**Task 1:**

#include<iostream>

#include<cmath>

using namespace std;

void swap(int\* a, int size)

{

int r = size; // this will do what? for printing my array i need array size equal to what user enters

a = new int[size]; // used DMA for array

int temp; // temp will be used in variable for swaping

int c = 0; // counter variable

if (size % 2 == 0) // this is for centeral value

{

cout << "the array size cannot be 0 ! " << endl;

c++;

}

while (c == 0)

{

a[size]; // array will be equal to said size

cout << "enter array elements = " << endl;

for (int i = 0;i < size;i++) // cin the values of array !

{

cin >> a[i];

}

int o = 1; // for managing the right index value

int num; // num will store the centeral value

num = round(size / 2); // using round function to round off to middle value

for (int i = 0;i < size;i++) // for main swaping

{

if (i == num) // if i = middle value than letb it stay as it is

{

a[i] = a[num];

}

else // swap the respective indexes

{

size = size - 1;

temp = a[i];

a[i] = a[size];

a[size] = temp;

o++;

}

}

cout << "the array after swaping around middle value is = " << endl;

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

{

cout << a[i] << endl; // printing the array

}

c++; // counter which i have used in while loop

}

}

int main()

{

cout << "PLEASE ENTER THE POINTER ARRAY SIZE= "<<endl;

int x;

cin >> x;

int \* a ;

a = new int [100]; // DMA array used

swap(a, x); // function called

system("pause");

return 0;

}



**Task 2**

1. The variable var firstly has the value 2. Then we called function change and have passed parameter which holds value 5. In the function z becomes 15. We add 15 to value 5 and return its value which is **20**. As function was called in var so now var becomes equal to **20**
2. After this line is executed, the function triple will be called into result and after it function double will be called , eventually **20** value would return .
3. After the execution of line 15 , the value that would be stored in var would be **8000.**
4. After execution of line 16 , the value that would be stored in var would be **8000**.
5. Once the function **change** has been called the value of z would ultimately become **15**.

**Task 3:**

#include<iostream>

#include<fstream>

using namespace std;

// as the question does not specifies what should be the function

// name so i have set the name of function as per my own name

int\* arham(int \*a,int size)

{

int\* b; // new array has been created that is twice the size of previous array

size = size \* 2;

b = new int[size ];

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

{

b[i] = a[i];

}

for (int i = (size/2);i < size;i++) // giving unused elements of array value 0

{

b[i] = 0;

}

for (int i = 0;i < size;i++) // printing the array after copying ;

{

cout << b[i] << endl;

}

return b; // return type wass a function so we have to return a pointer as well

}

int main()

{

ifstream read;

read.open("integers.txt");

if (read.fail())

{

cout << "the file was unable to open " << endl;

}

else

{

cout << "the file opened sucessfully " << endl;

int no\_of\_ints;

int counter = 0;

while (!read.eof())

{

read >> no\_of\_ints;

counter++;

}

read.close();

// now i will be using DMA for array

int\* a;

a = new int[counter]; // dynamic array will be equal to the number of integers

read.open("integers.txt");

if (read.fail()) // for checking the file status

{

cout << "the file was unable to open " << endl;

}

else

{

cout << "the file opened sucessfully " << endl;

int index = 0;

while (!read.eof())

{

read >> a[index]; // reading the values into dynamic array

index++; // for increment in index number

}

read.close();

cout << "the array after reading from the file is equal to = " << endl;

//for (int i = 0;i < counter-1;i++) // printing the array

//{

// cout << \*(a + i) << endl;

