Міністерство освіти і науки України Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Факультет інформатики та обчислювальної техніки Кафедра обчислювальної техніки

Лабораторна робота №6

з дисципліни «Алгоритми і структури даних»

Виконав:

Перевірила:

студент групи IM-31 Литвиненко Сергій Андрійович номер у списку групи: 14 Молчанова А. А.

Завдання

1. Задано двовимірний масив (матрицю) цілих чисел A[m,n] або A[n,n], де m та n — натуральні числа (константи), що визначають розміри двовимірного масиву. Виконати сортування цього масиву або заданої за варіантом його частини у заданому порядку заданим алгоритмом (методом).

Сортування повинно бути виконано безпосередньо у двовимірному масиві «на тому ж місці», тобто без перезаписування масиву та/або його будь-якої частини до інших одно- або двовимірних масивів, а також без використання спискових структур даних.

- 2. Розміри матриці та п взяти самостійно у межах від 7 до 10.
- 3. При тестуванні програми необхідно підбирати такі вхідні набори початкових значеннь матриці, щоб можна було легко відстежити коректність виконання сортування і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

Варіант 14:

Задано квадратну двовимірний масив (матрицю) цілих чисел A[n,n]. Відсортувати головну діагональ масиву алгоритмом №4 методу обмінів («шейкерне сортування») за незбільшенням.

Текст програми:

```
#include <stdio.h>
#define ROWS 7
#define COLS 10
void swap(int* x, int* y) {
  int temp = *x;
  *x = *y;
  *y = temp;
}
void sort(int matrix[][COLS], int length) {
  int left = 0, right = length - 1;
  int changes = 1;
  while (left < right && changes == 1) {</pre>
    changes = 0;
    for (int i = left; i < right; i++) {</pre>
      if (matrix[i][i] < matrix[i + 1][i + 1]) {</pre>
        swap(\&matrix[i][i], \&matrix[i + 1][i + 1]);
        changes = 1;
      }
    }
    right--;
```

```
if (matrix[i][i] > matrix[i - 1][i - 1]) {
       swap(&matrix[i][i], &matrix[i - 1][i - 1]);
       changes = 1;
     }
   }
   left++;
  }
}
void printMatrix(int matrix[][COLS], int rows, int cols) {
  for (int i = 0; i < rows; i++) {
   for (int j = 0; j < cols; j++) {
     printf("%6d", matrix[i][j]);
   }
   printf("\n");
  }
}
int main(int argc, char const *argv[]) {
  int matrix[][COLS] = {
   \{-23, -85, 37, -37, 68, 62, -80, 80, 0, -34\},
    { 27, 58, -40, 36, 45, -33, 77, 5,
                                             26, -35 },
   { 46, -98, 6, -55, -8, -85, 26, 14,
                                             35, 87 },
    \{-25, -36, -5, 88, -84, -54, -84, 55, -95, 6\},
```

for (int i = right; i > left; i--) {

```
\{-98, 41, 65, 14, 27, 13, 46, 88, -69, -24\},
 \{-97, 7, 95, 15, -84, -61, 72, -71, 90, -78\},
 { 75, 74, 94, 57, 44, 77, 80, -45, -20, 86 },
 { 70, 71, -57, -38, -64, 28, 63, -46, 43, -78 },
 { 53, -60, 20, -51, -40, 5, -93, 47, 68, -9 },
 \{-91, 6, 15, -93, 66, 71, 22, 68, -91, -94\},
};
printf("Initial matrix:\n");
printMatrix(matrix, ROWS, COLS);
printf("\n");
sort(matrix, ROWS);
printf("Result matrix:\n");
printMatrix(matrix, ROWS, COLS);
return 0;
```

}

Тестування програми

1) Для матриці:

Програма повертає наступний результат:

| Initial | matrix | x: | | | | | | | | | |
|----------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|--|--|
| 96 | 15 | -79 | 6 | 53 | 84 | 37 | -1 | 45 | -98 | | |
| 26 | 93 | 64 | -5 | -49 | 84 | -75 | -52 | 78 | -89 | | |
| -56 | 83 | 65 | -57 | -42 | -65 | 29 | -47 | -72 | 96 | | |
| 43 | -58 | 46 | 51 | 89 | 57 | -6 | -73 | 96 | -3 | | |
| 65 | -48 | 38 | 94 | 17 | -98 | 98 | 63 | 48 | 85 | | |
| 9 | 21 | 16 | -12 | 5 | 17 | -21 | -35 | -51 | -74 | | |
| -32 | 28 | -23 | -3 | -89 | 97 | 0 | -17 | 63 | -18 | | |
| 84 | -17 | 66 | 34 | 40 | 34 | 78 | -9 | 7 | -73 | | |
| -82 | 69 | -37 | 86 | 8 | -41 | 27 | 62 | -28 | -98 | | |
| 28 | -31 | -74 | -63 | 53 | 91 | 91 | 20 | -11 | -59 | | |
| | | | | | | | | | | | |
| Result i | Result matrix: | | | | | | | | | | |
| 96 | 15 | -79 | 6 | 53 | 84 | 37 | -1 | 45 | -98 | | |
| 26 | 93 | 64 | -5 | -49 | 84 | -75 | -52 | 78 | -89 | | |
| -56 | 83 | 65 | -57 | -42 | -65 | 29 | -47 | -72 | 96 | | |
| 43 | -58 | 46 | 51 | 89 | 57 | -6 | -73 | 96 | -3 | | |
| 65 | -48 | 38 | 94 | 17 | -98 | 98 | 63 | 48 | 85 | | |
| 9 | 21 | 16 | -12 | 5 | 17 | -21 | -35 | -51 | -74 | | |
| -32 | 28 | -23 | -3 | -89 | 97 | 0 | -17 | 63 | -18 | | |
| 84 | -17 | 66 | 34 | 40 | 34 | 78 | -9 | 7 | -73 | | |
| -82 | 69 | -37 | 86 | 8 | -41 | 27 | 62 | -28 | -98 | | |
| 28 | -31 | -74 | -63 | 53 | 91 | 91 | 20 | -11 | -59 | | |

2) Для матриці

Програма повертає наступний результат:

| Initial | matrix | κ: | | | | | | | | | | |
|----------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|--|--|--|
| -23 | -85 | 37 | -37 | 68 | 62 | -80 | 80 | 0 | -34 | | | |
| 27 | -94 | -40 | 36 | 45 | -33 | 77 | 5 | 26 | -35 | | | |
| 46 | -98 | 6 | -55 | -8 | -85 | 26 | 14 | 35 | 87 | | | |
| -25 | -36 | -5 | 88 | -84 | -54 | -84 | 55 | -95 | 6 | | | |
| -98 | 41 | 65 | 14 | 27 | 13 | 46 | 88 | -69 | -24 | | | |
| -97 | 7 | 95 | 15 | -84 | -61 | 72 | -71 | 90 | -78 | | | |
| 75 | 74 | 94 | 57 | 44 | 77 | 6 | -45 | -20 | 86 | | | |
| 70 | 71 | -57 | -38 | -64 | 28 | 63 | -46 | 43 | -78 | | | |
| 53 | -60 | 20 | -51 | -40 | 5 | -93 | 47 | 68 | -9 | | | |
| -91 | 6 | 15 | -93 | 66 | 71 | 22 | 68 | -91 | 58 | | | |
| | | | | | | | | | | | | |
| Result i | Result matrix: | | | | | | | | | | | |
| 88 | -85 | 37 | -37 | 68 | 62 | -80 | 80 | 0 | -34 | | | |
| 27 | 68 | -40 | 36 | 45 | -33 | 77 | 5 | 26 | -35 | | | |
| 46 | -98 | 58 | -55 | -8 | -85 | 26 | 14 | 35 | 87 | | | |
| -25 | -36 | -5 | 27 | -84 | -54 | -84 | 55 | -95 | 6 | | | |
| -98 | 41 | 65 | 14 | 6 | 13 | 46 | 88 | -69 | -24 | | | |
| -97 | 7 | 95 | 15 | -84 | 6 | 72 | -71 | 90 | -78 | | | |
| 75 | 74 | 94 | 57 | 44 | 77 | -23 | -45 | -20 | 86 | | | |
| 70 | 71 | -57 | -38 | -64 | 28 | 63 | -46 | 43 | -78 | | | |
| 53 | -60 | 20 | -51 | -40 | 5 | -93 | 47 | -61 | -9 | | | |
| -91 | 6 | 15 | -93 | 66 | 71 | 22 | 68 | -91 | -94 | | | |

