**Федеральное государственное образовательное бюджетное учреждение  
высшего образования  
«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**Отчет по домашнему заданию по УП.01.01 Учебная практика**

**Вариант №1**

**специальности 09.02.03 Программирование в компьютерных системах**

**Работа № 11**

Выполнил студент

группы 2ПКС-316

Адещенко Кирилл

Преподаватель:

Пестов А. И.

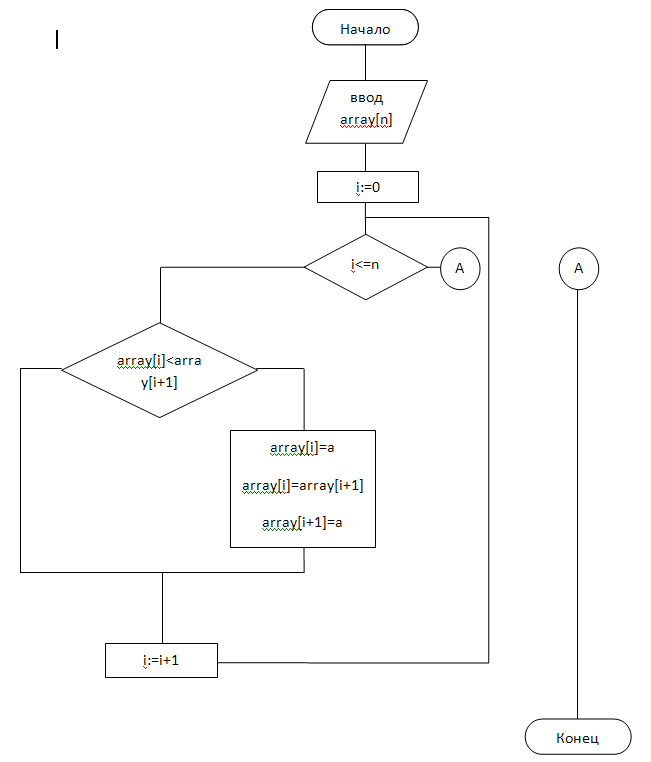
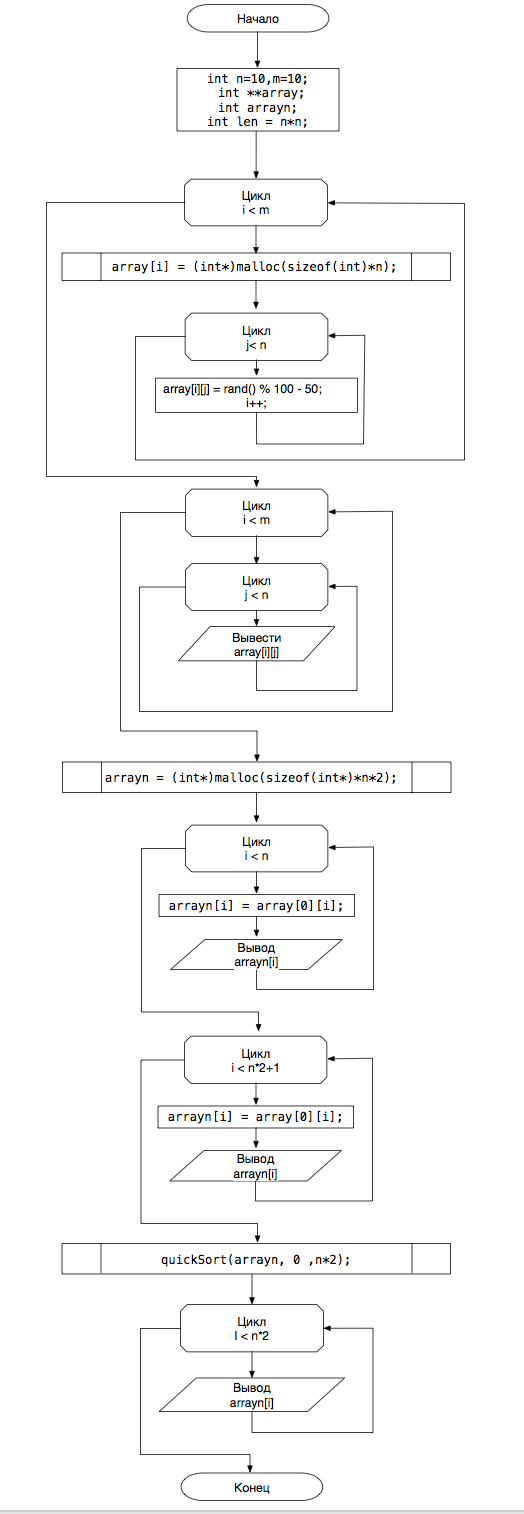
Москва 2018