//}

}

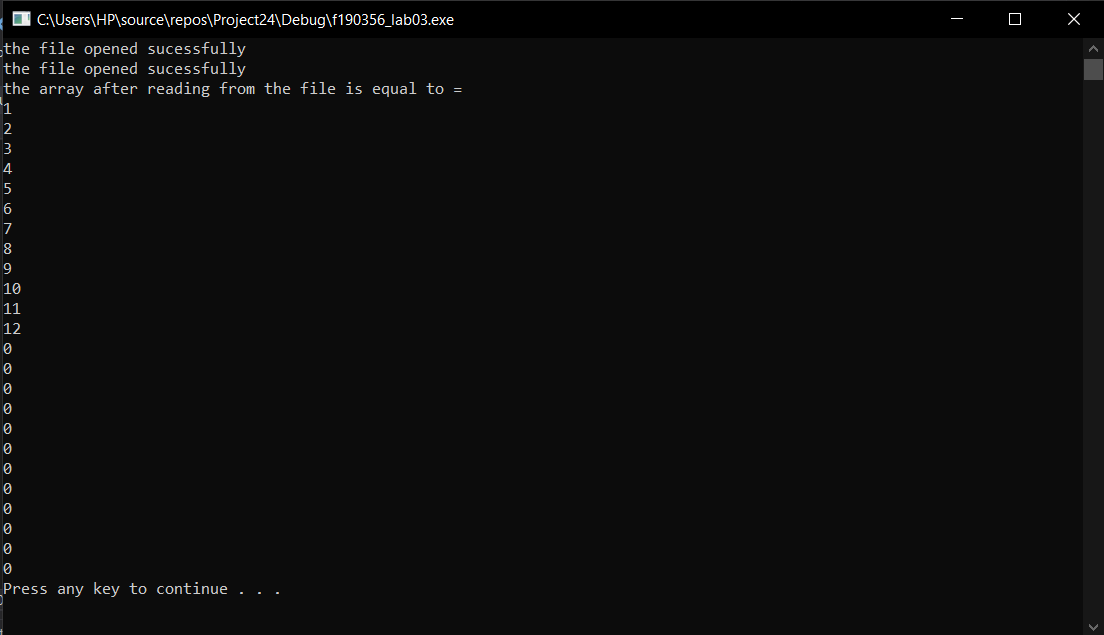
arham(a, no\_of\_ints);

system("pause");

return 0;

}

}



**Task 4:**

#include<iostream>

#include<fstream>

using namespace std;

void input(int \*\*p,int row,int col) // function for inputing the array

{

cout << "enter matrix value please = " << endl;

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

{

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

{

cin >> p[i][j];

}

}

}

void display(int \*\*p,int row,int col ) // function for displaying the matrix on the screen

{

cout << "printing the entered matrix = " << endl;

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

{

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

{

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

}

cout << endl;

}

}

void input2(int\*\* p, int row2, int col2) // function for inputing the seconf matrix (Q)

{

cout << "enter matrix value please = " << endl;

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

{

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

{

cin >> p[i][j];

}

}

}

int\*\* sum(int\*\* p,int row, int col, int\*\* b, int row1,int col2)

{

int\*\* c;

c = new int\* [row]; // here for my ease i have used third 2d DMA array for storing the sum

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

{

c[i] = new int[col];

}

int counter = 0; // if row and col number do not match

if (row != row1 || col != col2)

{ // will terminate the program if row and col no do not match xD

cout << "THE FUNCTION CANNOT WORK FURTHER HENCE SUM WILL NOT BE CARRIED OUT AND FUNC WILL TERMINATE NOW" << endl;

counter++;

while (counter != 0)

{

break;

}

}

else

{ // carried out addition

cout << "ADDING THE BOTH MATRICES = " << endl;

{

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

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

{

c[i][j] = p[i][j] + b[i][j];

}

}

// used display function as said in the requiremnets of the program

display(c, row, col);

}

return c;

}

int main(){

int\*\* a; // for p matrix

int col=0; // rows for p

int row=0; // cols for p

int row1 = 0; // rows for q

int col2 = 0; // col for q

cout << "enter row number of matrix please = " << endl;

cin >> row;

cout << "enter column number of matrix please = " << endl;

cin >> col;

a = new int\* [row];

{

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

{

a[i] = new int[col];

}

}

input(a, row, col); // function for p matrix only

display(a, row, col); // dunction for p matrix only right now

int\*\* b; // for b matrix

cout << "enter row number of matrix please = " << endl;

cin >> row1;

cout << "enter column number of matrix please = " << endl;

cin >> col2;

b = new int\* [row1];

{

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

{

b[i] = new int[col2];

