытых файлов или EOF, если возникает ошибка.

Функция

int putc(int символ, FILE * указатель _ на _ файл);

записывает один символ в текущую позицию указанного открытого файла.

Функция

int getc(FILE * указатель на файл);

читает один символ из текущей позиции указанного открытого файла.

Функция

int feof (FILE * указатель на файл);

возвращает отличное от нуля значение (true), если конец файла не достигнут, и ноль (false), если достигнут конец файла.

Функция

int fputs (const char * строка, FILE * указатель_ на _ файл); записывает строку символов в текущую позицию указанного открытого файла.

Функция

char * **fgets** (**char** * **строка**, **int длина**, **FILE** * **указатель_на_файл**); читает строку символов из текущей позиции указанного открытого файла до тех пор , пока не будет прочитан символ перехода на новую строку , или количество прочитанных символов не станет равным длина -1.

Функция

int * fprintf (FILE * указатель_на_файл, const char * управляющая_строка);

записывает форматированные данные в файл. Управляющая_строка определяет строку форматирования аргументов, заданных своими адресами. Обычно эта строка состоит из последовательности символов «%», после которых следует символ типа данных:

I или i – десятичное, восьмеричное или шестнадцатеричное целое;

D или d – десятичное целое;

U или u – десятичное целое без знака;

Е или е – действительное с плавающей точкой;

s - строка символов;

с – символ.

Функция

int * fscanf (FILE * указатель_на_файл, const char * управляющая строка);

читает форматированные данные из файла. Строка форматирования строится аналогично функции fprintf .

Функция

void rewind(FILE * указатель_на_файл);

устанавливает указатель текущей позиции выделенного файла в начало файла.

Функция

int ferror (FILE * указатель_ на _ файл);

определяет, произошла ли ошибка во время работы с файлом. Функция

size_t fwrite (const void * записываемое_данное, size_t размер элемента,

size_t число_элементов, FILE * указатель_на_файл);

записывает в файл заданное число данных определенного размера. Размер данных задается в байтах . Тип size_t определяется как целое без знака.

Функция

size_t fread (void * считываемое_данное, size_t размер_элемента, size_t число_элементов, FILE *указатель_на_файл);

считывает из файла указанное число данных заданного размера. Размер задается в байтах . Функция возвращает число прочитанных элементов. Если число прочитанных элементов не равно заданному, то при чтении возникла ошибка или встретился конец файла.

Функция

int fileno (FILE * указатель_ на _ файл);

возвращает значение дескриптора указанного файла (дескриптор – логический номер файла для заданного потока).

Функция

long filelength (int дескриптор);

возвращает длину файла с соответствующим дескриптором в байтах.

Функция

int fseek (FILE * указатель_на_файл, long int число_байт, int точ-ка_отсчета);

устанавливает указатель в заданную позицию. Заданное количество байт отсчи-тывается от позиции , которая задаётся следующими макросами: SEEK_SET — начало файла, SEEK_CUR — текущая позиция, SEEK_END — конец файла.

7.2 Пример выполнения работы

Условие: считать текст из файла и выделить часть из него с помощью функции. Результат отправить в другой текстовый файл.

Функция выделяет из заданной строки подстроку заданной длины, начиная с заданной позиции.

Текст программы:

```
#include <stdio.h>
#include <locale.h>
/*** функция выделения подстроки ***/
/* параметры: src - строка-источник
              dest - строка-результат
              pos - позиция, с которой выделяется подстрока
              len - длина подстроки */
void substr(char *src, char *dest, int pos, int len) {
  for(;pos&&(*src);pos--) src++; // перемещаемся на начало
  for(;len&&(*src);len--)
    *dest++=*src++;
                                 // премещаем символы
  *dest=0;
                                 // признак конца строки
int main(int an, char *av[]) {
    setlocale(LC_ALL, "Russian");
    FILE *inf, *outf; /* файловые переменные */
    int start, /* начальная позиция подстроки */
                 /* длинна подстроки */
    char filename[80]; /* имя файла */
    char bu1[81], bu2[81]; /* входной и выходной буферы */
```

```
printf("Введите имя файла:");
   scanf("%s",filename);
/* открытие файлов */
   if ((inf=fopen(filename, "r")) == NULL) {
        printf("Невозможно открыть файл: %s\n",filename);
        return(0);
   outf=fopen("TMP.txt","w");
/* параметры обработки */
   printf("Введите начало подстроки:");
   scanf("%i",&start);
   printf("Введите длинну подстроки:");
   scanf("%i",&len);
/* чтение фала строка за строкой */
   for (;fgets(bu1,80,inf)!=NULL; ) {
        substr(bu1, bu2, start, len);
                                        // выделяем подстроку
        fprintf(outf,"%s\n",bu2);
   fclose(inf);
   fclose(outf);
   return 0;
}
```

7.3 Индивидуальные задания

Составить программу, которая читает текстовый файл и выполняет обработку его строк функцией, заданной в Вашем варианте индивидуального задания. Использовать функции библиотеки string.h нельзя. Результат выводить в текстовый файл.

- 1. Функция дописывает опорную строку в конец другой.
- 2. Функция выделяет первое слово из строки.
- 3. Функция переводит строку, содержащую десятичное представление целого числа, в строку, содержащую его шестнадцатиричное представление..
- 4. Функция выравнивает строку по правому краю до заданной длины. Длинна определяется самой длинной строкой в исходном файле.
- 5. Функция заменяет в строке все множественные вхождения заданного символа одним.
- 6. Функция подсчитывает количество слов в строке. Результат дописать в начало строки.

- 7. Функция удаляет из строки заданное количество символов, начиная с заданной позиции.
- 8. Функция определяет в строке номер позиции слова с заданным номером. Результат дописать в начало строки.
 - 9. Функция переписывает все символы строки в обратном порядке.
- 10. Функция доводит длину строки до заданной, вставляя пробелы между словами.
 - 11. Функция дописывает опорную строку в начало другой.
 - 12. Функция удаляет из строки слово с заданным номером.
- 13. Функция перекрывает символы строки, символами опорной строки, начиная с заданной позиции.
- 14. Функция заменяет в строке одну заданную комбинацию символов другой заданной комбинацией.
- 15. Функция удаляет из начала и из конца строки заданное количество символов.
 - 16. Функция выделяет из строки заданное количество слов.
- 17. Функция выравнивает строку по правому краю до заданной длины пробелами в начале.
- 18. Функция вставляет в строку опорную строку, начиная с заданной позиции.
- 19. Функция разбивает строку на две части: до первого вхождения заданного символа и после него
- 20. Функция находит в строке первый символ, который входить в другую заданную строку. Номер символа дописать в начало строки.
- 21. Функция копирует строку в другую строку заданной длины и размещает текст первой строки по центру второй строки.
- 22. Функция находит в строке первый символ, который не входить в другую опорную строку. Номер символа дописать в начало строки.
- 23. Функция заменяет символы строки из одного заданного алфавита на символы другого алфавита.
- 24. Функция определяет длину слова с заданным номером. Результат дописать в начало строки.

25. Функция копирует строку в другую строку заданное количество раз.

ЛАБОРАТОРНАЯ РАБОТА №8 ПРОГРАММИРОВАНИЕ С ПРИМЕНЕНИЕМ ДИНАМИЧЕСКОГО СПИСКА

8.1 Односвязный список

Односвязный список – структура данных, в которой каждый элемент (узел) хранит информацию, а также ссылку на следующий элемент. Последний элемент списка ссылается на NULL.

Односвязный список полезен когда:

- 1) Начальный размер данных не определен и может меняться в широком диапазоне.
- 2) Информация требует сортировки одновременно по нескольким параметрам.
 - 3) Необходимо построение рекурсивных алгоритмов.
- 4) Когда размер данных или обычного массива определяется расчетом во время выполнения программы.

Для простоты рассмотрим односвязный список, который хранит целочисленное значение (рис. 8.1). Такой список состоит из узлов. Каждый узел содержит значение и указатель на следующий узел, поэтому представим его в качестве структуры

```
struct tstk
{ int inf;
tstk *a; } spt.
```

Для добавления нового узла необходимо

- 1) Выделить под него память.
- 2) Задать ему значение
- 3) Сделать так, чтобы он ссылался на предыдущий элемент (или на NULL, если его не было)
 - 4) Перекинуть указатель head на новый узел.