Задача № 1

-------------------------------------------------------------------------------------------------------------

Дана прямоугольная матрица. Сформировать из положительных элементов матрицы, расположенных в первой и последней строках, новый одномерный массив. Полученный массив сортировать в порядке убывания элементов. Отсортировать сортировкой Хоара.

-------------------------------------------------------------------------------------------------------------



-------------------------------------------------------------------------------------------------------------

Программа

-------------------------------------------------------------------------------------------------------------

//

// main.c

// Practice\_11

//

// Created by Kirill on 13.03.2018.

// Copyright © 2018 Kirill. All rights reserved.

//

#include <stdio.h>

void quickSort(int \*numbers, int left, int right)

{

int pivot; // разрешающий элемент

int l\_hold = left; //левая граница

int r\_hold = right; // правая граница

pivot = numbers[left];

while (left < right) // пока границы не сомкнутся

{

while ((numbers[right] >= pivot) && (left < right))

right--; // сдвигаем правую границу пока элемент [right] больше [pivot]

if (left != right) // если границы не сомкнулись

{

numbers[left] = numbers[right]; // перемещаем элемент [right] на место разрешающего

left++; // сдвигаем левую границу вправо

}

while ((numbers[left] <= pivot) && (left < right))

left++; // сдвигаем левую границу пока элемент [left] меньше [pivot]

if (left != right) // если границы не сомкнулись

{

numbers[right] = numbers[left]; // перемещаем элемент [left] на место [right]

right--; // сдвигаем правую границу вправо

}

}

numbers[left] = pivot; // ставим разрешающий элемент на место

pivot = left;

left = l\_hold;

right = r\_hold;

if (left < pivot) // Рекурсивно вызываем сортировку для левой и правой части массива

quickSort(numbers, left, pivot - 1);

if (right > pivot)

quickSort(numbers, pivot + 1, right);

}

// START MAIN

int main(int argc, const char \* argv[]) {

int n,m;

puts("\nПривет, это одинадцатоя задача в практики программировании!");

puts("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

puts("Task :");

puts("Дана прямоугольная матрица. Сформировать из положительных элементов матрицы, расположенных в первой и последней строках, новый одномерный массив. Полученный массив сортировать в порядке убывания элементов.");

puts("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

puts("");

n = 10;

m = 10;

int \*\*array;

array = (int\*\*)malloc(sizeof(int\*)\*m);

for(int i = 0; i < m; i++){

array[i] = (int\*)malloc(sizeof(int)\*n);

for (int j = 0; j < n; j++){

array[i][j] = rand() % 100 - 50;

}

}

puts("Массив задан :)\n");

printf("%s", "Оригинальный двумерный массив: \n");

for(int i = 0; i < m; i++){

for (int j = 0; j < n; j++){

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

}

puts("");

}

puts("\n\n");

printf("%s", "Оригинальный одномерный массив: \n");

int arrayn[n\*2];

int j = 0;

for (int i = 0; i < n; i++){

arrayn[i] = array[0][i];

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

j++;

}

for (int i = 0; i < n; i++){

arrayn[j] = array[n-1][i];

printf("%d ", array[n-1][i]);

j++;