}

}

input2(b, row1, col2); // function for q matrix only

display(b, row1, col2); // function for q matrix only

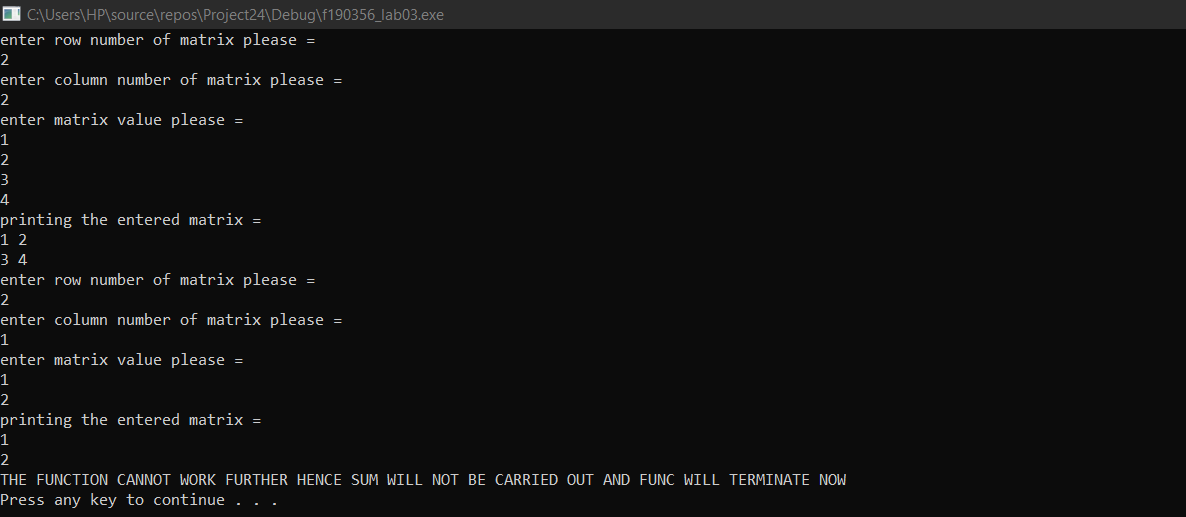
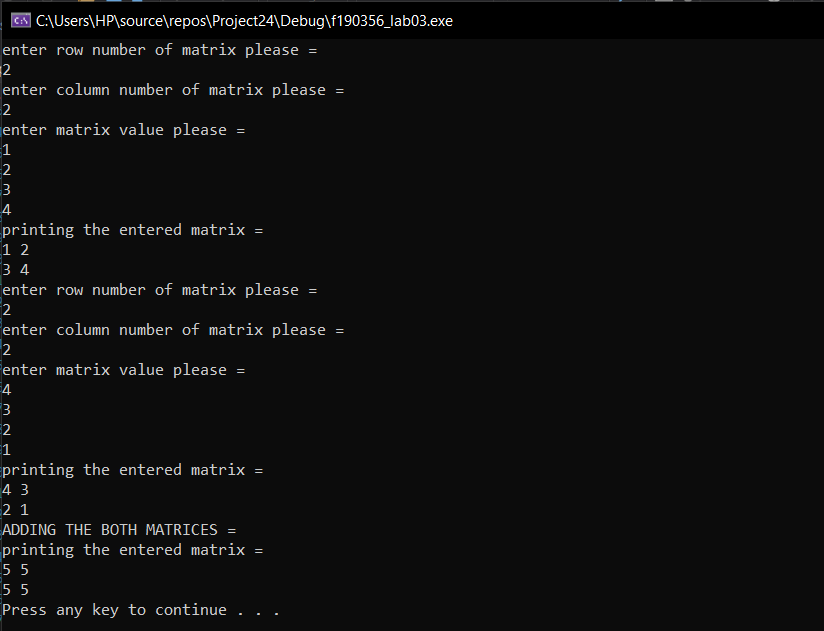
sum(a, row, col, b, row1, col2); // function called for sum of matrices

system("pause");

return 0;

// the end xD

}



**Task 5:**

#include<iostream>

#include<fstream>

using namespace std;

bool check(int n)

{

//function for checking if matrix is sqaure or not

int a = 0;