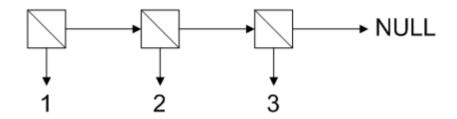


Рисунок 8.1 Односвязный список

Для выделения памяти используется команда new

*spt=new tstk;

Для освобождения памяти используется команда delete

delete spt;

Проще всего работать со списком типа стек, в котором заполнение происходит с конца в начало.

8.2 Функции работы со стеком

```
Добавление элемента в стек

tstk *AddStask( tstk *sp, int inf)

{ tstk *spt;
    spt = new tstk;
    spt->inf = inf;
    spt->a = sp;
    return spt; }

Удаление всего стека

void DelStackAll(tstk *sp)

{ tstk *spt;
    while(sp != NULL) {
    spt = sp;
    sp = sp->a;
    delete spt; }
}
```

Обмен текущего элемента со следующим за ним

```
void RevStackAfter(tstk *sp)
{ int tmp = sp->inf;
sp->inf = sp->a->inf;
sp->a->inf = tmp;
}
```

8.3 Пример выполнения работы

Условие: посчитать среднее арифметическое первой половины элементов списка.

```
Текст программы:
```

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
struct tstk
 { int inf;
   tstk *a;}*sp,
tstk *sp_s;
tstk *AddStask( tstk *sp, int inf)
{ tstk *spt;
spt = new tstk;
 spt->inf = inf;
spt->a = sp;
return spt; }
void DelStackAll(tstk *sp)
{ tstk *spt;
while(sp != NULL) {
spt = sp;
 sp = sp->a;
delete spt; }
int main(void) {
    int i,n;
    float s;
    srand(time(NULL)); // инициализация rand
                       // список пустой
    sp=NULL;
    n=rand()%11+10; // размер списка
    for (i=0;i<n;i++)
       sp = AddStask(sp, rand()%101-50);
                         // сохранить начало очереди
    sp s=sp;
/* печать списка */
```

8.4 Индивидуальные задания

Создать стек с числами в диапазоне от -50 до +50 случайного размера от 10 до 20 элементов. После создания стека выполнить индивидуальное задание. В конце работы все стеки должны быть удалены.

- 1. Создать новый стек, содержащий только отрицательные числа из первого стека.
- 2. Создать новый стек, содержащий только четные числа из первого стека.
- 3. Создать новый стек, содержащий только те числа из первого стека, которые больше среднего значения всех элементов первого стека.
- 4. Создать новый стек, содержащий только положительные числа из первого стека.
- 5. Подсчитать, сколько элементов стека имеют значение, которое превышает среднее значение всех элементов стека.
 - 6. Найти максимальное и минимальное значения стека.
- 7. Определить, сколько элементов стека, начиная с вершины, находится до элемента с максимальным значением.
- 8. Определить, сколько элементов стека, начиная от вершины, находится до элемента с минимальным значением.
- 9. Определить, сколько элементов стека находится между его минимальным и максимальным элементами.
 - 10. Поменять местами соседние элементы исходного списка.
- 11. Создать новый стек, содержащий только числа, большие первого элемента основного стека.

- 12. Преобразовать стек в два стека. В первый поместить все четные, а во второй все нечетные числа.
- 13. Создать новый стек, порядок следования элементов в котором обратный относительно первого стека.
- 14. Создать новый стек, в который поместить каждый третий элемент первого стека.
- 15. Создать новый стек, в который поместить элементы, лежащие во второй половите первого стека.
 - 16. Посчитать количество нечетных чисел стека.
- 17. Создать новый стек, содержащий только те числа из первого стека, которые меньше среднего значения всех элементов первого стека.
 - 18. Посчитать количество нулевых элементов списка.
- 19. Создать новый стек, в который поместить каждый второй элемент первого стека.
- 20. Создать новый стек, содержащий только числа, меньшие первого элемента основного стека.
- 21. Подсчитать, сколько элементов стека имеют значение, которое меньше среднего значения всех элементов стека.
- 22. Посчитать среднее арифметическое второй половины элементов списка.
 - 23. Посчитать количество отрицательных элементов списка.
- 24. Создать новый стек, содержащий только нечетные числа из первого стека.
 - 25. Удвоить значения элементов стека.

ЛАБОРАТОРНАЯ РАБОТА №9 ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ МАТРИЦ

9.1 Многомерные массивы

Многомерные массивы - это массивы с более чем одним индексом. Чаще всего используются двумерные массивы. При описании многомерного массива необходимо указать С, что массив имеет более чем одно измерение.

Пример 1.

int t[3][4];

Описывается двумерный массив, из 3 строк и 4 столбцов.

Элементы массива:

t[0][0]	t[0][1]	t[0][2]	t[0][3]
t[1][0]	t[1][1]	t[1][2]	t[1][3]
t[2][0]	t[2][1]	t[2][2]	t[2][3].

При выполнении этой команды под массив резервируется место. Элементы массива располагаются в памяти один за другим:

Для резервирования места под 3-х мерный массив записывается:

int temp [3] [15] [10];

В памяти многомерные массивы представляются как одномерный массив, каждый из элементов которого, в свою очередь, представляет собой массив.

Инициализация многомерных массивов производится по аналогии с одномерными.

На пример двумерный массив:

Второй способ инициализации при описании массива

int
$$a[3][2]=\{\{4,1\},\{5,7\},\{2,9\}\};$$

9.2 Пример выполнения работы

Условие: Заполнить секторы матрицы, которые лежат выше и ниже главной и побочной диагоналей, ЛП, от левого верхнего угла вниз - вправо. Остаток матрицы заполнить нулями.	0 0 0 0	1 0 0 0 0 0 0	-7	9 0 0 0 10	13 14 15 16 0 17 18	23 0 0 0 24 25	0 0 0	31 0 0 0 0 0 0		
---	---------	---------------------------------	----	------------------------	---------------------------------------	-------------------------------	-------	----------------------------------	--	--

Текст программы:

