National University of Computer and Emerging Sciences



Assignment 1

Object Oriented Programming Lab

|  |  |
| --- | --- |
| Name | Muhammad Zain |
| Roll No. | 19F-0228 |
| Course INSTRUCTOR | Sir Danish Shehzad |
| Lab INSTRUCTOR | Sir Mughees Ismail |
| Semester | Spring 2020 |

# Task 23:

## Source Code;

#include<iostream>

using namespace std;

int row1, col1, row2, col2, choice;

int\*\* array1 = new int\*[row1];

int\*\* array2 = new int\*[row2];

int\*\* array3 = new int\*[row1];

void subtraction()

{

if (row1 != row2 && col1 != col2)

{

cout << "Matrix Subtraction is not possible:" << endl;

cout << "no of coloumns of 1st matrix should be equal to no of rows of second matrix" << endl;

}

cout << "\t First Matrix:" << endl;

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

cout << "input entries" << endl;

cin >> array1[i][j];

}

}

cout << "\tSecond Matrix:" << endl;

for (int i = 0; i < row2; ++i)

{

for (int j = 0; j < col2; ++j)

{

cout << "input the entries :";

cin >> array2[i][j];

}

}

cout << "First Matrix is:" << endl;

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

cout << array1[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "Second Matrix is:" << endl;

for (int i = 0; i < row2; ++i)

{

for (int j = 0; j < col2; ++j)

{

cout << array2[i][j] << " ";

}

cout << endl;

}

cout << endl;

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

array3[i][j] = array1[i][j] - array2[i][j];

}

}

cout << "After Subtraction matrix is:" << endl;

{

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

cout << array3[i][j] << " ";

}

cout << endl;

}

}

}

void addition()

{

{

if (row1 != row2 && col1 != col2)

{

cout << "Matrix addition is not possible:" << endl;

cout << "no of coloumns of 1st matrix should be equal to no of rows of second matrix" << endl;

}

cout << "Enter elements for First Matrix:" << endl;

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

cout << "Enter Elemnet:";

cin >> array1[i][j];

}

}

cout << "Enter elements for Second Matrix:" << endl;

for (int i = 0; i < row2; ++i)

{

for (int j = 0; j < col2; ++j)

{

cout << "Enter Elemnet:";

cin >> array2[i][j];

}

}

cout << "First Matrix is:" << endl;

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

cout << array1[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "Second Matrix is:" << endl;

for (int i = 0; i < row2; ++i)

{

for (int j = 0; j < col2; ++j)

{

cout << array2[i][j] << " ";

}

cout << endl;

}

cout << endl;

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

array3[i][j] = array1[i][j] + array2[i][j];

}

}

cout << "After Addition matrix is:" << endl;

{

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

cout << array3[i][j] << " ";

}

cout << endl;

}

}

}

}

void multiplication()

{

if (col1 != row2)

{

cout << "Matrix Multiplication is not possible." << endl;

cout << "no of coloumns of 1st matrix should be equal to no of rows of second matrix" << endl;

exit(0);

}

cout << "First Matrix:" << endl;

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

cout << "input entries:";

cin >> array1[i][j];

}

}

cout << "Second Matrix:" << endl;

for (int i = 0; i < row2; ++i)

{

for (int j = 0; j < col2; ++j)

{

cout << "input entries:";

cin >> array2[i][j];

}

}

cout << "First Matrix :" << endl;

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col1; ++j)

{

cout << array1[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "Second Matrix :" << endl;

for (int i = 0; i < row2; ++i)

{

for (int j = 0; j < col2; ++j)

{

cout << array2[i][j] << " ";

}

cout << endl;

}

cout << endl;

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col2; j++)

{

array3[i][j] = 0;

for (int k = 0; k < row2; k++)

{

array3[i][j] = array3[i][j] + array1[i][k] \* array2[k][j];

}

}

}

cout << "Multipliction of matrix is:" << endl;

for (int i = 0; i < row1; ++i)

{

for (int j = 0; j < col2; ++j)

{

cout << array3[i][j] << " ";

}

cout << endl;

}

}

int main()

{

cout << "No of Rows for matrix 1= ";

cin >> row1;

cout << "Enter No of Coloumn for matrix 1 = ";

cin >> col1;

cout << "Enter No of Row for matrix2 = ";

cin >> row2;

cout << "Enter No of Coloumn for matrix2 = ";

cin >> col2;

for (int i = 0; i < row1; ++i)

{

\*(array1 + i) = new int[col1];

}

for (int i = 0; i < row2; ++i)

{

\*(array2 + i) = new int[col2];

}

for (int i = 0; i < row1; ++i)

{

\*(array3 + i) = new int[col2];

