Министерство образования Республики Беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра экономической информатики

**Лабораторная работа №4**

**«Одномерные массивы»**

**Вариант 15/45**

|  |  |
| --- | --- |
| Выполнил: | студент группы 477603  Климович Иван Вячеславович |
| Проверил: | ассистент  кафедры ЭИ  Полоско Е.И. |

Минск 2024

**Цель:** познакомиться с созданием и обработкой одномерных массивов данных. Научиться использовать одномерные массивы в алгоритмах вычисления выражений, сортировки, поиска.

**Задачи:** написать код программ с использованием одномерных массивов.

**Индивидуальное задание**

Библиотеки, функции main() и mainMenu():

#include <stdio.h>

#include <math.h>

#include <locale.h>

#include <stdlib.h>

#include <stdint.h>

#include <string.h>

#include <stdbool.h>

#include "LibForMe.h"

int main()

{

char\* locale = setlocale(LC\_ALL, "");

system("chcp 1251");

system("cls");

while (mainMenu());

}

int mainMenu()

{

int n; printf("Что вы хотите сделать?\n");

printf("1 -- Поиграться с массивом\n");

printf("2 -- Ещё раз поиграться с массивом\n");

printf("3 -- Выйти\n");

n = inputNatural();

switch (n)

{

case 1:

superMassive();

break;

case 2:

massiveXYZ();

break;

case 3:

return 0;

break;

default:

{

printf("Неверный ввод\n");

printf("Press any key to continue...\n");

\_getch();

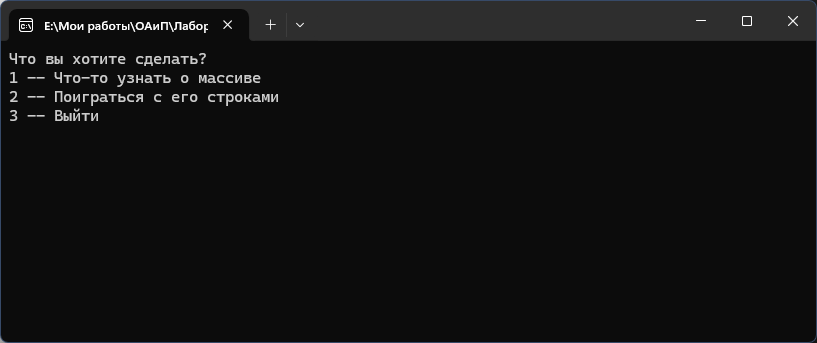
system("cls");

}

}

return 1;

}



**Подпрограмма 1(вариант 15)**

*Задача:*

*В одномерном массиве, состоящем из n вещественных элементов, вы-числить:*

*– номер минимального элемента массива;*

*– сумму элементов массива, расположенных между первым и вторым от-рицательными элементами.*

superMassive()

{

float\* massive; //Создание указателя на массив

float min, sum = 0;

char\* input;

int inputLength;

char buff[100];

int length;

printf("Введите длину массива: ");

length = inputNatural();

massive = (float\*)malloc(length \* sizeof(float)); //Динамическое выделение памяти для массива

printf("Введите элементы массива:\n"); //Введение элементов массива

for (int i = 0; i < length; i++)

{

printf("a[%d] = ", i);

fgets(buff, sizeof(buff), stdin);

inputLength = strlen(buff) - 1;

input = (char\*)malloc(inputLength \* sizeof(char));

for (int i = 0; i < inputLength; i++) input[i] = buff[i];

memset(buff, 0, sizeof(buff));

if (isDigit(input, inputLength) == true)

{

for (int i = 0; i < inputLength; i++)

if (input[i] == '.') { input[i] = ','; }

float buf;

sscanf\_s(input, "%f", &buf);

massive[i] = buf;

buf = 0;

}

else mainMenu();

free(input);

}

for (int i = 0; i < length; i++) printf("%.2f ", massive[i]);

printf("\n");

min = massive[0];

for (int i = 0; i < length; i++)

if (massive[i] < min) min = massive[i];

printf("Минимальный элемент массива: %.3f\n", min);

printf("Нажмите любую клавишу, чтобы найти сумму элементов между крайними отрицательными\n");

\_getch();

float negative1 = 0, negative2 = 0;

for (int i = 0; i < length; i++)

if (massive[i] < 0)