```
#include <stdio.h>
#define S 9
int Ar[S][S]; /* матрица */
int main(void) {
 short l, r; /* текущие индексы */
short r1, r2; /* граничные номера столбцов */
short dd; /* модификатор граничных номеров */
short k; /* текущий член ЛП */
  /* начальные значения переменных */
 r1=1; r2=S-2; dd=1; k=1;
  for (l=0; l<S; l++) { /* перебор строк */
    for (r=0; r<S; r++) /* перебор столбцов */
      /* условие ненулевого значения */
      if ((r<r1)||(r>r2)) Ar[1][r]=0;
      else Ar[1][r]=k++;
      /* конец перебора строк */
    /* модификация границ */
    r1+=dd; r2-=dd;
    /* уловие перехода в нижнюю часть */
    if (r1>r2) dd=-dd;
    } /* конец перебора столбцов */
  /* вывод матрицы */
  for (1=0; 1<S; 1++) {
    for (r=0; r<S; r++) {
      printf("%3d",Ar[1][r]);
    printf("\n");
 return 0;
```

9.3 Индивидуальные задания

Создать квадратную матрицу целых чисел размером 9х9. В индивидуальных заданиях указано, какую обработку матрицы требуется выполнить.

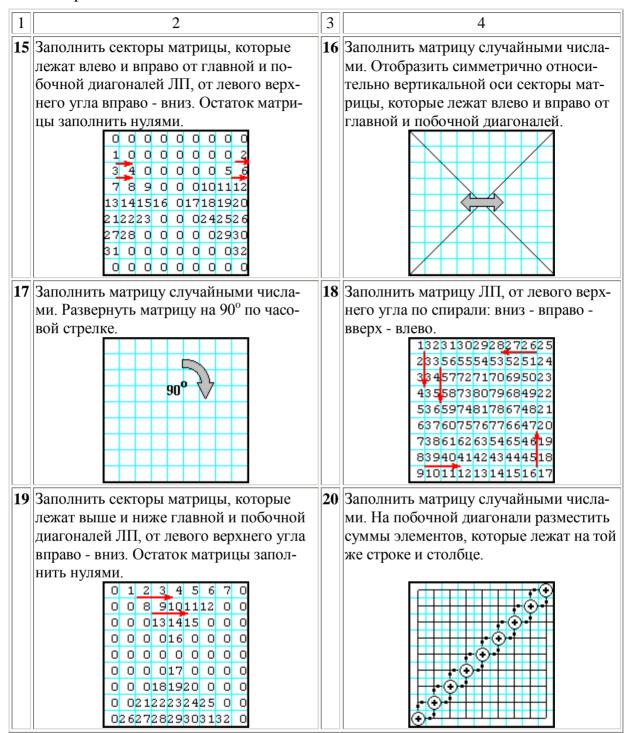
Таблица 9.1 – Индивидуальные задания

N.C.	1 аолица 9.1 — индивидуальные задани		2
№	Задание	No	Задание
1	2	3	4
1	Заполнить матрицу случайными числа-	2	Заполнить матрицу случайными числа-
	ми. Отобразить матрицу симметрично		ми. Отобразить матрицу симметрично
	относительно побочной диагонали		относительно главной диагонали
3	Заполнить матрицу ЛП, от левого верх-	4	Заполнить матрицу ЛП, от центра по
	него угла по спирали: вправо - вниз -		спирали: влево - вниз - вправо - вверх. 818079787776757473
	влево - вверх. 1 2 3 4 5 6 7 8 9		504948474645444372
	3233343536373839 L O		512625242322214271
	31 <mark>55575859606140</mark> 11		522710 9 8 7204170
	305572737475624112		532811 2 1 6194069 542912 3 4 5183968
	295471808176634213 285370797877644314		553013141516173867
	275269686766654415		563132333435363766
	26 <mark>5150494847</mark> 4645 <mark>16</mark>		575859606162636465
Ш	252423222120191817		
5	Заполнить матрицу случайными числа-	6	Заполнить секторы матрицы, которые
	ми. На главной диагонали разместить		лежат влево и вправо от главной и по-
	суммы элементов, которые лежат на той		бочной диагоналей, ЛП, от левого верх-
	же строке и том же столбце.		него угла вниз - вправо. Остаток матри-
			цы заполнить нулями.
	I		1,0000000000
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		2 8 0 0 0 0 02127
			3 9 13 0 0 0 182228
	Ψ		410 <mark>1416 017192329</mark> 51115 0 0 0202430
			612 0 0 0 0 02531
	<u> </u>		7 0 0 0 0 0 0 032
			0 0 0 0 0 0 0 0
7	Заполнить матрицу ЛП, от левого верх-	8	Заполнить матрицу ЛП, от левого ниж-
	него угла по диагонали: вправо - вверх.		него угла по диагонали: влево - вверх.
	1 3 6101521283645		455360667175788081
	2 5 9142027354453		364452596570747779
	4 813192634435260 71218253342515966		283543515864697376 212734425057636872
	111724324150586571		152026334149566267
	162331404957647075		101419253240485561
	223039485663697478		6 913182431394754
	293847556268737780 374654616772767981		3 5 8121723303846 1 2 4 71116222937
	P dadadotto (1710 talon		1 6 4 (1771105553)

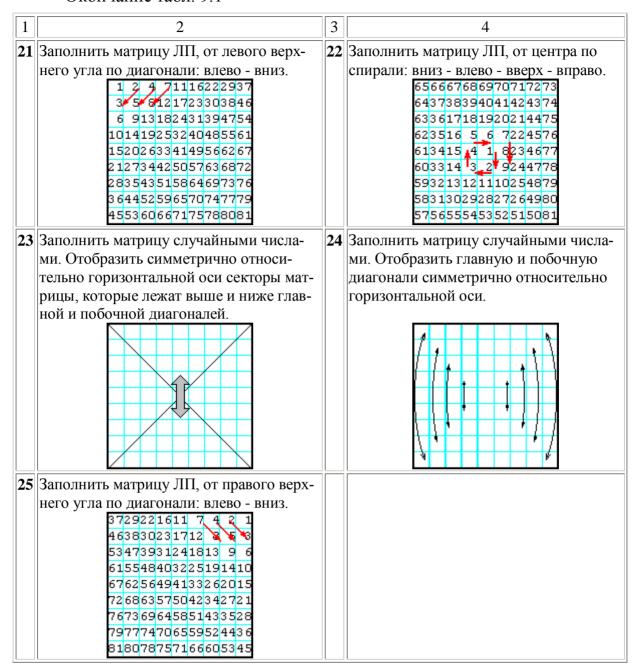
Продолжение табл. 9.1

1 3 9 Заполнить матрицу случайными числа-10 Заполнить матрицу случайными числами. Развернуть матрицу на 90° против ми. Отобразить главную и побочную часовой стрелки. диагонали симметрично относительно вертикальной оси. 11 Заполнить матрицу случайными числа-12 Заполнить матрицу случайными числами. Разместить на главной диагонали ми. Отобразить верхнюю половину матсуммы элементов, которые лежат на диарицы на нижнюю зеркально симметгоналях, перпендикулярных к главной. рично относительно горизонтальной оси. (•) (•) Œ 13 Заполнить матрицу случайными числа-14 Заполнить матрицу случайными числами. Розбити матрицу на квадраты размеми. Отобразить правую половину матром 3х3. В центре каждого квадрата порицы на левую зеркально симметрично местить сумму остальных элементов относительно вертикальной оси. квадрата. ١ 1

Продолжение табл. 9.1



Окончание табл. 9.1



10 РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ. ОБРАБОТКА ДАННЫХ ИЗ ТЕКСТОВОГО ФАЙЛА

10.1 Постановка задачи

Целью работы является построение векторного изображения по зашифрованным данным. Для достижения поставленной цели необходимо сохранить данные задания в текстовый файл и преобразовать эти данные с помощью программы, которая соответствует вашему варианту задания.

Программа выполняет следующие действия:

- считывает исходные данные из текстового файла;
- по этим данным, при необходимости, находит вспомогательные величины (среднее, максимум и т.д.);
- преобразует исходные данные в координаты точек искомого изображения по выражениям, которые указаны в индивидуальном задании;
- сохраняет полученные координаты в новый текстовый файл и выводит контрольные значения на экран.

Для проверки правильности расчета, в задании даны контрольные точки по нескольким значениям полученных координат X и обобщенное значение полученных координат Y. В программе необходимо предусмотреть вывод указанных значений на экран в процессе расшифровки исходных данных. При этом обобщенное значение координат Y рассчитывается в программе отдельно.

При совпадении контрольных значений с заданием, по координатам, которые записаны в файл, строится изображение в программе Exel. Для этого в программе Exel открывается файл с обработанными координатами как текстовый с разделителями «пробел».

Для получения изображения выделяем значения в столбиках, переходим во вкладку «Вставка», выбираем пункт «Диаграммы — Точечная — Точечная с прямыми линиями» и получаем график с изображением. На диаграмме оставляем только оси X и Y и переносим его в отчет.

10.2 Требования к отчету

Отчет выполняется индивидуально на вертикальных листах формата А4 и содержит следующие пункты.

- 1. Титульный лист;
- 2. Исходные данные и условия задания;
- 3. Блок-схема алгоритма программы;
- 4. Текст программы;

- 5. Полученные координаты точек изображения;
- 6. Диаграмма, выполненная в программе Microsoft Excel.

Титульный лист размещается на отдельном листе в соответствии с примером, показанном в приложении І. Алгоритм выполняется с учетом действующих требований к нему (см. приложение II).

10.3 Пример выполнения работы

Данные задания предварительно должны быть сохранены в текстовый файл in.txt. Создать текстовый файл можно либо средствами Windows либо в любом файл-менеджере.

В конце, программа записывает полученные координаты точек x и y в текстовый файл с именем out.txt.

Условие:

Преобразовать x по следующему выражению:

$$x = \begin{cases} x/(k+1), \ 0 \le k \le 9; \\ x-k-1, \ 10 \le k \le 19; \\ x-(k+1)^2, \ 20 \le k \le 29. \end{cases}$$

При этом k — порядковый номер числа.

Значение у преобразовать по следующему выражению:

$$y = y - \overline{X}/(k+1)^2,$$

где \overline{X} - среднее значение исходной переменной x.

Контрольные значения расчетных координат x:

$$k = 5$$
 $x = -14$
 $k = 18$ $x = 19$
 $k = 23$ $x = 6$

Среднее значение полученных координат y = 3.60.

Исходные данные для расчета приведены в таблице 10.1.

Таблица 10.1 – Исходные данные

k	X	y	k	X	y	k	X	y
0	-5	208.9667	10	1	3.693939	20	460	4.464777
1	-14	55.24167	11	3	2.42338	21	488	2.423485
2	-27	28.77407	12	4	1.212821	22	531	3.387461
3	-44	18.81042	13	6	-0.95425	23	582	9.355845
4	-60	13.19867	14	15	-2.08904	24	635	11.32795
5	-84	10.69352	15	19	-1.19935	25	683	11.30321
6	-84	8.182993	16	22	-1.29077	26	734	11.28116
7	-112	6.202604	17	28	-1.36739	27	787	11.26144
8	-108	5.530453	18	38	-1.43223	28	842	10.24372
9	-110	4.049667	19	24	0.512417	29	895	4.227741

Текст программы:

