Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Отчёт

по лабораторной работе № 7

«Функции»

по курсу:

«Основы алгоритмизации и программирования»

Выполнил: Проверил:

Студент группы 950503

Гуринович Андрей Викторович Дубовский Андрей Леонидович

Минск 2019

Цель работы: научиться создавать и использовать новые функции пользователя

Вариант: 6

Задание: дана действительная квадратная матрица порядка n. Построить последовательность действительных чисел a1,...an по правилу: если в i-ой строке матрицы элемент, принадлежащий главной диагонали, отрицателен, то ai равно сумме элементов i-ой строки, предшествующих первому отрицательному элементу, иначе ai равно сумме последних элементов i-ой строки, начиная по порядку с неотрицательного элемента. Память под массивы отводить динамически. Программу разбить на функции.

Код программы:

#include <stdio.h>

#include <stdlib.h>

**int** scanIntNumber(){

**int** number;

**do**{

rewind(stdin);

}**while**(!(scanf("%d", &number)) || number <= 0);

**return** number;

}

**int**\*\* CreateIntMatrixWithSize(**int** size){

**int** \*\*matrix = (**int**\*\*)malloc(size \* **sizeof**(**int**\*));

**for** (**int** rows = 0; rows < size; rows++){

matrix[rows] = (**int**\*)malloc(size \* **sizeof**(**int**));

}

**return** matrix;

}

**void** ScanIntMatrixWithSize(**int**\*\* matrix, **int** size){

**for**(**int** v = 0; v < size; v++){

**for** (**int** h = 0; h < size;){

rewind(stdin);

printf("matrix[%d][%d] = ", v, h);

**if**(scanf("%d", &matrix[v][h])){

h++;

}

}

}

}

**void** PrintIntMatrixWithSize(**int**\*\* matrix, **int** size){

**for** (**int** v = 0; v < size; v++){

**for** (**int** h = 0; h < size; h++) {

printf("%4d", matrix[v][h]);

}

printf("\n");

}

}

**int**\* CreateIntArrayWithSize(**int** size){

**int**\* array = calloc(size, **sizeof**(**int**));

**return** array;

}

**void** PrintIntArrayWithSize(**int**\* array, **int** size){

**for**(**int** i = 0; i < size; i++){

printf("%4d", array[i]);

}

}

**int** IsIntPositive(**int** number){

**if**(number >= 0){

**return** 1;

} **else** {

**return** 0;

}

}

**int** main() {

printf("Enter matrix size: ");

**int** size = scanIntNumber();

**int**\*\* matrix = CreateIntMatrixWithSize(size);

ScanIntMatrixWithSize(matrix, size);

PrintIntMatrixWithSize(matrix, size);

**int**\* array = CreateIntArrayWithSize(size);

**for**(**int** row = 0; row < size; row++){

**if** (IsIntPositive(matrix[row][row])){

**int** i = size - 1;

**while**(!IsIntPositive(matrix[row][i])){

i--;

}

**while**(i >= 0){

array[row] += matrix[row][i];

i--;

}

} **else** {

**for**(**int** i = 0; i < size; i++){

**if**(IsIntPositive(matrix[row][i])){

array[row] += matrix[row][i];

} **else** {

**break**;

}

}

}

}

free(matrix);

puts("Result:");

PrintIntArrayWithSize(array,size);

free(array);

**return** 0;

}

Вывод: при выполнении данной лабораторной был проведён глубокий анализ всех предыдущих работ, все функции, используемые в данной работе, были вычленены из предыдущих моих работ. Были усвоены принципы наименования функций: необходимо составлять изъясняющие суть функции предложения. Пример: функция CreateIntArrayWithSize очевидно сознаёт (create) массив (array) типа int, с известным размером (size).