Русенски университет „Ангел Кънчев“

Факултет *Природни науки и образование*

Катедра „Информатика и Информационни технологии“

**КУРСОВА ЗАДАЧА**

**по Увод в програмирането**

на Лазар Светлозаров Попов,

ФН 166503, курс I, група 50A, спец. КН

дата на предаване: проверил:

14.11.2016 г. доц. д-р П. Христова

**ЗАДАНИЕ I**:

**15.1.** По време на дългите учебни часове нашият приятел Умко от училището в село Каръшко, което в наша чест е наречено „Програмистите на България“, се забавлявал като измислял различни игри с числа. Случайно забелязал, че от едно петцифрено число, чрез разместване на цифрите му, могат да се получат много различни числа и дори ги преброил. Те били 120. Все пак написването им било досадна работа, дори когато ти е скучно в час. Умко решил, че не го интересуват всички тези числа, а само най-малкото и най-голямто от тях. Той започнал да разглежда различни петцифрени числа за всяко от тях да намира разликата между най-голямото и най-малкото число, получени от неговите цифри. Скоро и тази работа му омръзнала, но все пак искал на всяка цена да знае тази разлика за всяко едно петцифрено число. Сега вече само вие можете да му помогнете като съставите алгоритъм и напишете програма на C++, която да въвежда от клавиатурата на компютъра цяло петцифрено число N и извежда на екрана разликата между най-малкото и най-голямото измежду числата получени, като се разместят цифрите на N.

**ИЗИСКВАНИЯ КЪМ ПРОГРАМАТА**

**Желан изход:** Извежда на екрана разликата между най-малкото и най-голямото измежду числата като се разместат цифрите на **N**.

**Необходими входни данни:** В конзолата се въвежда цяло петцифрено число **N**.

**Необходима обработка:**

1. **N** се разделя на съставните си цифри чрез делене на последната цифра на N с остатък и вкарване на тази цифра в едномерен масив. Последната цифра се премахва чрез делене без остатък. Всичко това се извършва в цикъл с параметър .
2. Цифрите от едномерния масив се подреждат в низходящ ред чрез два цикъла с параметър и условен оператор **if**.
3. Подредените по големина цифри се преобразуват в най-голямото и най- малкото петцифрено число чрез умножаване на всяка от цифрите им със степени на 10 в цикъл с параметър.
4. Извежда се разлика между най-малкото и най-голямото от получените числа.

**ПРОЕКТИРАНЕ НА ПРОГРАМАТА**

C:\Users\User\AppData\Local\Microsoft\Windows\INetCache\Content.Word\KursovaZad1.png

**ТЕКСТ НА ПРОГРАМАТА**

#include "stdafx.h"

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

system("chcp 1251");

int N, nums[5], swap = 0, max = 0, min = 0;

cout << "Въведете цяло петцифрено число! \n"; cin >> N;

for (int i = 0; i < 5; i++)

{

nums[i] = N % 10;

N /= 10;

}

for (int j = 0; j < 5; j++)

{

for (int i = 0; i < 5 - j; i++)

{

if (nums[i] < nums[4 - j])

{

swap = nums[4 - j];

nums[4 - j] = nums[i];

nums[i] = swap;

}

}

}

for (int i = 0; i<5; i++)

{

max += nums[i] \* pow(10, 4 - i);

min += nums[i] \* pow(10, i);

}

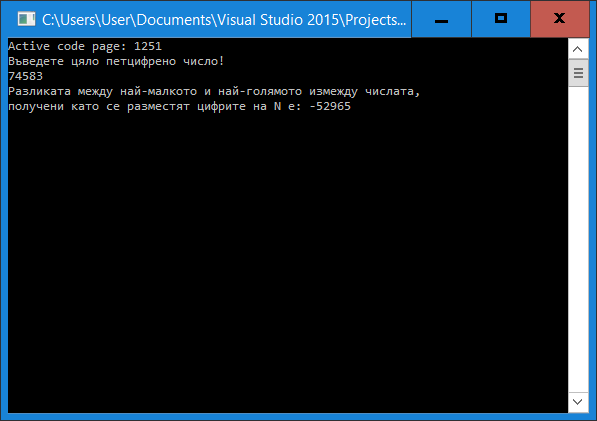
cout << "Разликата между най-малкото и най-голямото числата,"<<endl<<"получени като се разместят цифрите на N е: " << min - max;

system("pause>null");

return 0;