```
#include <stdio.h>
int main(void) {
  FILE *inf,*outf;
                                  // файловые переменные
  int i;
  float x[30], y[30], s=0; // массивы исходных данных,
среднее
  inf = fopen("in.txt", "rt"); // открытие файла для чтения
  for (i = 0; i < 30; i++) {
  fscanf(inf, "%f %f", &x[i], &y[i]);
  s=s + x[i];
                             // расчет среднего арифметическо-
ГΟ
  s=s / 30;
/// Преобразование исходных данных
  for (i = 0; i < 30; i++) {
        if (i < 10) x[i] = x[i] / (i + 1); // коррекция <math>x
           else
            if (i < 20) x[i] = x[i] - i-1;
            else x[i] = x[i] - (i + 1) * (i + 1);
        y[i] = y[i] - s / ((i + 1) * (i + 1)); // коррекция у
/// Сохраняем координаты в файл
  s = 0;
  outf = fopen("out.txt", "wt"); // открытие файла для записи
  for (i = 0; i < 30; i++) {
    fprintf(outf, "%.1f %.1f\n", x[i], y[i]); // запись в
файл
   s += y[i];
                                   // расчет контрольной точки
/// Вывод контрольных значений
 printf (" For k=5 x=%.1f\n", x[5]); // контрольная точка 1
 printf (" For k=18 x=%.1f\n", x[18]); // контрольная точка 2
```

```
printf (" For k=23 x=%.1f\n",x[23]); // контрольная точка 3
  printf (" Average y:%.2f\n",s/30); // среднее значение ко-
ординат Y