}

cout << "Press 1 for Addition " << endl;

cout << "Press 2 for subtraction" << endl;

cout << "press 3 for multiplication" << endl;

cin >> choice;

if (choice == 1)

{

addition();

}

else if (choice == 2)

{

subtraction();

}

else if (choice == 3)

{

multiplication();

}

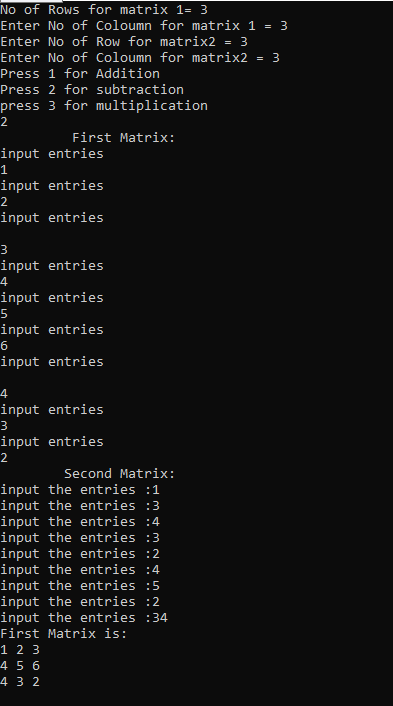
else

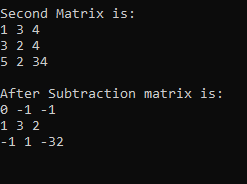
cout << "Error :-(" << endl;

return 0;

}

## Snip;





# Task 22;

## Source code;

#include <iostream>

using namespace std;

void FunctionForPerfect(int Check)

{

{

int Perfects = 0;

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

{

if (Check % i == 0)

{

Perfects = Perfects + i;

}

}

if (Check == Perfects)

{

cout << "Perfect Number = " << Check << endl;

cout << "Divisor Is = ";

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

{

if (Check % i == 0)

{

cout << i << " ";

}

}

}

}

}

int main()

{

for (int i = 1; i <= 1000; i++)

{

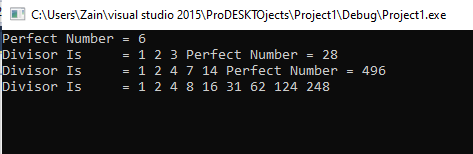
FunctionForPerfect(i);

}

system("pause>0");

}

## Snip;



# Task 20;

## Source code;

#include<iostream>

using namespace std;

int factorialrecursion(int a)

{

if (a == 1 || a == 0)

return 1;

else

return(a \* factorialrecursion(a - 1));

}

int Power(int& Prime, int& integer)

{

int Result, ForMode, number = integer;

Result = factorialrecursion(integer);

for (int i = 1; i <= factorialrecursion(number); i++)

{

ForMode = pow(Prime, i);

if (ForMode % Result == 0)

{

return i;

}

}

}

int main()

{

int Prime = 0, integer = 0;

cout << "Enter Prime Number and its power as well" << endl;

cin >> Prime;

cin >> integer;

cout << "Prime = " <<Prime <<endl;

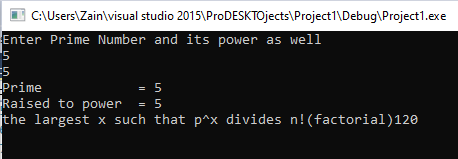
cout << "Raised to power = " <<integer<< endl;

cout << "the largest x such that p^x divides n!(factorial)" << Power(Prime, integer);

system("pause>0");

}

## Snip;



# Task 18;

## Source code;

#include <iostream>

#include <time.h>

using namespace std;

int main()

{const int length = 20;

int Array[length];

int storeRandom;

int Temperory = 0;

srand(time(0));//random number every time

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

{

Temperory = 0;

storeRandom = rand() % 51 + 25;

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

{

if (Array[j] == storeRandom)

{

Temperory++;

}

}

if (Temperory > 0)

{

i--;

}

else if (Temperory == 0)

{

Array[i] = storeRandom;

}

}

cout << "Enteries in the array" << endl;

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

{

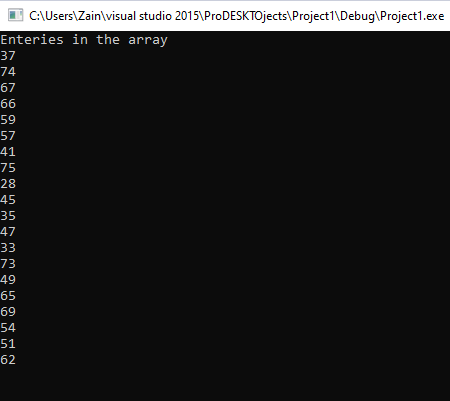
cout << Array[i] << endl;

