

**ІПЗ-19к/1**

**Безугла Кіра**

**Інндувідуальні завдання**

**Завдання 1 (3)**

Задано послідовність цілих чисел a0, a1, …, an-1. З’ясувати, яке число зустрічається раніше – додатне чи від’ємне.

#include<iostream>

#include<time.h>

#include<conio.h>

#include<iomanip>

#include<vector>

using namespace std;

int main()

{

srand((unsigned)time(NULL));

int m,n = 10;

vector<int> ivec;

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

{

m = rand() % 21 - 10;

ivec.push\_back(m);

}

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

{

if (ivec[i] != 0)

{

if (ivec[i] < 0)

{

cout << "first number < 0: " << ivec[i] << endl;

break;

}

else

{

cout << "first number > 0: " << ivec[i] << endl;

break;

}

}

}

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

{

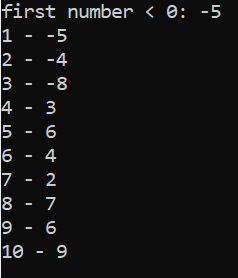
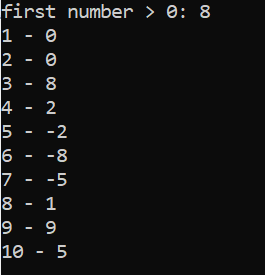
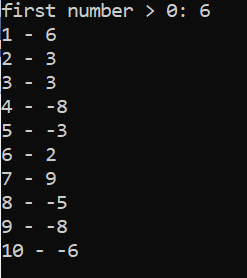
cout << i + 1 << " - " << ivec[i] << endl;

}

\_getch();

return 0;

}



**Завдання 2 (30)**

Дана послідовність цілих чисел a0, a1, …, an–1. Утворити нову послідовність, вилучивши з вихідної члени, що дорівнюють min (a0, a1, …, an–1).

#include<iostream>

#include<time.h>

#include<conio.h>

#include<iomanip>

#include<vector>

using namespace std;

void inputVector(vector<int>&, int n);

void outputVector(vector<int> ivec);

int main()

{

int n = 100;

vector<int> ivec;

inputVector(ivec, n);

outputVector(ivec);

/\*srand((unsigned)time(NULL));

int m;

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

{

m = rand() % 21 - 10;

ivec.push\_back(m);

}

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

{

cout << ivec[i] << " ";

cout << endl;

}\*/

//знаходження мінімального елементу

int min = 0;

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

{

if (min > ivec[i])

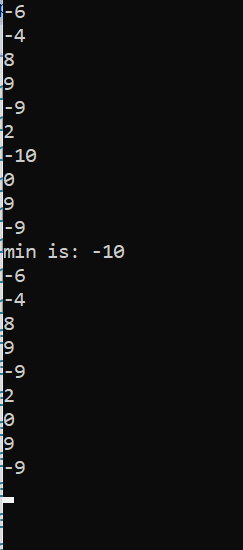
{

min = ivec[i];

}

}

cout << "min is: " << min << endl;



//видалення мінімальних елементів

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

{

if (ivec[i] == min)

{

ivec.erase(ivec.begin()+i);

}

}

outputVector(ivec);

\_getch();

return 0;

}

void inputVector(vector<int>& ivec, int n)

{

srand((unsigned)time(NULL));

int m;

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

{

m = rand() % 21 - 10;

ivec.push\_back(m);

}

}

void outputVector(vector<int> ivec)

{

int n = ivec.size();

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

{

cout << ivec[i] << " ";

cout << endl;

}

}

**Завдання 3 (62)**

Задано масив А. Знайти довжину найдовшої послідовності розташованих поспіль нульових елементів масиву.

#include <iostream>

//#include <string>

using namespace std;

int main()

{

//string s;

int mas[100] = { 2,3,4,2,3,2,4,0,0,0,

0,0,2,3,4,2,0,0,2,3,

4,0,2, 0,0,3,2,4,3,4,

0,2,3,0,4,2,3,2,3,4,

3,2,2,4,4,0,7,0,0,5,

0,0,0,0,0,0,0,0,0,0,

0,6,5,4,5,6,4 };

int count = 0, k = 0;

int max = -1;

//cout << "Input string = > \n";

//getline(cin, s);

//s = " " + s;

int len = 67;

cout << "Array is:" << endl;