fclose(inf); // закрытие файлов
  fclose(outf);
return 0;
}
```

По данным файла out.txt строится точечная диаграмма в Microsoft Excel и удаляются лишние данные. В результате построения получается изображение объекта (см. рис. 10.1), название которого необходимо указать в подрисуночной надписи.

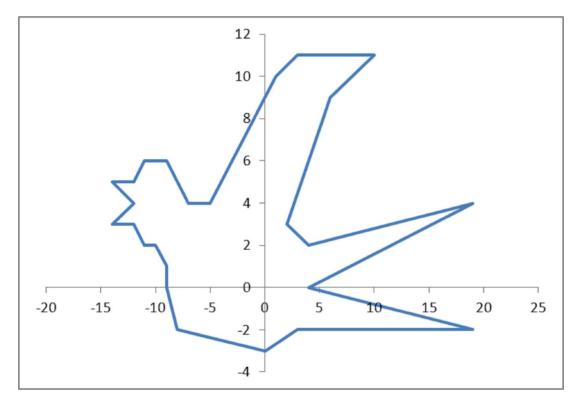


Рисунок 10.1 – Изображение голубя, построенное по данным задания

10.4 Индивидуальные задания

	Вариант № 1			Вариа	ант № 2	Вариант № 3				
k	X	у	k	X	у	k	X	y		
0	-4	42.3	0	1	-8	0	-210	-3		
1	-1.5	21.65	1	0	-7	1	-84	-0.75		
2	-1	14.77	2	-1	-3	2	-36.83	-0.444		
3	-1	11.82	3	-3	7	3	-12	0.5625		
4	-1	9.66	4	-4	16	4	-4.4	0.8		
5	-0.666	8.22	5	-6	26	5	5	1.111		
6	-0.428	7.04	6	-10	36	6	7	2.122		
7	-0.12	6.29	7	-17	49	7	6	1.641		
8	0	5.59	8	-20	65	8	7	2.099		
9	-0.2	4.93	9	-22	85	9	6	1.65		
10	-11	4.48	10	-25	105	10	7	0.736		
11	-11	4.11	11	-27	127	11	5	-1		
12	-15	3.72	12	-30	149	12	8	-4.136		
13	-15	3.38	13	-33	174	13	14	-5.770		
14	-14	3.15	14	-35	201	14	18	-5.422		
15	-13	2.89	15	-270	228	15	18	-11.57		
16	-13	2.61	16	-288	258	16	16	-16.58		
17	-14	2.41	17	-272	290	17	22	-31.90		
18	-14	2.33	18	-216	325	18	28	-46.33		
19	-16	2.11	19	-228	358	19	31	-50.74		
20	-96	2.01	20	-240	394	20	35	-124.1		
21	-103	1.92	21	-294	434	21	37	-292.7		
22	-110	1.84	22	-308	476	22	36	-541.7		
23	-117	1.76	23	-535	521	23	1576	-805		
24	-124	1.69	24	-580	569	24	1715	-1320		
25	-127	1.67	25	-629	622	25	1862	-2334		
26	-130	1.60	26	-675	673	26	2032	-3503		
27	-137	1.58	27	-727	723	27	2192	-4864		
28	-146	1.46	28	-783	777	28	2366	-7337		
29	-153	1.41	29	-840	833	29	2537	-2773		
	$\int x \cdot (k+1)$,	$0 \le k < 10;$		$\int x + 2 \cdot k$, ($0 \le k < 15;$		$\int x \cdot (k+1)/(k+1)$	$(x-15), 0 \le k < 5;$		
x =	$\begin{cases} x+k, & 10 \le 1 \end{cases}$	k < 20;	x =	$\begin{cases} x/(2 \cdot k). \end{cases}$	$15 \le k < 23$:	x =	$\begin{cases} x-2\cdot k+6, \end{cases}$	$5 \le k < 23;$		
$(x+5\cdot k)/2, 20 \le k < 30.$				$\begin{vmatrix} x + (2 & x), \\ x + k^2, & 23 \end{vmatrix}$	$15 \le k < 23;$ $6 \le k < 30.$		$\left(x-3\cdot k^2, 2\right)$	$3 \le k < 30.$		
$y = ((y+1) \cdot k + X_{AV})$		$\cdot k + X_{AV}$)		y = (y + X)			$y = (y \cdot (k$	· · · · · · · · · · · · · · · · · · ·		
где 2	где $X_{\scriptscriptstyle AV}$ - среднее значение		гле	X_{max} - макс				овательность		
	исходной переменной х						,	, 1, 2, 3, 5)		
	Контрольные точки:			TC			Контрольные точки: k=2 x = 8.5			
k=0 x = 3			k=1 $x=2$			k=2 x = 6.3 k=12 x = -10				
	k=10 x = -3 k=20 x = -3			k=16 x = -9			k=12 x = -10 k=22 x = -2			
	x = -3 enee 3hauehi	we $Y_{AV} = 0.85$		5 x = -4	V 0.70			ие $Y_{AV} = -3.23$		
СРСД	опос эпачені	1 AV - 0.03	Сред	ценее значен	ие $Y_{AV} = -0.73$					

	Вариа	нт № 4		Вариан	т № 5		Вариаг	нт № 6		
k	X	У	k	X	У	k	X	у		
0	0.5	-209.2	0	-3	0.216	0	-8.5	-13.25		
1	3	-103.6	1	1	2.432	1	-10.5	-3.312		
2	5	-68.7	2	-3	2.65	2	-12.5	-10.31		
3	8	-51.29	3	0	4.865	3	-13.5	-5.797		
4	11	-41.13	4	7	5.581	4	-16	0		
5	13.5	-34.19	5	18	6.297	5	-18.5	0		
6	14.5	-29.17	6	33	6.513	6	-20	-1.081		
7	16.5	-25.4	7	2.646	9.73	7	-22	-0.828		
8	18	-22.52	8	-8.48	8.946	8	-24	-0.654		
9	20	-20.06	9	-15	10.16	9	-26.5	0		
10	21	-18.38	10	-9.487	6.38	10	1.5	0		
11	22.5	-16.89	11	-19.9	7.59	11	1.5	0		
12	23.5	-15.55	12	-10.39	5.81	12	1	0.2352		
13	-0.02	-14.51	13	-18.03	5.03	13	0	0.2028		
14	-0.035	-13.58	14	-18.71	4.24	14	-0.5	0.2355		
15	-0.1	-12.88	15	-19.36	3.46	15	-1.5	0.2588		
16	-0.125	-12.24	16	-20	2.67	16	-3	0.2292		
17	-0.132	-11.7	17	-20.62	1.89	17	-4	0.1635		
18	-0.11	-11.14	18	-8.48	1.11	18	-4.5	0.1101		
19	-0.118	-10.63	19	-17.43	0.325	19	-6	0.0993		
20	396.5	-10.2	20	-13.42	0.541	20	41.25	0.0901		
21	440	-9.78	21	-9.165	0.758	21	43	0.0821		
22	485	-9.4	22	-0.014	0.974	22	44	0.0751		
23	532.5	-9.09	23	0	1.19	23	46	0.092		
24	580.5	-8.73	24	0.019	1.4	24	47.2	0.106		
25	629.5	-8.31	25	0.0375	2.62	25	48.85	0.098		
26	679	-7.93	26	0.0482	3.84	26	50.74	0.0727		
27	729	-7.63	27	0.058	6.05	27	52.74	0.0507		
28	785	-7.26	28	0.067	7.27		54.48	0.0473		
29	841.5	-6.97	29	0.033	6.49	29	56.48	-0.0147		
	$\int x - 2 \cdot k$,	$0 \le k < 13;$		$\int x/(2\cdot k-1)$), $0 \le k < 7$;		$\int x+5+3\cdot k$	$0 \le k < 14;$		
x	$=\begin{cases} x \cdot 2 \cdot k, \end{cases}$	$13 \le k < 20;$	<i>x</i> =	$= \begin{cases} x/\sqrt{k}, & 7 \end{cases}$	$\leq k < 22;$	<i>x</i> =	$= \begin{cases} x + (k-1)/ \end{cases}$	2, $14 \le k < 22$;		
		$20 \le k < 30.$		$x \cdot (3 \cdot k + 5)$), $22 \le k < 30$.		$(x-2\cdot k)^3,$	$22 \le k < 30$.		
	$y = y \cdot (k + 1)$	$-1) \perp Y$	J	$v = v + X_{MAY}$	$\cdot (k+1)/S_{v}$		$y = y \cdot (k +$.,,		
гле	где X_{AV} - среднее значение			$X_{ m MAX}$ - макси	мальное значе-	где	X_{MIN} — МИН	имальное		
	исходной переменной x			исходной х		знач	чение исхо	однои х		
	окодной переменной х Сонтрольные точки:			-	, ,		трольные т x = 1.5	очки:		
	=3 x = 2			nop ememority			x = 1.3 5 x = 5.5			
k=13	=13 x = -0.5			10111 0011211210 10 111111			k=13 x = 3.3 k=25 x = -1.5			
	x = 23 x = 3.5			k=14 x = -5			Сумма значений Y= -17			
Сред	енее значен	ие $Y_{AV} = 0.57$		4 x = 1	**					
<u> </u>			Cpe	денее значен	ие $Y_{AV} = 0.88$					

	Вариант № 7			Вариан	т № 8	Вариант № 9				
k	X	у	k	X	y	k	X	y		
0	0.0009	-676.14	0	-70	0.0093	0	-18	-44.0		