}

system("pause>0");

}

## Snip;



# Task 1;

# Source code;

#include <iostream>

using namespace std;

int sortingFunction(int\* Pointer, int size)

{

int swap;

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

{

for (int j = 0; j < size - 1; j++)

{

if (\*(Pointer + j) > \* (Pointer + j + 1))

{

swap = \*(Pointer + j);

\*(Pointer + j) = \*(Pointer + j + 1);

\*(Pointer + j + 1) = swap;

}

}

}

return \*Pointer;

}

int output(int\* ptr, int size)

{

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

{

cout << \*(ptr + i) << endl;

}

return \*ptr;

}

int Check(int\* pointer, int size)

{

int Even = 0, odd = 0;

cout << "Even Entries = ";

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

{

if (\*(pointer + i) % 2 == 0)

{

Even++;

cout << \*(pointer + i) << endl;

}

}

cout << "There are total " << Even << " even entries" << endl << endl;

cout << "Odd Entries = ";

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

{

if (\*(pointer + i) % 2 != 0)

{

odd++;

cout << \*(pointer + i) << endl;

}

}

cout << "There are total " << odd << " odd entries" << endl << endl;

return 0;

}

int Input(int\* pointer, int size)

{

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

{

cout << "Enter value:";

cin >> \*(pointer + i);

}

return \*pointer;

}

int main()

{

int const size = 5;

int const size1 = 10;

int Array1[size], Array2[size], Array3[size1];

int\* One = Array1, \*Two = Array2, \*Three = Array3;

cout << "Input The Entries of first array " << endl;

Input(One, size);

sortingFunction(One, size);

cout << "After Sorting First array = ";

output(One, size);

cout << "Input The Entries of second array " << endl;

Input(Two, size);

sortingFunction(Two, size);

cout << "After Sorting second array = ";

output(Two, size);

for (int i = 0; i < size1; i++)

{

if (i < 5)

{

Array3[i] = Array1[i];

}

else

{

Array3[i] = Array2[i - 5];

}

}

cout << "After Sorting third array = ";

sortingFunction(Three, size1);

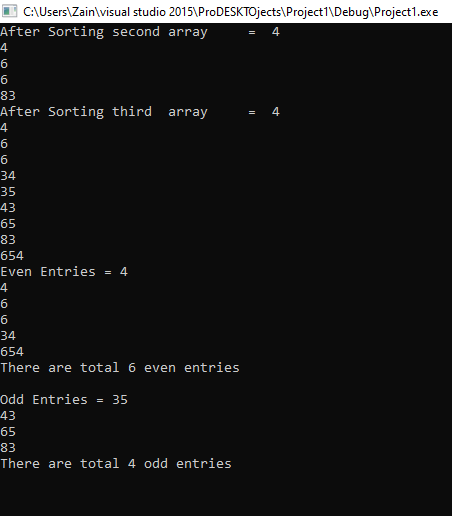
output(Three, size1);

Check(Three, size1);

system("pause>0");

}

# Snip;



# Task 8;

## Source code;

#include <iostream>

using namespace std;

int main()

{

int number;

cout << "Enter a number to check whether the number is even or odd"<<endl;

cin >> number;

if ((number / 2) != 0)

cout << "Even :-)" << endl;

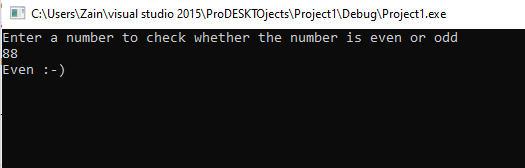
else

cout << "Odd :-)" << endl;

system("pause>0");

}

## Snip;



# Task 6;

## Source code;

#include<iostream>

#include <cmath>

using namespace std;

int LCM(int input1, int input2)

{

int gcd, lcm;

for (int i = 1; i <= input1 && i <= input2; ++i)

{

if (input1 % i == 0 && input2 % i == 0)

gcd = i;

}

lcm = (input1 \* input2) / gcd;

return lcm;

}

double HCFGCD(double INPUT1, double INPUT2)

{

while (INPUT1 != INPUT2)

{

if (INPUT1 > INPUT2)

{

INPUT1 -= INPUT2;

return INPUT1;

}

else

INPUT2 -= INPUT1;

return INPUT2;

}

}

double Power(double INPUT1, double INPUT2)

{

double result = 1;

result = pow(INPUT1, INPUT2);

return result;

}

double Addition(double INPUT1, double INPUT2)

{

double sum;

sum = INPUT1 + INPUT2;

return sum;

}

double Subtraction(double INPUT1, double INPUT2)