{

negative1 = i+1;

break;

}

if (negative1 == 0)

{

free(massive);

printf("В массиве нет отрицательных элементов\n");

printf("Press any key to continue...\n");

\_getch();

system("cls");

mainMenu();

}

for (int i = negative1; i < length; i++)

if (massive[i] < 0)

{

negative2 = i;

break;

}

if (negative2 == 0) printf("В массиве только 1 отрицательное число\n");

else

{

for (int i = negative1; i < negative2; i++)

{

sum += massive[i];

}

printf("Сумма элементов между первыми двумя отрицательным элементами: %.3f\n", sum);

}

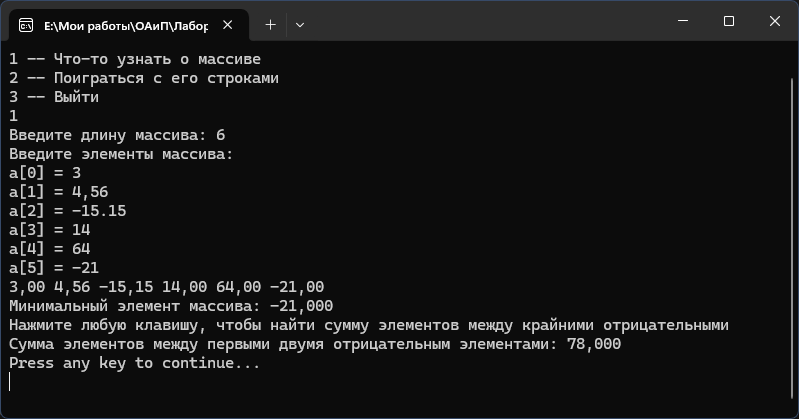
free(massive);

printf("Press any key to continue...\n");

\_getch();

system("cls");

}



**Подпрограмма 2(вариант 45)**

*Задача:*

*В массиве A из 20 элементов все отрицательные элементы отправить в конец массива. Вывести полученный массив в обратном порядке. Вывести исходный массив.*

int massiveXYZ()

{

float min, sum = 0;

char\* input;

int inputLength = 20;

char buff[100];

float massive[20];

printf("Введите элеметны массива: \n");

for (int i = 0; i < 20; i++)

{

printf("a[%d] = ", i);

fgets(buff, sizeof(buff), stdin);

inputLength = strlen(buff) - 1;

input = (char\*)malloc(inputLength \* sizeof(char));

for (int i = 0; i < inputLength; i++) input[i] = buff[i];

memset(buff, 0, sizeof(buff));

if (isDigit(input, inputLength) == true)

{

for (int i = 0; i < inputLength; i++)

if (input[i] == '.') { input[i] = ','; }

float buf;

sscanf\_s(input, "%f", &buf);

massive[i] = buf;

buf = 0;

}

else mainMenu();

free(input);

}

printf("Исходный массив:\n");

for (int i = 0; i < 20; i++) printf("%.2f ", massive[i]);

rearrangeArray(massive);

printf("\nМассив после перемещения отрицательных элементов в конец:\n");

for (int i = 0; i < 20; i++) printf("%.2f ", massive[i]);

printf("\nМассив в обратном порядке:\n");

for (int i = 19; i >= 0; i--) printf("%.2f ", massive[i]);

printf("\n");

free(massive);

printf("Press any key to continue...\n");

\_getch();

system("cls");

return 0;

}

rearrangeArray(float arr[]) //Функция, перемещающая все отрицательные элементы в конец массива

{

int temp[20];

int j = 0;

// Копируем все неотрицательные элементы в temp

for (int i = 0; i < 20; i++) {

if (arr[i] >= 0) {

temp[j++] = arr[i];

}

}

// Копируем все отрицательные элементы в temp

for (int i = 0; i < 20; i++) {

if (arr[i] < 0) {

temp[j++] = arr[i];