1	0.2231	-666.14	1	-57	0.0093	1	-17	-14.4		
2	0.7165	-662.14	2	-44	0.0154	2	-15.59	-8.556		
3	1.649	-646.14	3	-31	0.0347	3	-16.27	-3.850		
4	1.492	-654.14	4	-23	0.0509	4	-15	-2.800		
5	2.076	-650.14	5	-25	0.0308	5	-8.76	-1.069		
6	2.698	-653.14	6	-7	0.0086	6	-6.55	0		
7	3.341	-658.14	7	6	0.0182	7	-32	0.688		
8	4.000	-683.14	8	64	0.0175	8	-49.5	3.422		
9	4.673	-724.14	9	77	0	9	-60	3.850		
10	5.358	-773.14	10	85	0.0748	10	-77	4.118		
11	3.054	-770.14	11	93	0.0555	11	-60	3.565		
12	-0.241	-804.14	12	1.5	0	12	-32.5	1.247		
13	1.471	-744.14	13	1.442	-0.12	13	-28	2.750		
14	-2.809	-704.14	14	1.464	-0.94	14	-7.5	2.444		
15	16.94	-706.14	15	1.467	-1.28	15	-8	0.438		
16	-17.94	-708.14	16	1.469	-4.78	16	51	0.788		
17	-56.84	-710.14	17	1.353	-4.32	17	45	1.782		
18	-99.75	-712.14	18	1.306	-8.14	18	9.28	1.621		
19	-146.67	-714.14	19	1.211	-7.82	19	11.26	0.592		
20	0	-674.14	20	0.875	-14.47	20	12.25	1.086		
21	0	-630.14	21	490	-26.06	21	16.24	1.000		
22	47.92	-651.14	22	547	-40.34	22	15.23	0.231		
23	74.88	-674.14	23	620	-62.55	23	16.22	-0.214		
24	0.0	-674.14	24	681	-83.28	24	13.21	-0.795		
25	-107.9	-674.14	25	709	-118.8	25	5.2	-0.833		
26	-111.9	-647.14	26	739	-201.9	26	-4.81	-0.778		
27	-86.9	-646.14	27	806	-52.50	27	-9.81	-0.566		
28	-89.9	-674.14	28	875	0	28	-13.82	-0.455		
29	-123.9	-674.14	29	939	256.56	29	-16.83	-0.285		
1 1	$(k+1)\cdot \ln x, 0 \le k$			$\int (x-3\cdot k)/k$	5, $0 \le k < 12$;		$\int x - \sqrt{k}$, 0	$\leq k < 7;$		
$x = \begin{cases} x \end{cases}$	$+\sqrt[3]{k^2}$, $5 \le k < 1$	5;	<i>x</i> =	$=\begin{cases} (x-1)\cdot 2\cdot k \end{cases}$	k , $12 \le k < 21$;	<i>x</i> =	$=\begin{cases} 2 \cdot x / (k+1) \end{cases}$), $7 \le k < 18$;		
x	(k+2)/((k+1)	$(k+3)$, $15 \le k < 30$.			$, 21 \le k < 30.$			k , $18 \le k < 30$.		
	$y = (y - S_x)$	(k+1)				١,,				
гле :	V V A	211211211111	L.	$y = y \cdot X_{AV} \cdot$	ĸ	-	•	$(k+7)/X_{MAX}$		
	где S_X – сумма значений исходной переменной x				овательность					
	Контрольные точки:			` -		знач	чения х пр	и 3<=k<=15.		
k=3 x	x=3 $x=2$			- среднее		Кон	трольные т	очки:		
	x=12 x = 5			исходной переменной х			трольные т x = -9	V IIII.		
	k=26 x = -4 Сумма значений Y= -18.5			Контрольные точки:			k=0 x = -9 k=13 x = -4			
Cymn	ла эпачении		k=3 X = -8 k=14 Y = 12			k=27 x = -11				
			z-1/4 = -1/2			Сумма значений Y= -47				
				ма значений	Y= -41					

	Вариант № 10			Вариант № 11			Вариант № 12			
k	X	у	k	X	у	k	X	у		
0	-1.587	-9.78	0	7	-0.219	0	-20	252.1		
1	-1.518	-20.09	1	1.333	-0.094	1	-12.5	251.1		
2	-1.442	-30.54	2	-0.778	0.345	2	-10	250.1		
3	-1.145	-41.28	3	-0.263	0.439	3	-8.75	249.1		
4	-1.260	-53.06	4	-0.182	0.595	4	-8	248.1		
5	-1.260	-65.53	5	-0.118	0.814	5	-8.33	-257.1		
6	-1.145	-78.75	6	-6.817	1.128	6	-5.71	1255		
7	-1	-91.37	7	-5.913	1.566	7	-1.25	1254		
8	0	-104.4	8	-6	1.911	8	1.11	1253		
9	0.794	-116.0	9	-2.080	2.381	9	2	1252		
10	0.794	-125.6	10	-1.154	2.944	10	3.636	-262.1		
11	0.794	-141.4	11	-0.224	3.571	11	2.083	241.1		
12	1	-153.1	12	-3.289	4.542	12	-1.538	240.1		
13	1	-162.4	13	-0.351	5.075	13	-2.500	-4047		
14	1.080	-174.0	14	0.590	5.920	14	-984	-3796		
15	0.666	-182.7	15	1.534	6.797	15	-1132	-3292		
16	0.75	-191.2	16	4.480	7.800	16	-1279	-3041		
17	1.082	-202.5	17	5.429	8.959	17	-1440	-2538		
18	1.413	-213.7	18	4.379	10.09	18	-1615	-2287		
19	1.493	-221.7	19	-0.668	11.25	19	-1800	-2036		
20	1.321	-232.7	20	4.286	12.47	20	-1995	-1785		
21	1.648	-240.3	21	-18.39	13.78	21	-2200	-1534		
22	1.973	-251.2	22	-19.08	15.19	22	-2415	-904.4		
23	149.5	-254.7	23	-19.03	16.76	23	-19.49	-527.2		
24	0	-261.6	24	-29	18.20	24	-28.49	228.1		
25	375	-272.1	25	-30	19.77	25	-27.50	-529.2		
26	0	-278.7	26	-29.97	21.39	26	-27.50	-1287		
27	-229.5	-289.0	27	-30.68	23.09	27	-29.50	-2296		
28	-504	-299.3	28	-31.15	24.78	28	-31.50	-2549		
29	-826.5	-309.6	29	-31.92	26.50	29	-36.00	-3306		
	$\int x^3, 0 \le k < 1$	4;		$\int x \cdot (2 \cdot k^2 + 1)$), $0 \le k < 7$;		$(k+1)\cdot x/$	5, $0 \le k < 14$;		
$x = \langle$	$\sqrt[3]{k/2}-2\cdot x$,	$14 \le k < 23;$	<i>x</i> =	$= \begin{cases} x + \sqrt[3]{k}, & 7 \end{cases}$	$\leq k < 21;$	x =	$= \begin{cases} x + 5 \cdot k^2, \end{cases}$	$14 \le k \le 23$		
)), $23 \le k < 30$.			3 , $21 \le k < 30$.	1	I	$\frac{1}{2}$, $23 \le k < 30$.		
v	$=X_{MAY}\cdot(k-1)$	$+1)/y-X_{AV}$		$y = X_{MN}$	$\cdot v \cdot + k^2$		$y = 2 \cdot (y - y)$			
где X_{max} - максимальное			гле	X_{MIN} — мин			•	. 217		
	значение исходной х			нение исхо	U		-	нее значение		
	X_{AV} - среднее значение ис-			трольные т			одной перо трольные т			
	xодной переменной x			x = -6			x = -4	·		
Конт	Контрольные точки:			x = 8		k=1 x = -4 k=13 x = -7				
	$x=5 \ x=-2$			k=27 x = -2.5			k=24 x = -4			
	k=11 x = 0.5 k=27 x = -0.5			Сумма значений Y= 8.6			Сумма значений Y= 114.5			
	x = -0.5 иа значений	Y= -2								
∪ y IVII	ла эпалении	<u>. –</u>				<u> </u>				

	Вариант № 13			Вариан	т № 14	Вариант № 15				
k	X	y	k	X	y	k	X	y		
0	-28	-0.2	0	-6	1.567	0	-1.913	60.3		
1	-40	-0.0313	1	-8	1.728	1	-4.913	41.1		
2	-13	-0.0185	2	-9.243	1.817	2	-7.710	37.7		
3	-16	-0.0469	3	-9.196	2.014	3	-10.71	11.60		
4	57	-0.0400	4	-10	2.067	4	-13.87	10.77		
5	-176	-0.0370	5	-9.708	2.442	5	-16.71	0		
6	-125	0.0379	6	-10.35	3.776	6	-19.71	-18.84		
7	-196	0.0410	7	-6.937	-6.558	7	-22.82	-26.60		