}

// Start sorted

quickSort(arrayn, 0 ,n\*2);

puts("\n\n\n");

printf("%s", "Отсортированный одномерный массив: \n");

for (int i = 0; i < n\*2; i++){

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

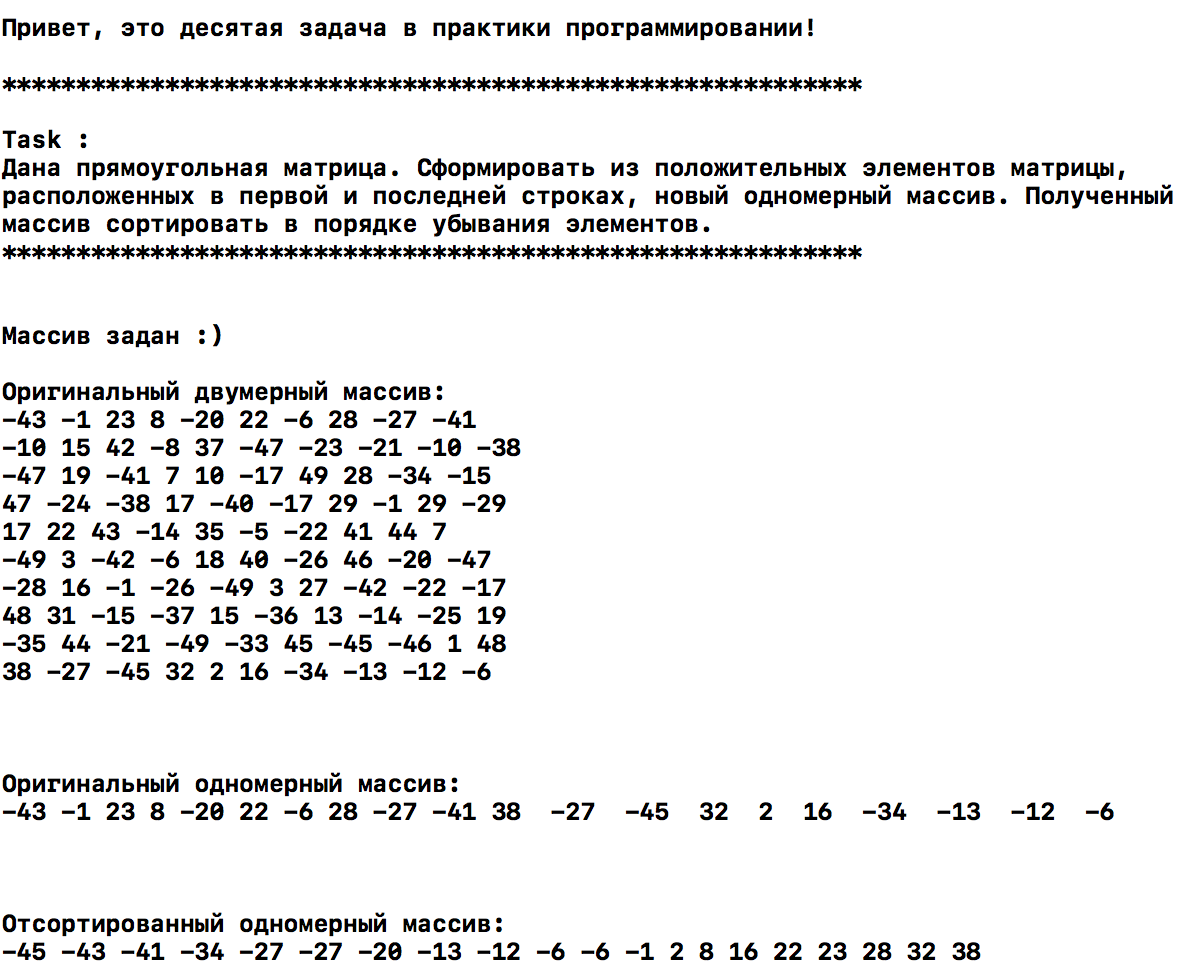
}

puts("");

return 0;

}

-------------------------------------------------------------------------------------------------------------



-------------------------------------------------------------------------------------------------------------