{

double sub;

sub = INPUT1 - INPUT2;

return sub;

}

double Multiplication(double INPUT1, double INPUT2)

{

double MULTIPLY;

MULTIPLY = INPUT1 \* INPUT2;

return MULTIPLY;

}

double Division(double INPUT1, double INPUT2)

{

double div;

div = INPUT1 / INPUT2;

return div;

}

int main()

{

start:

system("cls");

int choice;

cout << "Main Menu" << endl << endl;

cout << "Press 1 for LCM " << endl;

cout << "Press 2 for HCF or GCD" << endl;

cout << "Press 3 for Power " << endl;

cout << "Press 4 for Addition " << endl;

cout << "Press 5 for Subtraction" << endl;

cout << "Press 6 for Multiplication" << endl;

cout << "Press 7 for Division"<<endl;

cout << "Press 8 for Exit " << endl << endl;

cout << "Enter Choice"<<endl;

cin >> choice;

if (choice <= 8 && choice >= 1)

{

switch (choice)

{

case 1:

{

double input1, input2;

cout << "Enter First Number =";

cin >> input1;

cout << "Enter Second Number =";

cin >> input2;

cout << "The LCM Of These Numbers is=" << LCM(input1, input2) << endl;

}

case 2:

{

double input1, input2;

cout << "Enter First Number =";

cin >> input1;

cout << "Enter Second Number =";

cin >> input2;

cout << "The HCF or GCD Of These Two Number =" << HCFGCD(input1, input2) << endl;

}

case 3:

{

system("cls");

double input1, input2;

cout << "Enter Base Number =";

cin >> input1;

cout << "Enter Exponent Number =";

cin >> input2;

cout << "The Result Of These Two Values is =" << Power(input1, input2) << endl;

system("pause");

goto start;

break;

}

case 4:

{

system("cls");

double input1, input2;

cout << "Enter First Number =";

cin >> input1;

cout << "Enter Second Number =";

cin >> input2;

cout << "The Sum Of These Number Is =" << Addition(input1, input2) << endl;

system("pause");

goto start;

break;

}

case 5:

{

system("cls");

double input1, input2;

cout << "Enter First Number =";

cin >> input1;

cout << "Enter Second Number =";

cin >> input2;

cout << "The Subtraction Of These Two Number is =" << Subtraction(input1, input2) << endl;

system("pause");

goto start;

break;

}

case 6:

{

system("cls");

double input1, input2;

cout << "Enter First Number =";

cin >> input1;

cout << "Enter Second Number =";

cin >> input2;

cout << "The Multiplication Of These Two Numbers Is =" << Multiplication(input1, input2) << endl;

system("pause");

goto start;

break;

}

case 7:

{

system("cls");

double input1, input2;

cout << "Enter First Number =";

cin >> input1;

cout << "Enter Second Number =";

cin >> input2;

cout << "The Division Of These Numbers Is =" << Division(input1, input2) << endl;

system("pause");

goto start;

break;

}

case 8:

cout << "Thank you :-)";

system("pause");

goto end;

}

}

else

{

system("cls");

cout << "Wrong Entry" << endl;

system("pause");

goto start;

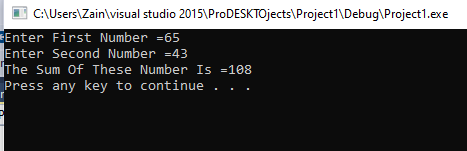
}

end:

system("pause>0");

}

## Snip;



# Task 12;

## SOURCE CODE;

#include<iostream>

using namespace std;

void MessageCheck(int& , int a[], int b[]);

int main()

{

int Text[250], sent[250], received;

cout << "Enter the message" << endl;

cin >> received;

MessageCheck(received, Text, sent);

system("pause>0");

}

void MessageCheck(int& Text, int sent[], int received[])

{

{

bool check = true;

cout << "Sent MEssage"<<endl;

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

{

cout << "Entr message in binary form ";

cin >> sent[i];

}

cout << "Received message" << endl;

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

{

cout << "Enter the Recieved message " << endl;

cin >> received[j];

if (sent[j] != received[j])

{

check = false;

cout << "failed to recieved a complete message" << endl;

}

}

if (check == true)

{

system("cls");

cout << "Sent Code Code Digit Copy" << endl;

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

{

cout << sent[i] << " " << received[i] << endl;

}

cout << " Message transmitted OK." << endl;

}

else

{

cout << "failed or crroupted" << endl;

}

}

## }

# Task 5

## Source code ;

#include <iostream>

using namespace std;

void Diamond(int input)

{

int counter = 0;

while (counter < 20)