8	-124	0.0155	8	-6.485	10.44	8	-25.82	-33.5		
9	-204	0.0183	9	-12	4.776	9	-28.59	-47.6		
10	-111	0.0134	10	-8.487	6.942	10	-31.44	-45.2		
11	-200	0.0167	11	-5.950	5.442	11	-0.125	-50.2		
12	-86	0.0133	12	-4.392	4.042	12	-0.118	-41.1		
13	-184	0.0172	13	-5.817	6.109	13	-0.056	-45.9		
14	0	0.0151	14	-1.442	5.192	14	-0.053	-18.84		
15	7.47	0.0301	15	-1.587	9.442	15	-0.1	-12.06		
16	3.50	0.0406	16	-1.442	7.109	16	-0.048	-5.80		
17	8.24	0.0886	17	-1.710	-7.558	17	-0.045	-11.17		
18	4.67	0.1219	18	-2.000	-4.891	18	3.333	-21.54		
19	8.84	0.2819	19	-2.080	-3.558	19	7.5	-20.79		
20	5.60	0.3940	20	-2.154	-5.558	20	10	-10.05		
21	9.33	1.6633	21	-114	-9.558	21	8.333	0		
22	1	2.3553	22	-150	-21.56	22	5.714	4.71		
23	2.5	-3.354	23	-188	13.44	23	3.75	0		
24	-1	-4.801	24	-194	9.776	24	3.333	8.87		
25	-1	-1.726	25	-200	7.942	25	1	17.23		
26	0.5	-2.494	26	-242	6.842	26	0	16.75		
27	0.5	-2.895	27	-286	7.042	27	-1.667	16.30		
28	-0.5	-4.216	28	-256	6.276	28	-2.308	15.87		
29	-3.5	-6.163	29	-263	5.192	29	-5	15.46		
	$\int x/(3\cdot k+7)$	(0) , $0 \le k < 14$;		$\int x + 3 \cdot \sqrt{k}$, 0	$\leq k < 14;$		$\left[(3\cdot k + x)^3 \right]$	$0 \le k < 11;$		
x	$=\begin{cases} x \cdot k / 28, \end{cases}$	$14 \le k < 22;$	x = 0	$x^3 + 9, 14 \le$	k < 21;	x =	$\begin{cases} x \cdot (k+5), \end{cases}$	$11 \le k < 18;$		
	$2 \cdot x + 3$, 2				10), $21 \le k < 30$.			10, $18 \le k < 30$.		
$y = F_K / (y \cdot (k+1)^3)$				y = (k+1)/($y = y \cdot (k \cdot k)$			
гле і	где F_k – последовательность			X_{max} - макс	имальное	где	S_X — сумма	а значений		
	тде <i>F_к</i> — последовательность Фибоначчи (1, 1, 2, 3, 5)			нение исхо			одной пер			
	Фиооначчи (1, 1, 2, 3, 3) Контрольные точки:			Контрольные точки:			Контрольные точки:			
k=7 x	$k=7 \ x = -7$			k=7 x = 1			k=3 x = -5			
	k=18 x = 3						k=15 x = -2 k=29 x = -7			
	x = 1			s x = -o ма значений			э x = -7 ма значений	Y= 24		
Cymn	иа значений	1 = 39	- 5	1911111		J .v1	5114 10111111	- - ·		

	Вариант № 16			Вариан	т № 17	Вариант № 18				
k	X	у	k	X	y	k	X	y		
0	-12	0.004	0	-3	0	0	-1	-3.034		
1	-17	-0.022	1	-2	116.0	1	-1.333	-1.821		
2	-32	-0.022	2	0	261.0	2	-2.333	-1.821		
3	-39	-0.049	3	4	309.3	3	-2.667	-0.607		
4	-39	-0.040	4	11	145.0	4	-2.333	-0.607		
5	-45	-0.049	5	19	130.5	5	-2.667	0.607		
6	-33	-0.027	6 29 218.6				-3.667	0.607		
7	-37	-0.031	7	-2.6	353.5	7	-4	1.821		
8	-30	0.004	8	-2.9	345.4	8	-3.667	1.821		
9	-33	0.271	9	-3.1	316.4	9	-4	3.034		
10	42	0.347	10	-3.4	157.7	10	-5.333	3.034		
11	59	0.378	11	-3.6	116.0	11	7.071	-3.034		
12	48	0.409	12	-3.7	42.07	12	9.487	-3.034		
13	35	0.564	13	-3.4	30.15	13	10.95	3.034		
14	37	0.604	14	-3.7	32.09	14	11.83	0.607		
15	0.04	0.644	15	-3.9	22.57	15	12.65	3.034		
16	0.050	0.911	16	-4.2	20.99	16	13.78	-3.034		
17	0.028	1.204	17	2.333	14.54	17	14.14	3.034		
18	0.024	1.524	18	2.211	7.512	18	15.81	3.034		
19	0.026	1.693	19	1.8	5.144	19	16.12	1.821		
20	-320	2.151	20	1.714	2.337	20	15.49	1.821		
21	-378	2.350	21	1.364	1.585	21	15.81	0.607		
22	-418	2.355	22	1.304	-0.5353	22	17.03	0.607		
23	-483	2.457	23	1.75	-1.0807	23	559	-3.034		
24	-672	2.670	24	1.44	-0.9663	24	603	-3.034		
25	-750	3.008	25	0.692	-0.1615	25	654	-2.428		
26	-806	3.483	26	-0.444	-0.1076	26	706	0		
27	-891	3.737	27	-1.071	-0.2862	27	762	0		
28	-952	3.612	28	-1.241	-0.1423	28	818	-2.428		
29	-986	3.870	29	-0.6	0	29	874	-2.428		
	$\int (x+k)/(k+$	-3), $0 \le k < 15$;		$(x+k^2)$, если	$u \ 0 \le k < 7;$		$3 \cdot x + k$	$0 \le k < 11;$		
x =	$\left\{ (k-5)^2 \cdot x, \right.$	$15 \le k < 20$:	x =		если $7 \le k < 17$;	r		$11 \le k < 23;$		
		$k, 20 \le k < 30.$			если $17 \le k < 30$.		I	$23 \le k < 30.$		
			,							
1 -	717	$-k^2$)/($k+1$)		Α -	$f_{MAX} \cdot (k+1)^2$		$y = y \cdot (X_A)$	171111		
где	$X_{\scriptscriptstyle AV}$ - среди				овательность	где	X_{MIN} — мин	нимальное		
исхо	одной пере	еменной х		` '		знач	чение исхо	дной <i>х</i>		
	Контрольные точки:			X_{max} - максимальное зна-			- среднее	значение		
	$x=9 \ x = -2$			ие исходно		исх	одной пер	еменной х		
	k=17 x = 4 k=21 x = 4			Контрольные точки:			исходной переменной <i>х</i> Контрольные точки:			
	л – т ма значений	Y= 42	K=3 X = -3 $k=11 Y = -3$			k=3 x = -5				
				6 x = -2		k=12 x = -3 k=25 x = 4				
				ма значений	V — /I X		о x = 4 ма значений	Y= -3		
<u></u>			<u> </u>			- J 141	GIIM IOIIIIII			

	Вариан	т № 19		Вариан	т № 20	Вариант № 21				
k	X	у	k	X	y	k	X	y		
0	54.6	0.067	0	-2	-5	0	-7	-3.17		
1	7.389	0.2	1	-2	-40	1	-8	3.17		
2	7.389	0.2	2	-15	-81	2	-14	15.87		
3	2.718	0.167	3	-44	-106.7	3	-16	15.87		
4	20.09	0.067	4	-90	-100	4	-17	12.70		
5	2.718	0	5	-174	-54	5	-18	9.52		
6	0.0183	-0.067	6	-287	0	6	-8.83	6.35		
7	-3.271	-0.167	7	-420	-24.38	7	-8	3.17		
8	-3.634	-0.133	8	-603	-42.88	8	-4.16	-9.52		
9	-3.78	-0.167	9	-820	18.18	9	-0.32	-15.87		
10	-3.42	-0.233	10	-1089	14.96	10	1.54	-28.57		
11	-3.208	-0.233	11	-1452	36	11	4.39	-38.09		
12	-2.884	-0.233	12	-1872	47.15	12	5.26	-47.62		
13	-2.351	-0.233	13	-2352	50.95	13	4.13	-50.79		
14	0	-0.233	14	-2895	38.73	14	3	-53.97		
15	2.466	-0.233	15	0.564	20.75	15	1.877	-57.14		
16	0	-0.2	16	-0.5	15.38	16	0.757	-57.14		
17	-2.571	-0.2	17	-0.062	13.54	17	-0.359	-60.32		
18	3.302	-0.067	18	-0.621	8.203	18	-2.472	-63.49		
19	3.849	-0.233	19	-1.179	5.913	19	-0.583	-79.36		