}

}

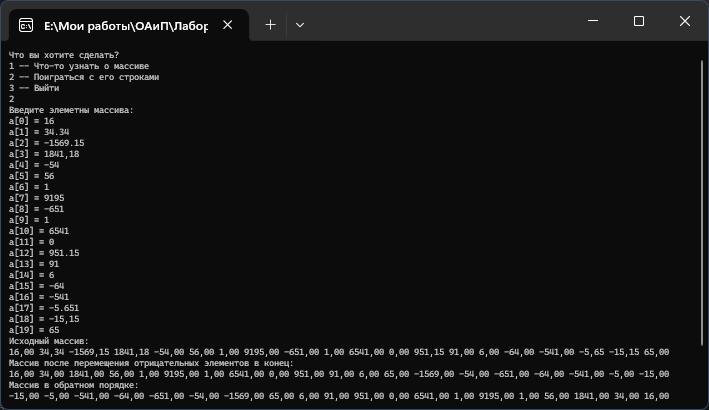
// Копируем temp обратно в arr

for (int i = 0; i < 20; i++) {

arr[i] = temp[i];

}

}

**

**Вспомогательные функции**

int inputNatural()

{

char\* input;

int inputLength;

char buff[100];

int n = 0;

fgets(buff, sizeof(buff), stdin);

inputLength = strlen(buff) - 1;

input = (char\*)malloc(inputLength \* sizeof(char));

for (int i = 0; i < inputLength; i++) input[i] = buff[i];

memset(buff, 0, sizeof(buff));

if (isNatural(input, inputLength) == true) {

sscanf\_s(input, "%d", &n); free(input); return n;

}

else mainMenu();

}

int inputInteger()

{

char\* input;

int inputLength;

char buff[100];

int n = 0;

fgets(buff, sizeof(buff), stdin);

inputLength = strlen(buff) - 1;

input = (char\*)malloc(inputLength \* sizeof(char));

for (int i = 0; i < inputLength; i++) input[i] = buff[i];

memset(buff, 0, sizeof(buff));

if (isInteger(input, inputLength) == true) {

sscanf\_s(input, "%d", &n); free(input); return n;

}

else mainMenu();

}

int isNumber(char input[], int length)

{

int n, i, counter = 0;

n = length;

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

if (input[i] >= '0'

&& input[i] <= '9'

|| input[i] == '+'

|| input[i] == '-'

|| input[i] == '.'

|| input[i] == ',') counter++;

if (counter == n) return true;

else

{

printf("Вы явно ввели не число\n");

printf("Press any key to continue...\n");

\_getch();

system("cls");

return false;

}

}

int isNatural(char arr[], int length)

{

int nat, i, n, errors = 0;

n = length;

if (isNumber(arr, n) == true)

{

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

if (arr[i] == '-' || arr[i] == '+') errors++;

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

if (arr[i] == ',' || arr[i] == '.') errors++;

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

if ((arr[i] < '0'

|| arr[i] > '9')

&& arr[i] != " ") errors++;

if (errors != 0)

{

printf("Вы явно ввели что-то не то\n");

printf("Press any key to continue...\n");

\_getch();

system("cls");

return false;

}

return true;

}

else return false;

}

int isInteger(char arr[], int length)

{

int i, n, errors = 0;

n = length;

if (isNumber(arr, n) == true)

{

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

if (arr[i] == ',' || arr[i] == '.') errors++;

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

if (arr[i] != ' '

&& arr[i] != '+'

&& arr[i] != '-'

&&( arr[i] < '0'

|| arr[i] > '9')) errors++;

if (errors != 0)

{

printf("Вы явно ввели что-то не то\n");

printf("Press any key to continue...\n");

\_getch();

system("cls");

return false;

}

return true;

}

else return false;

}

int isDigit(char arr[], int length)

{

int i, n, errors = 0;

n = length;

if (isNumber(arr, n) == true)

{

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

if ((arr[i] < '0'

|| arr[i] > '9')

&& arr[i] != ' '

&& arr[i] != '+ '

&& arr[i] != '-'

&& arr[i] != '.'

&& arr[i] != ',') errors++;

int pointCounter = 0;

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

{

if (arr[i] == ',' || arr[i] == '.')

{

if (arr[i] == '.') { arr[i] == ','; }

pointCounter++;

}

}

if (pointCounter > 1) errors++;

if (errors != 0)

{

printf("Вы явно ввели что-то не то\n");

printf("Press any key to continue...\n");

\_getch();

system("cls");

return false;

}

return true;

}

else return false;

}

**Выводы:** Изучил одномерные массивы.