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

{

cout << mas[i] << endl;

}

cout << "\n========\n";

for (int i = 0; i < len - 1; i++)

{

if (mas[i] == 0 && mas[i + 1] != 0)

{

k = 0;

count++;

}

else {

k++;

if (mas[i] != 0 && mas[i + 1] == 0)

{

if (max < k)

{

max = k;

}

}

}

}

cout << "\n Array has " << count <<

" consecutive numbers \n";

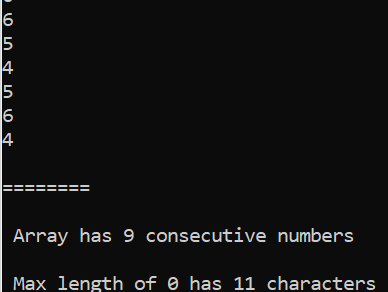
cout << "\n Max length of 0 has " << max <<

" characters\n";

system("pause");

return 0;

}



**Завдання 4 (77)**

Нехай дано дві неспадаючі послідовності дійсних чисел a0≤a1≤…≤аn–1 і b0≤b1≤…≤bm–1. Слід знайти місця, на які потрібно вставити елементи 2-ї послідовності в 1-у послідовність так, щоб нова послідовність залишилася неспадаючою

#include<iostream>

#include<time.h>

#include<conio.h>

#include<iomanip>

#include<vector>

#include<algorithm>

using namespace std;

int main()

{

srand((unsigned)time(NULL));

int n, m, i, j, k = 0;

cout << "please input length of FIRST sequence: ";

cin >> n;

cout << "please input length of SECOND sequence: ";

cin >> m;

vector<int> seq1, seq2;

vector<int> seq3;

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

{

seq1.push\_back(rand() % 10);

}

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

{

seq2.push\_back(rand() % 10);

}

//cout << "\nFIRST sequence is:\n";

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

//{

// cout << seq1[i] << " ";

//}

//cout << "\nSECOND sequence is:\n";

//for (i = 0; i < m; i++)

//{

// cout << seq2[i] << " ";

//}

sort(seq1.begin(), seq1.end());

sort(seq2.begin(), seq2.end());

cout << "\nafter sorting:";

cout << "\nFIRST sequence is:\n";

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

{

cout << seq1[i] << " ";

}

cout << "\nSECOND sequence is:\n";

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

{

cout << seq2[i] << " ";

}

i = 0; j = 0;

while (i < n && j < m)

{

if (seq1[i] > seq2[j])

{

seq3.push\_back(seq2[j]);

j++; k++;

}

else

{

if ((seq1[i] < seq2[j]))

{

seq3.push\_back(seq1[i]);

i++; k++;

}

else

{

seq3.push\_back(seq1[i]);

seq3.push\_back(seq2[j]);

i++; j++; k++; k++;

}

}

}

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

{

seq3.push\_back(seq1[k]);

}

for (k = j; k < m; k++)

{

seq3.push\_back(seq2[k]);

}

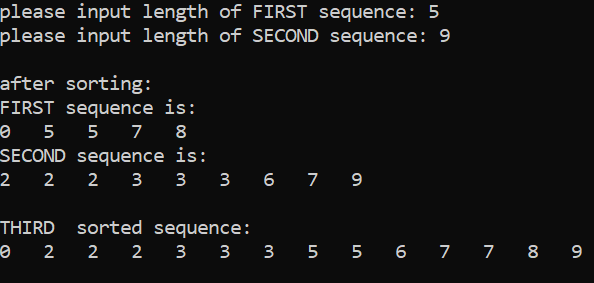
cout << "\n\nTHIRD sorted sequence:" << endl;

for (i = 0; i < seq3.size(); i++)

{

cout << seq3[i] << " ";

}



\_getch();

return 0;

}

**Завдання 5 (91)**

Визначити, чи є задана ціла квадратна матриця n-го порядку симетричною відносно головної діагоналі.

