Проектна работа №17

Изготвил: Петко Господинов Господинов

Фк. Номер: 371222015

Специалност: ИСИИ-АЕ, Група: 90Б

Условие:

Да се състави програма за обработка на 3 масива A[N,N], В[N,N] и С[N,N], където данните са

цели числа в интервала [-1000; 1000]. Програмата да извърши следните действия:

* отпечатване на условието на задачата;
* отпечатване на имената на автора на програмата;
* въвеждане на входните данни;
* отпечатване на входните данни;
* а) да се образува едномерен масив D[N], който се образува от всички елементи по-големи от Р
* (Р се въвежда от клавиатурата), които лежат под главния диагонал на дадения масив;
* б) полученият масив да се сортира по големина;
* отпечатване на получените резултати след обработка а) и след обработка б)

Блок схема:Картина, която съдържа диаграма

Описанието е генерирано автоматично

Source code:

#include <stdio.h>

#include <stdlib.h>

#define N 3 // Array size

int main() {

int A[N][N], B[N][N], C[N][N];

int p, D[10];

int i, j;

//Requirements

printf("Program to process 3 arrays A[N,N], B[N,N] and C[N,N], where given integers in the interval [-1000; 1000].\n");

printf("The program performs the following actions:\n");

printf("a) to form a one-dimensional array D[N], which is formed by all elements greater than P that lie below the main diagonal of the given array;\n");

printf("b) sort the array by size.\n");

//Author

printf("Ot: Petko Gospodinov Gospodinov #371222015\n");

//Input Arrays

printf("Enter data for Array A[3][3]:\n");

do {

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

printf("A[%d][%d]: ", i, j);

scanf\_s("%d", &A[i][j]);

}

}

} while (A[i][j] <= -1000 && A[i][j] >= 1000);

printf("Enter data for Array B[3][3]:\n");

do {

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

printf("B[%d][%d]: ", i, j);

scanf\_s("%d", &B[i][j]);

}

}

} while (A[i][j] <= -1000 && A[i][j] >= 1000);

printf("Enter data for Array C[3][3]:\n");

do {

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

printf("C[%d][%d]: ", i, j);

scanf\_s("%d", &C[i][j]);

}

}

} while (A[i][j] <= -1000 && A[i][j] >= 1000);

//Print matrices

printf("Data in A:\n");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

printf("%d ", A[i][j]);

}

printf("\n");

}

printf("Data in B:\n");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

printf("%d ", B[i][j]);

}

printf("\n");

}

printf("Data in C:\n");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

printf("%d ", C[i][j]);

}

printf("\n");

}

printf("Enter the value of P: ");

scanf\_s("%d", &p);

int idx = 0;

for (i = 0; i < N; i++) {

for (j = 0; j < i; j++) {

if (A[i][j] > p && i > j) {

D[idx] = A[i][j];

idx++;

}

}

}

for (i = 0; i < N; i++) {

for (j = 0; j < i; j++) {

if (B[i][j] > p && i > j) {

D[idx] = B[i][j];

idx++;

}

}

}

for (i = 0; i < N; i++) {

for (j = 0; j < i; j++) {

if (B[i][j] > p && i > j) {

D[idx] = C[i][j];

idx++;

}

}

}

printf("Data in D:\n");

for (i = 0; i < idx; i++) {

printf("%d ", D[i]);

}

//sorting array D (Bubble method)

for (i = 0; i < idx - 1; i++) {

for (j = 0; j < idx - i - 1; j++) {

if (D[j] > D[j + 1]) {

int temp = D[j];

D[j] = D[j + 1];

D[j + 1] = temp;

}

}

}

printf("\nSorted data in D:\n");

for (i = 0; i < idx; i++) {

printf("%d ", D[i]);

}

return 0;

}

Примерен тест:

Картина, която съдържа текст

Описанието е генерирано автоматично

Картина, която съдържа текст

Описанието е генерирано автоматично