}

**ТЕСТВАНЕ**



**ЗАДАНИЕ II**

Зададена е окръжност (абсциса и ординатата на центъра и радиус) и m точки в равнината (абсциса и ордината). Съставете алгоритъм и напишете програма за намиране най-отдалечената точка от центъра на окръжността.

**ИЗИСКВАНИЯ КЪМ ПРОГРАМАТА**

**Желан изход:** Извежда на екрана координатите на най-отдалечената точка от центъра на окръжността.

**Необходими входни данни:** Координатите на центъра на окръжността, радиуса на окръжността, броя точки в равнината – **m**, координатите на точките в равнината.

**Необходима обработка:**

1. Точките в равнината се въвеждат в двумерен масив A[m][2], чрез цикъл с параметър, където m е поредността на точката + 1, а А[m][0] и A[m][1] са нейните координати.
2. Ако абсциса или ордината на центъра О на окръжността са различни от 0, те се „изместват“ в центъра на координатната система и това изместване се компенсира като към координатите на всяка от точките в равнината (чрез цикъл с параметър) се прибавят отрицателната стойност на абциса (**ex**) и ордината (**ey**) на центъра. Координатите на променените точки се запазват в масив a[m][2], така че a[i][0] = A[i][0] + ex, a[i][ 1] = A[i][1] + ey.
3. Чрез прилагане на Питагоровата теорема за △О1 a[i][0] a[i][1] (където О1 е центърът на координатната система) се намира хипотенузата на тръгълника, която се явява разстояние на a[i][0], a[i][1] от центъра на координатната система. Най-отдалечената точка се намира чрез сравняване в **if** оператор и цикъл с параметър. Когато се намери най-отдалечената точка, нейният индекс **i** се запазва в променливата place.
4. Извеждат се координатите на непроменена точка A[place][0], A[place][1].

**ПРОЕКТИРАНЕ НА ПРОГРАМАТА**

**C:\Users\User\Downloads\KursovaZadacha_Zad1 (2).png**

**ТЕКСТ НА ПРОГРАМАТА**

#include "stdafx.h"

#include <iostream>

#include <math.h>

using namespace std;

int main() {

system("chcp 1251");

int O[2], R, m, max\_dist = 0, place = 0, ex = 0, ey = 0;

cout << "Въведете координатите на центъра на окръжността: \n";

cout << "x= "; cin >> O[0];

cout << "y= " ; cin >> O[1];

cout << "Въведете размера на радиуса на окръжността: \n"; cin >> R;

cout << "Въведете броя на точките в равнината: \n"; cin >> m;

int\*\* A = new int\*[m];

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

A[i] = new int[2];

}

int\*\* a = new int\*[m];

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

a[i] = new int[2];

}

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

cout << "Въведете координатите на точка номер " << i + 1 << ": \n";

cout << "x= "; cin >> A[i][0];

cout << "y= "; cin >> A[i][1];

}

ex = -O[0];

ey = -O[1];

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

a[i][0] = A[i][0] + ex;

a[i][1] = A[i][ 1] + ey;

}

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

if (sqrt(pow(a[i][0], 2) + pow(a[i][1], 2)) > max\_dist) {

max\_dist = sqrt(pow(a[i][0], 2) + pow(a[i][1], 2));

place = i;

}

}

if (m != 0) {

cout << "Най-отдaлечената точка от центъра е с координати: x= " << A[place][0] << ", y= " << A[place][1];

}

else {

cout << "Няма точки в равнината.";

}

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

delete[] A[i];

}

delete[] A;

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

delete[] a[i];

}

delete[] a;

system("pause>null");

return 0;

}

**ТЕСТВАНЕ**