#include<iostream>

#include<time.h>

#include<conio.h>

#include<iomanip>

#include<vector>

using namespace std;

int main()

{

srand((unsigned)time(NULL));

const int n = 5;

int matrix[n][n]; /\*= {

{1,0,0,1,1},

{0,1,1,1,0},

{0,1,0,0,1},

{1,1,0,1,1},

{1,0,1,1,1}

};\*/

int i, j;

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

{

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

{

matrix[i][j] = rand() % 2;

cout << matrix[i][j] << " ";

}

cout << endl;

}

//порівняння

bool prap = true;

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

{

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

{

if (matrix[i][j] != matrix[j][i])

{

prap = false;

cout << "no symmetry :(" << endl;

break;

}

}

} if (prap)

{

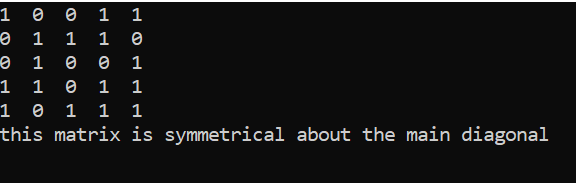
cout << "this matrix is symmetrical about the main diagonal" << endl;

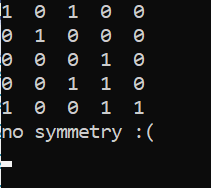
}

\_getch();

return 0;

}





**Завдання 6 (113)**

Задано квадратну матрицю. Отримати транспоновану матрицю, тобто матрицю, де стовпці і рядки міняються місцями.

#include<iostream>

#include<time.h>

#include<conio.h>

#include<iomanip>

#include<vector>

using namespace std;

e

int main()

{

srand((unsigned)time(NULL));

const int n = 5;

int matrix1[n][n], matrix2[n][n];

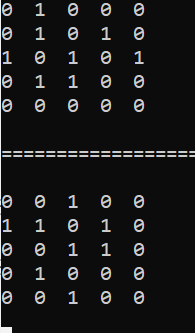
int i, j;

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

{

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

{

 matrix1[i][j] = rand() % 2;

cout << matrix1[i][j] << " ";

}

cout << endl;

}

cout << "\n=====================\n\n";

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

{

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

{

matrix2[j][i] = matrix1[i][j];

cout << matrix2[j][i] << " ";

}

cout << endl;

}

\_getch();

return 0;

}

**Завдання 7 (130)**

Задано матрицю з дійсними елементами порядку n. Перетворити її за правилом: останній рядок зробити останнім стовпцем і навпаки - останній стовпець зробити останнім рядком.

#include<iostream>

#include<time.h>

#include<conio.h>

#include<iomanip>

#include<vector>

using namespace std;

int main()

{

srand((unsigned)time(NULL));

const int n = 5;

double matrix[n][n];

int i, j;

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

{

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

{

matrix[i][j] = rand() % 8 \* 0.250;

cout << matrix[i][j] << " ";

}

cout << endl;

}

cout << "\n==============================\n\n";

double tmp;

//remaking

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

{

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

{

if (i == n - 1 )

{

tmp = matrix[i][j];

matrix[i][j] = matrix[j][i];

matrix[j][i] = tmp;

}

}

}

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

{

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

{

cout << matrix[i][j] << " ";

}

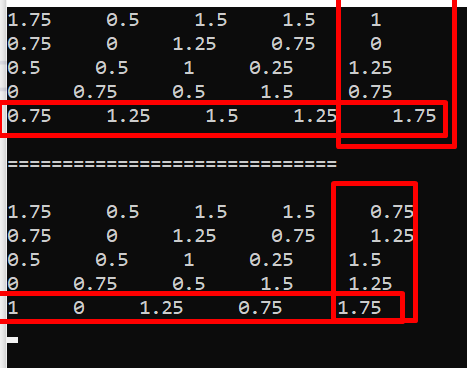
cout << endl;

}

\_getch();

return 0;

}



**Завдання 8 (151)**

На площині задано координати n точок. Скласти матрицю, елементами якої є відстань між кожною парою точок

#include<iostream>

#include<time.h>

#include<conio.h>

#include<iomanip>

#include<vector>

#define STOP '\*'

struct Point

{

char name;

double x;

double y;

};

using namespace std;

void ioPoint(Point&);

double Distance(Point, Point);

int main()

{

srand((unsigned)time(NULL));

Point p;

int i, j;

char k;

cout << "please input name of the point or \* to finish: ";

cin >> k;

vector<Point> dotes;

while (k != STOP)

{

p.name = k;

ioPoint(p);

dotes.push\_back(p);

cout << "please input name of the point or \* to finish: ";

cin >> k;

}

cout << endl << endl << "List of dotes" << endl;

for (i = 0; i < dotes.size(); i++)