{

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

{

for (int j = 1; j <= input - i; j++)

cout << " ";

for (int j = 1; j <= 2 \* i - 1; j++)

cout << "\*";

cout << endl;

}

for (int i = input - 1; i >= 1; i--)

{

for (int j = 1; j <= input - i; j++)

cout << " ";

for (int j = 1; j <= 2 \* i - 1; j++)

cout << "\*";

cout << endl;

}

system("cls");

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

{

for (int j = 1; j <= input - i; j++)

{

cout << " ";

}

for (int j = 1; j <= 2 \* i - 1; j++)

{

if (j == 1 || j == i \* 2 - 1)

cout << "\*";

else

cout << " ";

}

cout << endl;

}

for (int i = input - 1; i >= 1; i--)

{

for (int j = 1; j <= input - i; j++)

{

cout << " ";

}

for (int j = 1; j <= 2 \* i - 1; j++)

{

if (j == 1 || j == i \* 2 - 1)

cout << "\*";

else

cout << " ";

}

cout << endl;

}

system("cls");

counter++;

}

}

int main()

{

start:

int input;

cout << "Enter length Of Diamond ";

cin >> input;

Diamond(input);

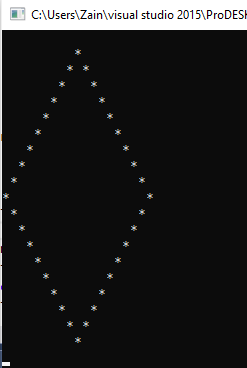
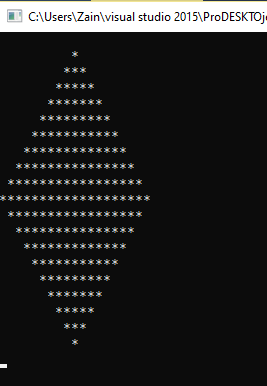
system("cls");

goto start;

system("pause>0");

}

## Snip;

# Task 15;

## Source;

#include<iostream>

using namespace std;

bool Function(char array1[], char array2[], int arrayone, int arraytwo) {

if (arraytwo == arrayone)

{

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

{

if (toupper(array1[i]) != toupper(array2[i]))

{

return false;

}

}

}

else {

return false;

}

return true;

}

int main()

{

start:

int const size1 = 5;

int const size2 = 5;

char array1[size1];

cout << "Input the entries in first arrays:";

for (int i = 0; i < size1; i++)

{

cin >> array1[i];

}

cout << "Enter the entries in second arrays:";

char array2[size2];

for (int i = 0; i < size2; i++)

{

cin >> array2[i];

}

if (Function(array1, array2, size1, size2) == true)

{

cout << "Equal Arrays :-) "<<endl;

}

else {

cout << "Arrays are not equal:-(";

}

system("pause>0");

}

## Snip;

# 

# TASK 10;

## SOURCE;

#include<iostream>

#include <string>

using namespace std;

string reverseString(string input)

{

char\* array = new char[size(input)];

for (int i = 0; i < size(input); i++)

{

array[i] = input[i];

}

for (int i = 0; i < size(input); i++)

{

input[i] = array[size(input) - i - 1];

}

return input;

}

int main()

{

string input;

cout << "Input String = ";

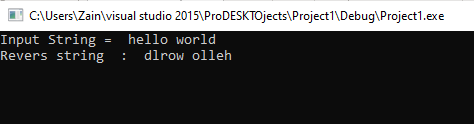
getline(cin, input);

cout << "Revers string : " << reverseString(input) << endl;

system("Pause>0");

}

## SNIP;



# Task 2;

## Source code;

#include<iostream>

#include<fstream>

#include<string>

using namespace std;

int main()

{

fstream in;

ofstream out;

string ch = "a", read = "a", guess;

int size, stat;

out.open("OPP\_Lab.txt");

if (!out.is\_open())

{

cout << "File not open ";

return 0;

}

while (ch != ".")

{

getline(cin, ch);

if (ch != ".")

out << ch;

}

out.close();

in.open("OPP\_Lab.txt");

if (!in.is\_open())

{

cout << "File not open ";

return 0;

}

else

cout << "file is open";

cout << "Enter the number you want to guess \n";

cin >> guess;

size = sizeof guess;

for (int a = 0; a < size; a++)

{

stat = static\_cast <int>(guess[a]);

if (stat < 97)

{

stat = stat - 32;

guess[a] = static\_cast <char>(stat);

}

}

while (read != ".")

