|  |  |
| --- | --- |
|  | **Отчёт по лабораторной работе** № 15 по курсу 1 ­  студента группы M80-108Б-18 Коростелев Д. В., № по списку 12  Адреса www, e-mail, jabber, skype  Работа выполнена: “ “ 200 г.  Преподаватель: Поповкин А.В. каф.806  Входной контроль знаний с оценкой  Отчёт сдан “ “ 20 г., итоговая оценка  Подпись преподавателя |

1. **Тема**: практикум по курсам «Основы информатики» и «Алгоритмы и структуры данных», обработка матриц
2. **Цель работы**: составить программу на языке Си, производящую обработку квадратной матрицы порядка N\*N (N<9), из целых чисел, вводимой из стандартного входного текстового файла.
3. **Задание** (*вариант №12*): циклический сдвиг элементов матрицы по спирали против часовой стрелки на n элементов.
4. **Оборудование** (*лабораторное*):

ЭВМ Cameron, процессор Intel Core2 Duo CPU E8500 3.16ГГц, имя узла сети cameron с ОП 16029 МБ

НМД 50 ГБ. Терминал gnome адрес 172.16.80.213. Принтер лазерный принтер с технологией spooling Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel Core i7-7700HQ 4x 2.808ГГц, ОП 3945 МБ, НМД 48,9 ГБ. Монитор

Другие устройства

1. **Программное обеспечение** (*лабораторное*):

Операционная система семейства UNIX , наименование UBUNTU версия 16.04

Интерпретатор команд bash версия 4.3.48

Система программирования gcc версия 7.3.0

Редактор текстов nano версия 2.9.3

Утилиты операционной системы gсс, nano

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства UNIX , наименование UBUNTU версия 18.04

Интерпретатор команд bash версия 4.3.48

Система программирования gcc версия 5.4.0

Редактор текстов nano версия 2.5.3

Утилиты операционной системы gсс, nano

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

1. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями)

Проделывать циклический сдвиг матрицы по спирали будем итерационно, то есть совершать сдвиг матрицы на 1 n-раз. Сам сдвиг будет производиться с помощью 4 функций которые будут перезаписывать элементы матрицы по линиям (влево, вправо) и по столбцам (вверх, вниз). То есть программа будет проходить сначала вниз по столбцу и передвигать его элементы вниз на 1, когда встретим границу, завершаем обход данного столбца, запоминаем, что столбец пройден (вычеркиваем его с в дополнительной матрице), за тем идем вправо по линии, повторяем те же действия, далее вверх и влево, не забываем вычеркивать пройденные линии и столбцы. Повторяем эти действия, пока не пройдем таким образом всю матрицу. Значение последнего элемента обхода присвоим первому элементу. Алгоритм перезаписи состоит в следующем: запомнить значение изначальной переменной (записать в temp), за тем записать переменную на ранее сохраненное значение прошлой.

1. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты, либо соображения по тестированию].

**Тесты**

|  |  |  |  |
| --- | --- | --- | --- |
| **Исходные данные** | | | **Результат** |
| **N** | **Matrix** | **n** |
| 1 | 5 | 1 | 5 |
| 1 | 2 | 5 | 2 |
| 2 | 1 2  3 4 | 1 | 2 4  1 3 |
| 2 | 1 2  3 4 | 3 | 3 1  4 2 |
| 3 | 1 2 3  4 5 6  7 8 9 | 1 | 5 3 6  1 2 9  4 7 8 |
| 3 | 1 2 3  4 5 6  7 8 9 | 2 | 2 6 9  5 3 8  1 4 7 |
| 4 | 1 2 3 4  5 6 7 8  9 10 11 12  13 14 15 16 | 1 | 7 3 4 8  1 2 11 12  5 6 10 16  9 13 14 15 |
| 8 | 1 2 3 4 5 6 7 8  9 10 11 12 13 14 15 16  17 18 19 20 21 22 23 24  25 26 27 28 29 30 31 32  33 34 35 36 37 38 39 40  41 42 43 44 45 46 47 48  49 50 51 52 53 54 55 56  57 58 59 60 61 62 63 64 | 1 | 37 4 5 6 7 8 16 24  29 3 13 14 15 23 31 32  1 2 12 22 30 38 39 40  9 10 11 21 36 46 47 48  17 18 19 20 28 45 55 56  25 26 27 35 43 44 54 64  33 34 42 50 51 52 53 63  41 49 57 58 59 60 61 62 |

*Пункты 1-7 отчёта составляются* ***строго до*** *начала лабораторной работы.*

Допущен к выполнению работы. Подпись преподавателя

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с текстовыми примерами, подписанный преподавателем)