{

cout << dotes[i].name << ": (" << dotes[i].x << ", " << dotes[i].y << ")\n";

}

cout << "\n==============================\n\n";

int n = dotes.size();

double\*\* matrix = new double\* [n];

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

{

matrix[i] = new double[n];

}

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

{

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

{

matrix[i][j] = Distance(dotes[i], dotes[j]);

cout << matrix[i][j] << setw(10);

}

cout << endl;

}

cout << "\n==============================\n\n";

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

{

delete[] matrix[i];

}

delete[] matrix;

\_getch();

return 0;

}

void ioPoint(Point& A)

{

A.x = rand() % 11 - 5.0;

A.y = rand() % 11 - 5.0;

cout << "x" << " " << "y" << endl;;

cout << A.x << " " << A.y << endl;;

}

double Distance(Point A, Point B)

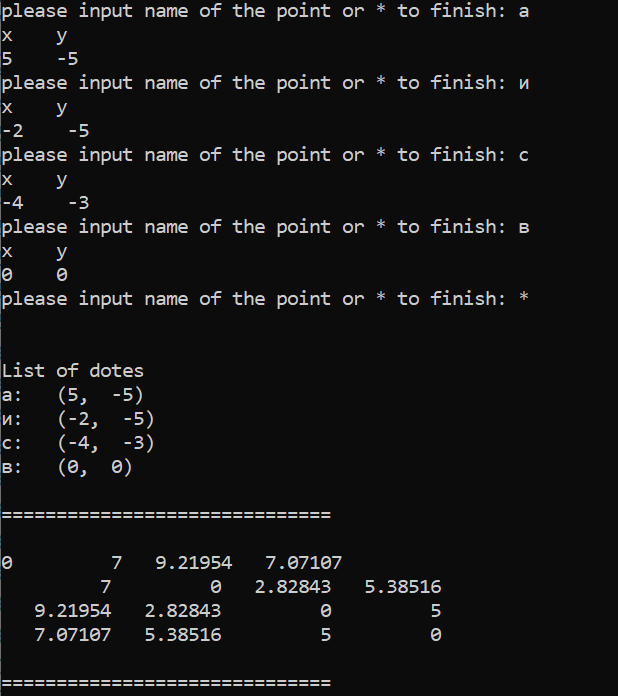
{

double d;

d = sqrt(((A.x - B.x) \* (A.x - B.x)) + ((A.y - B.y) \* (A.y - B.y)));

return d;

}



**Завдання 9 (170)**

Написати програму, яка знаходить і виводить всі чотиризначні числа вигляду abcd , для яких виконується ab - cd = a + b + c + d,причому a, b, c, d – різні цифри.

#include <iostream>

#include <string>

using namespace std;

bool CheckIn1(int n);

bool CheckIn2(int n);

void Separation(int n, int\* a, int\* b, int\* c, int\* d);

int main()

{

int n;

cout << "this strange numbers:" << endl;

for (n = 1000; n < 10000; n++)

{

if (CheckIn1(n))

{

if (CheckIn2(n))

{

cout << n << endl;

}

}

}

system("pause");

return 0;

}

void Separation(int n, int\* a, int\* b, int\* c, int\* d)

{

\*d = n % 10;

n /= 10;

\*c = n % 10;

n /= 10;

\*b = n % 10;

n /= 10;

\*a = n % 10;

n /= 10;

}

bool CheckIn1(int n)

{

int a = 0, b = 0, c = 0, d = 0;

Separation(n, &a, &b, &c, &d);

if (a != b && a != c && a != d && b != c && b != d)

{

return true;

}

else

{

return false;

}

}

bool CheckIn2(int n)

{

int a = 0, b = 0, c = 0, d = 0;

Separation(n, &a, &b, &c, &d);

if ((a \* 10 + b) - (c \* 10 + d) == a + b + c + d)

{

return true;

}

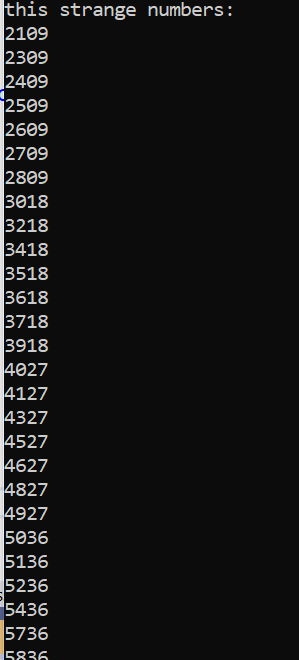
else

{

return false;

}

}



**Завдання 10 (200)**

Звести число у степінь (основа і показник ступеню записано в P-ічній системі числення). Відповідь видати в P-ічній і десятковій системах числення.