{

in >> read;

if (guess == read)

{

cout << "Word found :" << guess << endl;

}

}

}

# Task 11;

## Source code;

#include<iostream>

using namespace std;

int main()

{

int rn1, cn1;//rn1 for rows of first coloumn plus cn1 for coloumns

int rn2, cn2;//rn2 for rows in 2nd matrix plus cn2 for coloumns

cout << "Enter the number of row and coloumn first Matrix " << endl;

cout << "note! it must be less then ten" << endl;

cout << "Rows For Matrix "; cin >> rn1;

cout << "Coloumns for Matrix"; cin >> cn1;

int First[10][10];

int Second[10][10];

int Multiplication[10][10];

cout << "Input the enteries of 1st matrix one by one" << endl;

for (int i = 0; i < rn1; i++)

{

for (int j = 0; j < cn1; j++)

cin >> First[i][j];

}

cout << "Enter the number of row and coloumn Second Matrix" << endl;

cout << "rows"; cin >> rn2;

cout << "col"; cin >> cn2;

for (int i = 0; i < rn2; i++)

{

for (int j = 0; j <cn2; j++)

cin >> Second[i][j];

}

cout << endl;

if (cn1 == rn2)

{

for (int i = 0; i < cn1; i++)

for (int j = 0; j < rn2; j++)

Multiplication[i][j] = First[i][j] \* Second[i][j];

cout << endl;

cout << "The Resultant Matrix is " << endl;

for (int i = 0; i < rn1; ++i)

for (int j = 0; j < cn2; ++j)

{

Multiplication[i][j] = 0;

}

// Multiplying matrix a and b and storing in array mult.

for (int i = 0; i < rn1; ++i)

for (int j = 0; j < cn2; ++j)

for (int k = 0; k < cn1; ++k)

{

Multiplication[i][j] += First[i][k] \* Second[k][j];

}

// Displaying the multiplication of two matrix.

cout << endl << "Output Matrix: " << endl;

for (int i = 0; i < rn1; ++i)

for (int j = 0; j < cn2; ++j)

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

}

else

{

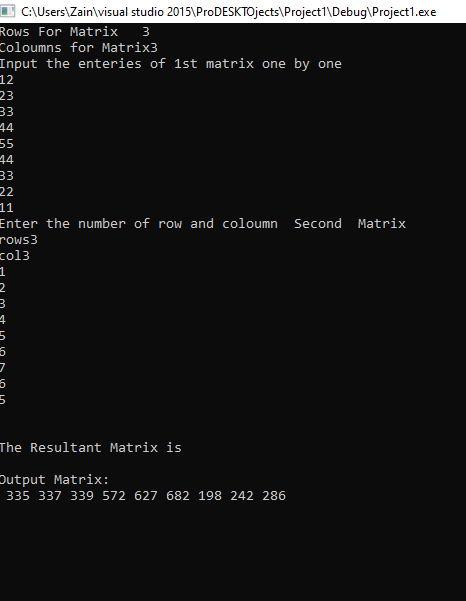
cout << "Columns of matrix 1 not equal to rows of matrix B, then matrices cannot be multiplied" << endl;

}

system("pause>0");

}

## Snip;



# Task 13;

## Source code;

#include<iostream>

#include <iomanip>

using namespace std;

int main()

{

int option, Select;

int ArrayOfSeats[100] = { 0 };

cout << "Welcome to Fly Emirates!" << endl;

cout << "Please type 1 for 'First Class'" << endl;

cout << "Please type 2 for 'Economy Class'" << endl;

cin >> option;

if (option == 1)

{

cout << "The Seats available are " << endl;;

for (int seat = 1; seat <= 50; seat++)//seat for first class

{

cout << seat << " ";//displaying available seats

}

cout << endl;

cout << "Choose your seat from 1 to 50" << endl;

cin >> Select;

cout << "Your Boarding Pass " << endl;

cout << "Class" << " Seat Number" << endl;;

cout << "First Class" << setw(13) << Select;

}

else if (option == 2)

{

cout << "The Seats available in Economy Class are " << endl;

for (int seat = 51; seat <= 100; seat++)

{

if (seat == 60 || seat == 70 || seat == 80 || seat == 90)

cout << seat << " " << endl;

else

{

cout << seat << " ";

}

}

cout <<endl<< "Select your seats from 51 to 100" << endl;

cin >> Select;

cout << "Your Boarding Pass " << endl;

cout << endl << endl;

cout << "Class" << " Seat Number" << endl;

cout << endl << endl;

cout << "Economy Class" << setw(13) << Select;

cout << endl << endl;