20	4.309	-0.233	20	-1.236	2.538	20	-1.690	-98.41		
21	3.979	-0.167	21	2.924	0.6012	21	-2.796	-101.6		
22	4.041	-0.067	22	2.962	-0.8491	22	-6.899	-88.89		
23	-11.5	-0.033	23	2.884	-1.193	23	-0.0062	-79.36		
24	-14.4	-0.033		2.802	-0.8331	24	-0.0111	-88.89		
25	-16.67	0		2.962	-0.5791		-0.0125	-92.06		
	-15.6	0.033	26		-0.5010		-0.0136	-92.06		
27	-13.5	0.133		3.107	-0.3454		-0.0124	-92.06		
28	-9.33	0.167		3.107	-0.2846		-0.0132	-95.24		
29	-14.5	0.067	29	3.107	-0.1623	29	-0.0104	-95.24		
	$\int \ln x$, $0 \le k$	< 7;		$\int k^2 + x/(k +$	1), $0 \le k < 15$;		$\int x + 2 \cdot k + 1$	$0 \le k < 6;$		
x =	$\left\{ x^3 / k, 7 \le \right.$	k < 23;	<i>x</i> =	$= \left\{ 2 \cdot x + \sqrt{k} \right\},$	$15 \le k < 21;$	<i>x</i> =	$=\begin{cases} x+\sqrt{(k+2)} \end{cases}$	$\frac{1}{2}$, $16 \le k < 23$;		
	I), $23 \le k < 30$.		$\left(x^3-k\right)$, 21			I	$23 \le k < 30$.		
	$y = y \cdot S_X / X_{AV}$			$y = y \cdot F_K$	$/\left(k+1\right)^3$		$y = y/\Delta$	$X_{AV} - k$		
где	$X_{\scriptscriptstyle AV}$ - среди	нее значение	где	F_k - послед	овательность	где		нее значение		
	исходной переменной <i>х</i>			боначчи (1,	1 7 2 5 1		одной пер			
	S_X — сумма значений ис-			трольные т	0111211		одной пер трольные т			
	ходной переменной х			2 x = 0 $2 x = 3$		k=3	x = -9			
	Контрольные точки:			k=18 x = 3			k=11 x = 8			
	k=5 x=1			Сумма эпанаций V- 11			k=25 x = -5 Сумма значений Y= -13			
	k=16 x = 0 k=28 x = 3					Сум	ма значении	1 1 – -13		
	x = 3 иа значений	Y = -62								
~ J 1V11	JIM IVIIIIII	- UL	l			l				

	Вариант № 22			Вариант № 23			Вариант № 24			
k	X	y	k	X	y	k	X	у		
0	-1	0.26	0	-5	23.65	0	-5	-0.049		
1	-2.442	1.26	1	-7	53.29	1	-4.204	-0.852		
2	-3.913	1.913	2	-10	79.94	2	-4.559	-1.852		
3	-4.913	2.74	3	-16	110.6	3	-4.243	-2.754		
4	-5.817	3.74	4	-25	138.2	4	-2.675	-3.754		
5	-6.587	4.913	5	-35	183.9	5	-1.278	-4.655		
6	-7.442	5.913	6	-34	214.5	6	-0.615	-5.655		
7	-8.442	6.827	7	-62	245.2	7	0.595	-6.655		
8	-9.26	7.74	8	-79	275.8	8	-2.984	-7.606		
9	-8	8.74	9	-89	336.5	9	-2.975	-8.606		
10	-9	9.827	10	-107	364.6	10	-2.980	-9.705		
11	-9.74	10.83	11	-129	415.8	11	-3.017	-10.70		
12	-10.74	11.91	12	-151	450.4	12	-3.021	-11.75		
13	-11.56	12.91	13	-177	450.1	13	-3.018	-12.90		
14	3.738	14	14	13.92	459.7	14	-3.010	-13.95		
15	4.094	15	15	5.904	490.4	15	-2.973	-14.95		
16	4.159	15.83	16	5.359	470	16	-2.977	-16.10		
17	3.526	16.65	17	3.082	497.7	17	17.88	-17.10		
18	3.584	17.39	18	2.828	506.3	18	18.76	-18		
19	4.331	18.39	19	4.142	532.9	19	18.64	-19		
20	4.382	19.48	20	3.835	538.6	20	19.53	-20.25		
21	4.654	20.57	21	3.04	520.2	21	19.42	-21.25		
22	4.7	21.65	22	3.00	543.9	22	20.31	-22.15		
23	-3.8	22.57	23	3.04	639.5	23	20.20	-23.10		
24	-4.2	23.57	24	3.00	666.2	24	17.10	-24.10		
25	-4.4	24.65	25	3.04	614.8	25	18.00	-25.25		
26	-4.2	25.83	26	3.11	638.5	26	17.90	-26.25		
27	-4.4	27.09	27	3.17	662.1	27	18.80	-27.10		
28	-5	28.26	28	3.24	685.8	28	18.71	-28.05		
29	-6	29.26	29	2.88	709.4	29	18.61	-29.05		
	$(x+k)^3$,	$0 \le k < 14;$		$\int x + k + k^2$	$, 0 \le k < 14;$		$\int (x \cdot \sqrt[4]{k+1},$	•		
<i>x</i> =	$= \begin{cases} e^x / k, & 1 \end{cases}$	$4 \le k < 23;$	<i>x</i> =	$=\begin{cases} \sqrt[3]{x} \cdot \sqrt{k}, \end{cases}$	$14 \le k < 21$;	<i>x</i> =	$=$ $\left\{ (x+3)\cdot k^2, \right.$	$8 \le k < 17;$		
	$x = \begin{cases} e^x / k, & 14 \le k < 23; \\ k + 5 \cdot x, & 23 \le k < 30. \end{cases}$			x^3-k , 2	$14 \le k < 21;$ $21 \le k < 30.$		$(x-k)+\sqrt{k}$	\sqrt{k} , $17 \le k < 30$.		
	$y = X_{MIN} \cdot (y - k)$			$v = X_{AV} +$	$v \cdot / (k+1)$		$y = X_{MAX}$	•		
где 2	где X_{MIN} – минимальное			X_{AV} - сред	нее значение	где	X_{max} - макс	имальное		
знач	значение исходной х				менной ч	знач	ление исхо	днои х		
	Контрольные точки:			L'OTTENO ET TEL TO TEOTETATE			трольные то v — _2	очки:		
	k=1 x = -3			1-21			k=5 x = -2 k=13 x = -3			
	k=15 x = 4 k=29 x = -1			k=15 x = 7			k=13 x = -3 k=28 x = -4			
	л — -1 иа значений	Y= 61	k=22 x = 5 Сумма значений Y= -22			Сумма значений Y= 38				
			СуМ	ма значении	122					

приложение і

Министерство науки и образования Украины Национальный технический университет «Харьковский политехнический институт»

Кафедра: «Промышленная и биомедицинская электроника»

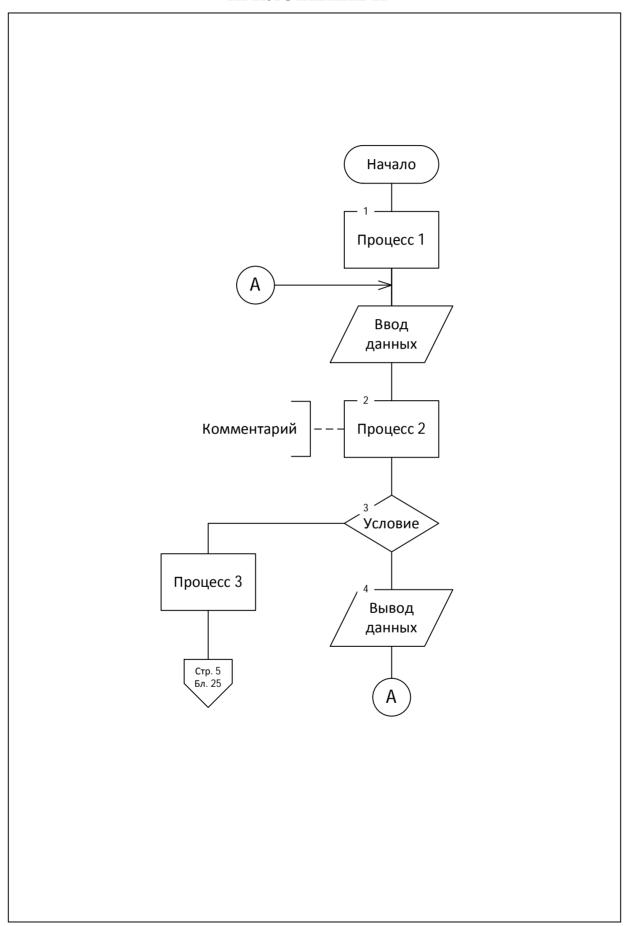
РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ по дисциплине «Информатика»

Выполнил: Студент группы ЭМС-XXX Иванов И.И.

Принял: доц. Король Е.И.

Харьков 2016

приложение и



Навчальне видання

Методичні вказівки

до лабораторних робіт та розрахунково-графычного завдання з дисципліни «Інформатика», Частина 2 для студентів, що навчаються за напрямами 6.050801 «Мікро- та наноелектроніка», 6.050802 «Електронні пристрої та системи» денної та заочної форм навчання

Укладачі: КОРОЛЬ Євген Ігорович

ВЕРЖАНОВСЬКА Марія Ростиславівна

МАХОНІН Микола Васильович

Російською мовою

Роботу до видання рекомендував В. В. Воінов

В авторській редакції

План 2016 р., поз. 45

Підп. до друку 20.07.16. Формат 60×84 1/16. Папір друк. №2. Друк – ризографія. Гарнітура Times New Roman. Ум. друк. арк. 4,0 Наклад 50 прим. Зам №_____. Ціна договірна.

Видавничий центр НТУ "ХПІ". Свідоцтво про державну реєстрацію ДК № 3657 від 24.09.2009 р. 61002, Харків, вул. Фрунзе, 21.

Друкарня НТУ "ХПІ". 61002, Харків, вул. Фрунзе, 21.