**Завдання 11 (205)**

Знайдіть суму цифр заданого натурального числа

#include <iostream>

#include <string>

using namespace std;

int Recurs(int, int);

int main()

{

int n, sum = 0;

cout << "please input natural number: ";

cin >> n;

cout << endl;

cout<< Recurs(n, sum);

cout << endl;

system("pause");

return 0;

}

int Recurs(int n, int sum)

{

if (n != 0)

{

sum += n % 10;

n /= 10;

Recurs(n, sum);

}

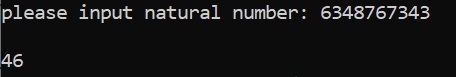
else

{

return sum;

}

}



**Завдання 12 (215)**

Задано рядок, що закінчується крапкою. Підрахувати, скільки в ньому слів.

#include <iostream>

#include <string>

using namespace std;

int main()

{

string s;

int count = 0;

cout << "Enter a string = > \n";

getline(cin, s);

s = " " + s;

int len = s.length();

for (int i = 0; i < len - 1; i++)

{

if (s[i] == ' ' && s[i + 1] != ' ')

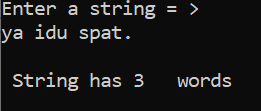
count++;

}

cout << "\n String has " << count << " words\n";

return 0;

}



**Ігра**

Ігра «фікс і ран». Ведучий вводить кількість літер та слово. Гравець пропонує слова такої ж довжини. Програма порівнює слова і виводить кількість «фіксів» - літер, які в обох словах на одному й тому ж місці та «ранів» - літер, що повторюються у другому слові, але стоять не на своєму місці.

Задача гравця, використовуючи щонайменшу кількість слів вгадати слово ведучого. Програма підраховує кількість слів, що використав гравець для вгадування

#include<iostream>

#include<time.h>

#include<conio.h>

#include<iomanip>

#include<vector>

#include<Windows.h>

using namespace std;

bool CheckSameSymbols(char[], int, int, vector<int>);

bool CheckLength(char[], int);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//cout << " а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ь ы ъ э ю я " << endl

// << " А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ы Ъ Э Ю Я" << endl;

int k, n;

cout << "какое количество букв в слове? ";

cin >> k;

n = k + 1;

char\* main = new char[n];

char\* word = new char[n];

int fix = 0, run = 0, count = 0;

cout << "ведущий, введите слово: ";

cin >> main;

if (CheckLength(main, k))

// cout << main;

{

cout << "игрок, вводите слова." << endl <<

"программа выведет количество fix и run" <<

endl << "введите правильное слово для победы" <<

endl;

vector<int> fixIter;

while (word != main)

{

if (word != main)

{

cout << "слово: ";

cin >> word;

if (CheckLength(word, k))

{

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

{

if (main[i] == word[i])

{

fix++;

fixIter.push\_back(i);

}

}

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

{

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

{

if (main[i] == word[j] && i != j)

{

if (CheckSameSymbols(word, j, k, fixIter))

{

run++;

}

}

}

}

cout << fix << " fix, " << run << " run" << endl;

}

}

else

{

break;

}

fixIter.clear();

fix = 0; run = 0;

count++;

}

}

cout << "поздравляю, это правильное слово): " << main << endl

<< "вам понадобилось " << count << "слов, чтобы отгадать" << endl;

delete[] main;

delete[] word;

\_getch();

return 0;

}

bool CheckSameSymbols(char word[], int j, int k, vector<int> fix)

{

int i = 0;

if (!fix.empty()) {

while (fix[i] != fix.size())

{

if (word[j] != word[fix[i]])

{

return true;

}

else

{

return false;

}

i++;

}

}

else

{

return true;

}

}

bool CheckLength(char word[], int k)

{

int i = 0;

while (word[i] != 0)

{

i++;

}

if (i != k)

{

cout << "неправильно введено слово" << endl;

return false;

}

else

{

return true;

}

}