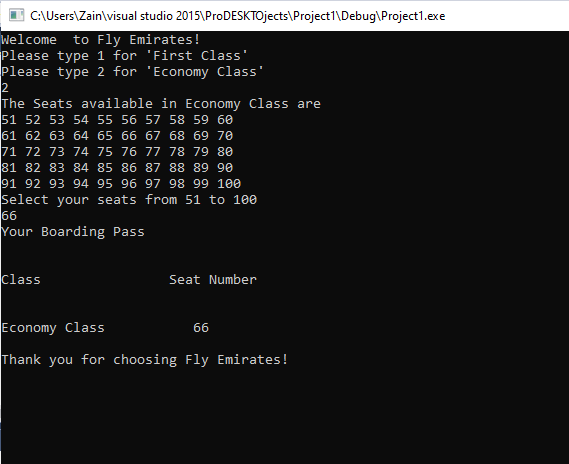
cout << "Thank you for choosing Fly Emirates!" << endl;

}

system("pause>0");

}

## Snip;



# Task 14;

# Source code;

#include <iostream>

#include <string>

using namespace std;

int main()

{

string names[5] = { "paper"," true ","soap ","floppy ","flower" };

//Show array

cout << "The unsorted values are\n";

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

{

cout << names[i] << " ";

}

//Sort array

string temp;

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

{

for (int j = i + 1; j < 5; j++)

{

if (names[j] < names[i])

{

temp = names[j];

names[j] = names[i];

names[i] = temp;

}

}

}

cout << "The sorted values are\n";

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

{

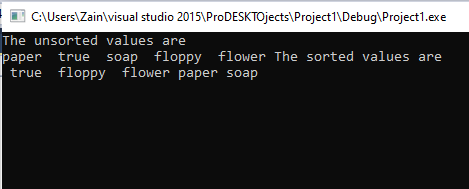
cout << names[i] << " ";

}

system("pause> 0");

}

## Snip;



# Task 21;

## Source code;

#include <iostream>

using namespace std;

void eqofLine(double& var1, double& var2)

{

double m = 2 \* var1 \* var2;

cout << "The Equation Of Straight Line Is" << endl;

cout << var1 << "x" << "+" << var2 << "y=\t" << m << "\n\n";

}

int main()

{

double var1, var2;

cout << "Enter the value of x axis"<<endl;

cin >> var1;

cout << "Enter the value of y-axis" << endl;

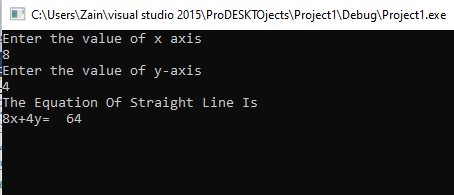
cin >> var2;

eqofLine(var1, var2);

system("pause>0");

}

## Snip;



# Task 3;

## Source code;

#include<iostream>

#include<fstream>

#include<string>

using namespace std;

int main()

{

fstream in;

ofstream out;

string rollnum, find;

cout << "Enter roll number to find :";

cin >> rollnum;

in.open("Record.txt");

if (!in.is\_open())

{

return 0;

}

while (!in.eof())

{

in >> find;

if (find == rollnum)

{

for (int a = 0; a < 3; a++)

{

cout << find << " ";

in >> find;

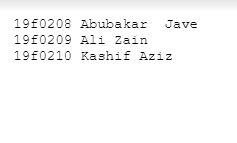
}

}

}

}

## Source file input



# Task 16

#include<iostream>

#include<cctype>

using namespace std;

int check(char array[], const int size)

{

int no = 0;

array[size];

cout << "Input the Entries"<<endl;

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

{

cin >> array[i];

}

for (int i = 0; i < 10; i++)

{

if (check(array[i]))

no++;

}

cout << "Digit in letters:";

system("pause>0");

}

int main()

{

const int size = 10;

char array[size];

cout << check(array, size);

system("pause");

return 0;

}

# Task 19

## Source code

#include <iostream>

using namespace std;

int main()

{

int array1[] = { 1, 5, 7, 5, 8, 9, 11, 12 };

int s1 = sizeof(array1) / sizeof(array1[0]);

cout << "Original array: ";

for (int i = 0; i < s1; i++)

cout << array1[i] << " ";

int i, sum = 12, check = 0;

cout <<endl<< "Array pairs whose sum equal to 12: ";

for (int i = 0; i<s1; i++)

for (int j = i + 1; j<s1; j++)

if (array1[i] + array1[j] == sum)

{

cout << "\n" << array1[i] << "," << array1[j];

check++;

}

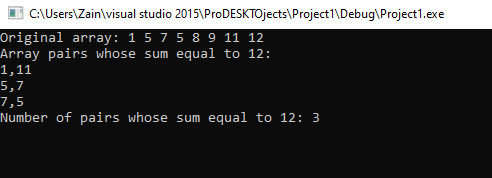
cout << endl<< "Number of pairs whose sum equal to 12: ";

cout << check;

system("pause>0");

