National University of Computer and Emerging Sciences



Assignment 1

Object Oriented Programming

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

# Task 1

## Source code:

#include <iostream>

using namespace std;

int main()

{

int array[5],sum;

int \*pointer;

cout<<"input elements for sum"<<endl;

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

{ cin>>array[i];

}

pointer=array;

cout<<"The first array elements are"<<" "<<endl;

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

{

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

}

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

{

sum+=pointer[i];

}

cout <<endl;

cout<<"The Sum of elements of an array ="<<sum;

system("pause>0");

}

## Snip:

A screenshot of a cell phone

Description automatically generated

# TASK 2

## SOURCE CODE:

#include<iostream>

using namespace std;

int main()

{

int array[10] = { 03,05,87,47,88,60,87,47,99,55 };

int sum=0;

int avg=0;

int \*pointer[10];

for (int i = 0; i < 10; i++)//iteration for pointing address

{

pointer[i] = &array[i];

}

for (int i = 0; i < 10; i++)//itreation for the sum of indexes

{

sum = sum + \*pointer [i];

avg = sum / 10;

}

cout << "Sum of elements = " << sum << endl << endl;

cout << "Average of elements = " << avg << endl;

return 0;

}

## SNIP

A screen shot of a social media post

Description automatically generated

# Task 3

# SOURCE CODE:

#include<iostream>

using namespace std;

int MODEFUNCTION(int \*pointer,int size);//mode function prototype

int main()

{

int array[10], size;

int \*pointer=array;

cout << "Total entries you want to enter: ";

cin >> size;

cout << "Input the entries" << endl;

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

{

cin >> array[i];

}

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

{

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

}cout<<endl;

cout<<"The Most repeated values is = "<<MODEFUNCTION(pointer,size)<<endl;

return 0;

}

int MODEFUNCTION(int \*pointer,int size)

{

int Mode = 0, Max=0, Counter=0;

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

{

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

{

if (pointer[i] == pointer[j])

{

if (pointer[i] > Max)

{

Max = pointer[i];

Mode = pointer[i];

}

}

}

}

if (Mode > 0)

return Mode;

else

return -1;

}

## Snip

A screenshot of a cell phone

Description automatically generated

# Task 5

#include<iostream>

#include<string.h>

using namespace std;

char check(char\* Name, char\* password)

{ int conditionmeet = 0;

int size = strlen(password);

if (size >= 6)

{

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

{

if (password[i]>=65 && password[i]<=90)//check for capital letters via ascii

{

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

{

if (password[i] >= 97 && password[i] <= 122)//check for lower case via ascii

{

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

{

if (password[i]>=48 && password[i]<=57)//check for numbers via ascii

{

conditionmeet = 1;

}

}

}

}

}

}

}

if (conditionmeet == 1)

{

cout << "Strong Password :-) "<<endl;

}

if (conditionmeet==0)

{

cout << "Your Password couldnot meet the requirements " <<endl;

}

return 0;

}

int main()

{

char Name[20], Password[20];

cout << "Input your Name = ";

cin.getline(Name, 20);//accepts space as well

cout << "Enter a Password which will meeat the following requirments "<<endl<<endl;

cout<<"1:The password should be at least six characters long"<<endl;

cout<<"2:The password should contain at least one uppercase and at least one lowercase letter"<<endl;

cout<<"3:The password should have at least one digit"<<endl;

cin.getline(Password, 20);

check(Name, Password);

return 0;

}

## Snip 1

A screenshot of a cell phone

Description automatically generated

## Snip 2

A screen shot showing a black background

Description automatically generated

# Task 6

# Source code

# include <iostream>

# include <cstring>

using namespace std;

int main ()

{

char check[MaxSize];

char \*condition = 0;

const int NameEnteries = 11;

const int MaxSize = 100;

char Phonebook[NameEnteries][MaxSize] = { "Becky Warren, 555-1223",

"Joe Looney, 555-0097",

"Geri Palmer, 555-8787",

"Lynn Presnell, 555-1212",

"Holly Gaddis, 555-8878",

"Sam Wiggins, 555-0998",

"Bob Kain, 555-8712",

"Tim Haynes, 555-7676",

"Warren Gaddis, 555-9037",

"Jean James, 555-4949",

"Ron Palmer, 555-2783"};

cout << "Enter full name or partial name "<<endl;

cin.getline(check, MaxSize);

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

{

condition = strstr(Phonebook[i], check);

if (condition != 0)

{

cout <<"Contact info"<<endl <<Phonebook[i];

}

}

else if (condition == 0)

cout << "invalid input :-("<<endl;

return 0;

}

# Snip:

A screenshot of a computer screen

Description automatically generated

# Task 7:

## Source code:

#include<iostream>

using namespace std;

int row1, col1, row2, col2, choice;

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

int\*\* array1 = new int\* [row2];

int\*\* array2 = 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 >> array[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 >> array1[i][j];

}

}

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

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

{

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

{

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

}

cout << endl;

}

cout << endl;

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

{

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

{

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

}

}

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

{

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

{

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

{

cout << array2[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 >> array[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 >> array1[i][j];

}

}

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

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

{

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

{

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

}

cout << endl;

}

cout << endl;

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

{

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

{

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

}

}

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

{

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

{

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

{

cout << array2[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 >> array[i][j];

}

}

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

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

{

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

{

cout << "input entries:";

cin >> array1[i][j];

}

}

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

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

{

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

{

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

}

cout << endl;

}

cout << endl;

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

{

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

{

array2[i][j] = 0;

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

{

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

}

}

}

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

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

{

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

{

cout << array2[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)

{

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

}

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

{

\*(array1 + i) = new int[col2];

}

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

{

\*(array2 + 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();

}

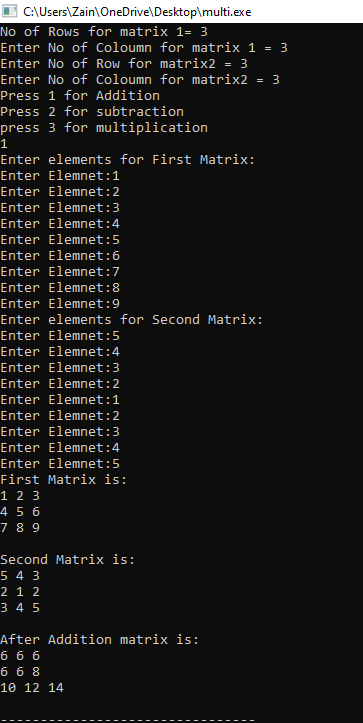
else

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

return 0;

}

## Snip;



# Task 8;

#include <iostream>

using namespace std;

void main()

{

int x = 4, y = 5, z = 6;

int i, j, k;

int\*\* array = new int\* [x];

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

{

array[i] = new int\* [y];

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

{

array[i][j] = new int[z];

}

}

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

{

cout << i << endl;

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

{

cout << endl;

for (k = 0; k < z; k++)

{

array[i][j][k] = (i \* y \* z) + (j \* z) + k;

cout << '\t' << array[i][j][k];

}

}

cout << endl << endl;

}

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

{

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

{

delete[] array[i][j];

}

delete[] array[i];

}

delete[] array;

system("pause");