|  |
| --- |
| dmitry@dmitry-VirtualBox:~/lubs/15$ gcc -std=c99 -pedantic -Wall -o 15.out 15.c  dmitry@dmitry-VirtualBox:~/lubs/15$ cat 15.c  #include<stdio.h>  #define Size 8  void \_sRight(int matrix[Size][Size],int \*i,int \*j,int columns[Size],int \*end, int \*temp){  int k = \*j;  \*j = \*j + 1;  while(\*j<Size && columns[\*j] == 0){  int temp\_2 = \*temp;  \*temp = matrix[\*i][\*j];  matrix[\*i][\*j] = temp\_2;  \*j = \*j + 1;  }  \*j= \*j - 1;  if( k == \*j)  \*end = 1;  }  void \_sDown(int matrix[Size][Size],int \*i,int \*j,int lines[Size],int \*end, int \*temp){  int k = \*i;  \*i = \*i + 1;  while(\*i<Size && lines[\*i] != 1){  int temp\_2 = \*temp;  \*temp = matrix[\*i][\*j];  matrix[\*i][\*j] = temp\_2;  \*i = \*i + 1;  }  \*i = \*i - 1;  if(k == \*i)  \*end = 1;  }  void \_sLeft(int matrix[Size][Size],int \*i,int \*j,int columns[Size],int \*end, int \*temp){  int k = \*j;  \*j = \*j - 1;  while(\*j>-1 && columns[\*j]!= 1){  int temp\_2 = \*temp;  \*temp = matrix[\*i][\*j];  matrix[\*i][\*j] = temp\_2;  \*j = \*j - 1;  }  \*j = \*j + 1;  if(k == \*j)  \*end = 1;  }  void \_sUp(int matrix[Size][Size], int \*i,int \*j,int lines[Size],int \*end, int \*temp){  int k = \*i;  \*i = \*i - 1;  while(\*i>-1 && lines[\*i] != 1){  int temp\_2 = \*temp;  \*temp = matrix[\*i][\*j];  matrix[\*i][\*j] = temp\_2;  \*i = \*i - 1;  }  \*i = \*i + 1;  if(k == \*i)  \*end = 1;  }  void \_sMatrix(int matrix[Size][Size], int size){  int temp = matrix[0][0];  int i = 0, j = 0,k = 0;  int lines[Size] = {0};  int columns[Size] = {0};  for(int h = 0;h<Size-size;h++){  lines[Size-1-h] = 1;  columns[Size-1-h]= 1;  }  int end = 0;  while(1){  \_sDown(matrix,&i,&j,lines,&end,&temp);  columns[k]=1;  \_sRight(matrix,&i,&j,columns,&end,&temp);  lines[size-1-k] = 1;  \_sUp(matrix,&i,&j,lines,&end,&temp);  columns[size-1-k] = 1;  \_sLeft(matrix,&i,&j,columns,&end,&temp);  lines[k] = 1;  k++;  if(end == 1){  matrix[0][0] = temp;  break;  }  }  }  int main(void){  printf("Введите размерность квадратной матрицы: ");  int size;  scanf("%d",&size);  printf("Введите матрицу %dx%d:\n",size,size);  int Matrix[Size][Size];  for(int i =0;i<size;i++)  for(int j = 0;j<size;j++)  scanf("%d",&Matrix[i][j]);  printf("Введите кол-во сдвигов: ");  int n;  scanf("%d",&n);  for(int i =0;i<n;i++)  \_sMatrix(Matrix,size);  for(int i =0;i<size;i++){  for(int j =0;j<size;j++)  printf("%d\t ",Matrix[i][j]);  printf("\n");  }  return 0;  }  dmitry@dmitry-VirtualBox:~/lubs/15$ ./15.out  Введите размерность квадратной матрицы: 1  Введите матрицу 1x1:  5  Введите кол-во сдвигов: 1  5  dmitry@dmitry-VirtualBox:~/lubs/15$ ./15.out  Введите размерность квадратной матрицы: 1  Введите матрицу 1x1:  2  Введите кол-во сдвигов: 5  2  dmitry@dmitry-VirtualBox:~/lubs/15$ ./15.out  Введите размерность квадратной матрицы: 2  Введите матрицу 2x2:  1 2  3 4  Введите кол-во сдвигов: 1  2 4  1 3  dmitry@dmitry-VirtualBox:~/lubs/15$ ./15.out  Введите размерность квадратной матрицы: 2  Введите матрицу 2x2:  1 2  3 4  Введите кол-во сдвигов: 3  3 1  4 2  dmitry@dmitry-VirtualBox:~/lubs/15$ ./15.out  Введите размерность квадратной матрицы: 3  Введите матрицу 3x3:  1 2 3  4 5 6  7 8 9  Введите кол-во сдвигов: 1  5 3 6  1 2 9  4 7 8  dmitry@dmitry-VirtualBox:~/lubs/15$ ./15.out  Введите размерность квадратной матрицы: 3  Введите матрицу 3x3:  1 2 3  4 5 6  7 8 9  Введите кол-во сдвигов: 2  2 6 9  5 3 8  1 4 7  dmitry@dmitry-VirtualBox:~/lubs/15$ ./15.out  Введите размерность квадратной матрицы: 4  Введите матрицу 4x4:  1 2 3 4  5 6 7 8  9 10 11 12  13 14 15 16  Введите кол-во сдвигов: 1  7 3 4 8  1 2 11 12  5 6 10 16  9 13 14 15  dmitry@dmitry-VirtualBox:~/lubs/15$ ./15.out  Введите размерность квадратной матрицы: 8  Введите матрицу 8x8:  1 2 3 4 5 6 7 8  9 10 11 12 13 14 15 16  17 18 19 20 21 22 23 24  25 26 27 28 29 30 31 32  33 34 35 36 37 38 39 40  41 42 43 44 45 46 47 48  49 50 51 52 53 54 55 56  57 58 59 60 61 62 63 64  Введите кол-во сдвигов: 2  37 4 5 6 7 8 16 24  29 3 13 14 15 23 31 32  1 2 12 22 30 38 39 40  9 10 11 21 36 46 47 48  17 18 19 20 28 45 55 56  25 26 27 35 43 44 54 64  33 34 42 50 51 52 53 63  41 49 57 58 59 60 61 62  dmitry@dmitry-VirtualBox:~/lubs/15$ |

1. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб.  или  дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  |  |  |  |  |  |  |

1. Замечание автора по существу работы
2. Выводы В этой лабораторной работе студент может показать и закрепить все, чему он научился за весь пройденный семестр. Данная работа являются обобщающей и самой трудоемкой из всех, при выполнении данного задания закрепляются все полученные навыки, также появляется опыт в обработке квадратных целочисленных матриц.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом

Подпись студента