}

## Snip;



# Task 17

## Souce code;

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

int const length = 10;

int Position;

int array1[length];

cout << "Input yhe entries";

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

{

cin >> array1[i];

}

int array2[length];

cout << "enteries are" << endl;

for (int i = 0; i < length; i = i + 1)

{

cout << array1[i] << endl;

}

cout << "Enter the no places you want to move : ";

cin >> Position;

if (Position >= 0)

{

for (int i = 0; i < Position; i = i + 1)

{

array2[0] = array1[length - 1];

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

{

array2[i + 1] = array1[i];

}

for (int i = 0; i < length; i = i + 1)

{

array1[i] = array2[i];

}

}

}

else {

Position = Position \* -1;

for (int i = 0; i < Position; i = i + 1)

{

array2[length - 1] = array1[0];

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

{

array2[i - 1] = array1[i];

}

for (int i = 0; i < length; i = i + 1)

{

array1[i] = array2[i];

}

}

}

cout << endl;

for (int i = 0; i < length; i = i + 1)

{

if (Position != 0) {

cout << array2[i] << endl;

}

else {

cout << array1[i] << endl;

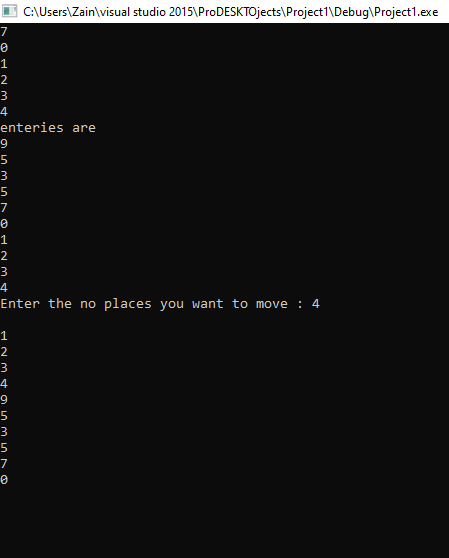
}

}

system("pause>0");

}

## Snip;



# Task 9;

## Source code;

#include<iostream>

#include<string>

using namespace std;

int Trans\_Matrix(int array[][3], int size);

void Display(int array[][3], int size);

void inputData(int array[][3], int size);

void Result\_Array(int A, int array[][3], int size);

void div(int A, int array[][3], int size);

int rollNO();

int main()

{

int const size = 3;

int Array[size][size];

int A = rollNO();

cout << "ENCODING :";

inputData(Array, size);

cout << "\n Actual Array :";

Display(Array, size);

cout << "\n Transpose Of The Array :";

Trans\_Matrix(Array, size);

Display(Array, size);

cout << "\n Inverted Array :";

invertedMatrix(Array, size);

Display(Array, size);

cout << "Resultant of the Array :";

Result\_Array(A, Array, size);

Display(Array, size);

cout << "\nDECODING :";

cout << " \nDivision ";

div(A, Array, size);

Display(Array, size);

cout << "\n Reinverted Array :";

invertedMatrix(Array, size);

Display(Array, size);

cout << "\nActual Decoded Array :";

Trans\_Matrix(Array, size);

Display(Array, size);

}

int Trans\_Matrix(int array[][3], int size)

{

int Array[3][3];

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

{

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

{

Array[j][i] = array[i][j];

}

}

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

{

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

{

array[i][j] = Array[i][j];

}

}

return 0;

}

void Display(int array[][3], int size)

{

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

{

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

{

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

}

cout << endl;

}

}

void inputData(int array[][3], int size)

{

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

{

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

{

cout << "Enter Value For Array :";

cin >> array[i][j];

}

}

}

void invertedMatrix(int array[][3], int size)

{

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

{

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

{

if (array[i][j] == 1)

{

array[i][j] = 0;

}

else {

array[i][j] = 1;

}

}

}

}

int rollNO()

{

string rollNo;

cout << "Enter your ROll NO.(\*\*F\*\*\*) :";

cin >> rollNo;

char A, B, c;

int num1, num2, num3, y, z;

int\* ptr1, \*ptr2;

A = rollNo.at(5);

B = rollNo.at(6);

c = rollNo.at(7);

num1 = static\_cast<int>(A) - 48;

num2 = static\_cast<int>(B) - 48;

num3 = static\_cast<int>(c) - 48;

y = num3 + num1 + num2;

return y;

}

void Result\_Array(int A, int array[][3], int size)

{

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

{

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

{

if (array[i][j] == 1)

{

array[i][j] = array[i][j] \* A;

}

}

}

}

void div(int A, int array[][3], int size)

{

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

{

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

{

if (array[i][j] == A)

{

array[i][j] = array[i][j] / A;

}

}

}

}