Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

По дисциплине «Основы алгоритмизации и программирования»

За 2 семестр

Тема: «Указатели на функции»

Выполнили:

Студент 1 курса

Группы ИИ-21(1)

Кабак Д. Н.

Проверил:

Гирель Т. Н.

Брест 2022

**Цель:** научиться пользоваться функциями и рекурсией.

**Ход работы**

Вариант 3

**Задание 1:**

Описать процедуру NMinmax(A, N, NMin, NMax), находящую номера минимального и максимального элемента вещественного массива A размера N. Выходные параметры целого типа: NMin (номер минимального элемента) и NMax (номер максимального элемента). С помощью этой процедуры найти номера минимальных и максимальных элементов массивов A,B, C размера NA, NB, NC соответственно.

Код:

#include <iostream>

#include <random>

using namespace std;

void nMaxMin(double\* &arr, int N)

{

double max = arr[0], min = arr[0];

int nMax = 0, nMin = 0;

for (int i = 1; i < N; i++)

{

if (arr[i] > max)

{

max = arr[i];

nMax = i;

}

if (arr[i] < min)

{

min = arr[i];

nMin = i;

}

}

cout << "nMin = " << nMin << ", nMax = " << nMax;

}

int main()

{

setlocale(LC\_CTYPE, "rus");

int NA, NB, NC;

cout << "Введите количество элементов массива A:";

cin >> NA;

cout << "Введите количество элементов массива B:";

cin >> NB;

cout << "Введите количество элементов массива C:";

cin >> NC;

double\* A = new double[NA];

double\* B = new double[NB];

double\* C = new double[NC];

random\_device rand;

mt19937 gen(rand());

uniform\_real\_distribution<> dist(-100, 100);

cout << "\n" << "A:" << "\t";

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

{

A[i] = dist(gen);

cout << A[i] << "\t";

}

cout << "\n" << "B:" << "\t";

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

{

B[i] = dist(gen);

cout << B[i] << "\t";

}

cout << "\n" << "C:" << "\t";

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

{

C[i] = dist(gen);

cout << C[i] << "\t";

}

cout << "\nA:\t"; nMaxMin(A, NA);

cout << "\nB:\t"; nMaxMin(B, NB);

cout << "\nC:\t"; nMaxMin(C, NC);

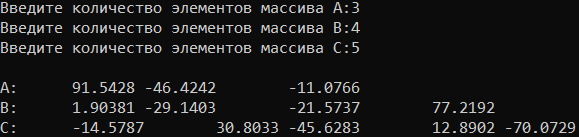
delete[] A;

delete[] B;

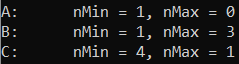
delete[] C;

}

Ввод:



Вывод:



**Задание 2:**

Описать процедуру UpCaseRus(S), преобразующую все строчные русские буквы строки S в прописные (остальные символы строки S не изменяются). Строка S является входным и выходным параметром. Используя процедуру UpCaseRus, преобразовать пять данных строк.

Код:

#include <iostream>

#include <cstdlib>

#include <string>

using namespace std;

void LowCaseRus(std::string& S) {

for (int i = 0; i < S.size(); i++)

{

if (((int)S[i] >= -32) & ((int)S[i] < 0))

{

S[i] = char(int(S[i]) - 32);

}

if ((int)S[i] == -72)

{

S[i] = char(int(S[i]) - 16);

}

}

}

int main()

{

system("chcp 1251>nul");

std::string one, two, three, four, five;

cout << "Введите 5 произвольных строк\n";

cin >> one >> two >> three >> four >> five;

UpCaseRus(one); cout << one << "\n";

UpCaseRus(two); cout << two << "\n";

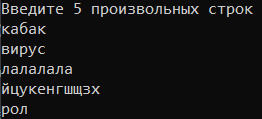
LowCaseRus(three); cout << three << "\n";

LowCaseRus(four); cout << four << "\n";

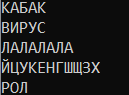
LowCaseRus(five); cout << five << "\n";

}

Ввод:



Вывод:



**Задание 3:**

Описать рекурсивную функцию PowerN(X, N) вещественного типа, находящую значение N-й степени числа X по формулам:

X0 = 1,

Xn = X·Xn-1 при нечетных N > 0,

Xn = 1/X-n при N < 0

(X ≠ 0 — вещественное число, N — целое; в формуле для четных N должна использоваться операция целочисленного деления). С помощью этой функции найти значения Xn для данного X при пяти данных значениях N.

Код:

#include <iostream>

using namespace std;

int powerN(int x, int n)

{

if (x == 0)

{

cout << "Ошибка! Вы ввели неподдерживаемое число (0)!";

return 0;

}

if (n < 0) return 1 / powerN(x, -n);

if (n == 0) return 1;

else return x \* powerN(x, n - 1);

}

int main()

{

setlocale(LC\_ALL, "rus");

double X;

int oneN, twoN, threeN, fourN, fiveN;

cout << "Введите число для возведения в степень:";

cin >> X;

cout << "Введите 5 степеней для этого числа.\n";

cin >> oneN;

cin >> twoN;

cin >> threeN;

cin >> fourN;

cin >> fiveN;

cout << oneN <<".\t" << powerN(X, oneN) << "\n";

cout << twoN <<".\t" << powerN(X, twoN) << "\n";

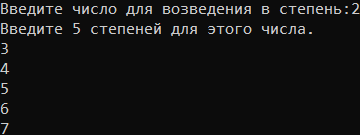
cout << threeN <<".\t" << powerN(X, threeN) << "\n";

cout << fourN <<".\t" << powerN(X, fourN) << "\n";

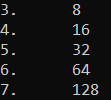
cout << fiveN <<".\t" << powerN(X, fiveN) << "\n";

}

Ввод:



Вывод:



**Вывод:** научился пользоваться функциями и рекурсиями.