3) Для матриці:

Програма повертає наступний результат:

| Initial matrix: | | | | | | | | | | | |
|-----------------|---|---|--|---|--|---|---|---|--|--|--|
| 59 | -16 | -45 | -75 | -62 | -13 | 12 | 85 | 85 | | | |
| -82 | 8 | -98 | 12 | -88 | -57 | -79 | -29 | 82 | | | |
| -33 | -46 | -44 | 88 | 67 | 68 | 94 | -22 | -23 | | | |
| -93 | -6 | -34 | -26 | -63 | -6 | 72 | 27 | -65 | | | |
| 55 | 19 | -96 | -34 | 12 | -60 | -76 | 48 | 68 | | | |
| -78 | 94 | 73 | 96 | 4 | 81 | 25 | 33 | -22 | | | |
| 1 | 77 | 16 | 6 | -52 | 18 | -38 | -43 | -95 | | | |
| -52 | -78 | 16 | -3 | -19 | 54 | 19 | -70 | 56 | | | |
| 0 | -32 | 89 | -30 | -3 | 37 | 11 | 71 | -76 | | | |
| -61 | 73 | 66 | -56 | 79 | 90 | -51 | -48 | 95 | | | |
| | | | | | | | | | | | |
| Result matrix: | | | | | | | | | | | |
| 59 | -16 | -45 | -75 | -62 | -13 | 12 | 85 | 85 | | | |
| 71 | 8 | -98 | 12 | -88 | -57 | -79 | -29 | 82 | | | |
| -33 | 19 | -44 | 88 | 67 | 68 | 94 | -22 | -23 | | | |
| -93 | -6 | 18 | -26 | -63 | -6 | 72 | 27 | -65 | | | |
| 55 | 19 | -96 | 4 | 12 | -60 | -76 | 48 | 68 | | | |
| -78 | 94 | 73 | 96 | -34 | 81 | 25 | 33 | -22 | | | |
| 1 | 77 | 16 | 6 | -52 | -34 | -38 | -43 | -95 | | | |
| -52 | -78 | 16 | -3 | -19 | 54 | -46 | -70 | 56 | | | |
| 0 | -32 | 89 | -30 | -3 | 37 | 11 | -82 | -76 | | | |
| -61 | 73 | 66 | -56 | 79 | 90 | -51 | -48 | -85 | | | |
| | 59 -82 -33 -93 55 -78 1 -52 0 -61 atrix 59 71 -33 -93 -55 -78 1 -52 0 | 59 -16 -82 8 -33 -46 -93 -6 55 19 -78 94 1 77 -52 -78 0 -32 -61 73 matrix: 59 -16 71 8 -33 19 -93 -6 55 19 -78 94 1 77 -52 -78 0 -32 | 59 -16 -45 -82 8 -98 -33 -46 -44 -93 -6 -34 55 19 -96 -78 94 73 1 77 16 -52 -78 16 0 -32 89 -61 73 66 atrix: 59 -16 -45 71 8 -98 -33 19 -44 -93 -6 18 55 19 -96 -78 94 73 1 77 16 -52 -78 16 0 -32 89 | 59 -16 -45 -75 -82 8 -98 12 -33 -46 -44 88 -93 -6 -34 -26 55 19 -96 -34 -78 94 73 96 1 77 16 6 -52 -78 16 -3 0 -32 89 -30 -61 73 66 -56 matrix: 59 -16 -45 -75 71 8 -98 12 -33 19 -44 88 -93 -6 18 -26 55 19 -96 4 -78 94 73 96 1 77 16 6 -52 -78 16 -3 0 -32 89 -30 | 59 -16 -45 -75 -62 -82 8 -98 12 -88 -33 -46 -44 88 67 -93 -6 -34 -26 -63 55 19 -96 -34 12 -78 94 73 96 4 1 77 16 6 -52 -52 -78 16 -3 -19 0 -32 89 -30 -3 -61 73 66 -56 79 matrix: 59 -16 -45 -75 -62 71 8 -98 12 -88 -33 19 -44 88 67 -93 -6 18 -26 -63 55 19 -96 4 12 -78 94 73 96 -34 1 77 16 6 -52 -78 94 73 96 -34 1 77 16 6 -52 -52 -78 16 -3 -19 0 -32 89 -30 -3 | 59 -16 -45 -75 -62 -13 -82 8 -98 12 -88 -57 -33 -46 -44 88 67 68 -93 -6 -34 -26 -63 -6 55 19 -96 -34 12 -60 -78 94 73 96 4 81 1 77 16 6 -52 18 -52 -78 16 -3 -19 54 0 -32 89 -30 -3 37 -61 73 66 -56 79 90 eatrix: 59 -16 -45 -75 -62 -13 71 8 -98 12 -88 -57 -33 19 -44 88 67 68 -93 -6 18 -26 -63 -6 55 19 -96 4 12 -60 -78 94 73 96 -34 81 1 77 16 6 -52 -34 -52 -78 16 -3 -19 54 0 -32 89 -30 -3 37 | 59 -16 -45 -75 -62 -13 12 -82 8 -98 12 -88 -57 -79 -33 -46 -44 88 67 68 94 -93 -6 -34 -26 -63 -6 72 55 19 -96 -34 12 -60 -76 -78 94 73 96 4 81 25 1 77 16 6 -52 18 -38 -52 -78 16 -3 -19 54 19 0 -32 89 -30 -3 37 11 -61 73 66 -56 79 90 -51 matrix: 59 -16 -45 -75 -62 -13 12 -71 8 -98 12 -88 -57 -79 -33 19 -44 88 67 68 94 -93 -6 18 -26 -63 -6 72 -55 19 -96 4 12 -60 -76 -78 94 73 96 -34 81 25 -78 94 73 96 -34 81 25 -78 94 73 96 -34 81 25 -78 94 73 96 -34 81 25 -78 94 73 96 -34 81 25 -78 94 73 96 -34 81 25 -78 94 73 96 -34 81 25 -52 -78 16 -3 -19 54 -46 0 -32 89 -30 -3 37 11 | 59 -16 -45 -75 -62 -13 12 85 -82 8 -98 12 -88 -57 -79 -29 -33 -46 -44 88 67 68 94 -22 -93 -6 -34 -26 -63 -6 72 27 55 19 -96 -34 12 -60 -76 48 -78 94 73 96 4 81 25 33 1 77 16 6 -52 18 -38 -43 -52 -78 16 -3 -19 54 19 -70 0 -32 89 -30 -3 37 11 71 -48 -78 -94 88 67 68 94 -22 -93 -6 18 -98 12 -88 -57 -79 -29 -33 19 -44 88 67 68 94 -22 -93 -6 18 -26 -63 -6 72 27 55 19 -96 4 12 -60 -76 48 -78 94 73 96 -34 81 25 33 1 77 16 6 -52 -34 81 25 33 1 77 16 6 -52 -34 81 25 33 1 77 16 6 -52 -34 -38 -43 -52 -78 16 -3 -19 54 -46 -70 0 -32 89 -30 -3 37 11 -82 | | | |