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

{

a = i \* i;

if (a == n)

return 1;

}

return 0;

}

void swapit(int\* matrix, int s, int r)

{

// sawping the rows of matrix

int temp;

cout << "matrix after swapping rows = " << endl;

for (int i = 0; i < r / 2; i++)

{

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

{

temp = matrix[r \* i + j];

matrix[r \* i + j] = matrix[r \* (r - 1 - i) + j];

matrix[r \* (r - 1 - i) + j] = temp;

}

}

}

void InputMatrix(int\* matrix, int size)

{

// simply would intake matrix values

cout << "enter your matrix elements= " << endl;

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

{

cin >> matrix[i];

}

}

void Display(int\* matrix, int size, int r)

{

// i have written this function so user can demonstrate before and after swapping

cout << "the matrix entered is as follows = " << endl;

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

{

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

{

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

}

cout << endl;

}

}

void DisplayMatrix(int\* matrix, int size, int r)

{

// this loop will print the matrix in square form

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

{

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

{

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

}

cout << endl;

}

}

int main()

{

int size; // it would be user defined size of array

int r;

cout << "enter square size of array = ";

cin >> size;

int\* matrix; // one dynamic array

matrix = new int[size];

while (check(size) != 1) // i wrote this function so i can indentify the matrix as a square matrix

{

cout << "the matrix is not a square matrix! " << endl;

cout<<"Enter square size of array= "<<endl;

cin >> size;

}

r = sqrt(size); // foe getting the row number and col number equally

InputMatrix(matrix, size);

Display(matrix, size, r); // called functions as per instruction of mannual

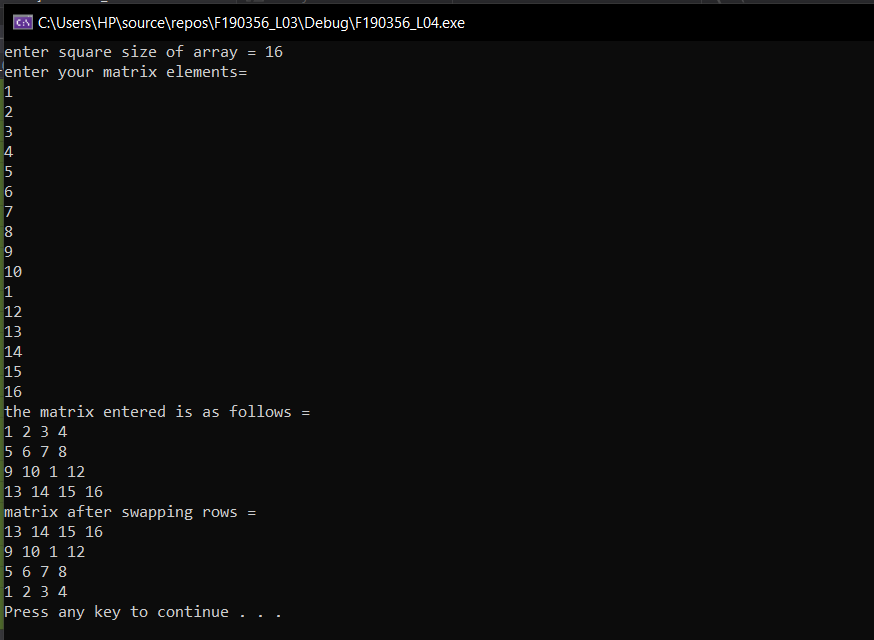
swapit(matrix, size, r); // called functions as per instruction of mannual

DisplayMatrix(matrix, size, r); // called functions as per instruction of mannual

system("pause");

return 0;

}



**Task 6:**

#include<iostream>

#include<fstream>

using namespace std;

void inputMatrix(int\*\* matrix, int row, int col)

{

// simply intaking the values of matrix

cout << " enter llements of row " << row + 1 << " = ";

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

{

cin >> matrix[row][i];

}

}

void displayMatrix(int\*\* matrix, int row, int col, int\* a)

{

// displaying the matrix simply

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

{

for (int j = 0; j < a[i]; j++)

{

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

}

cout << endl;

}

}

int main()

{

int row, col;

cout << "enter the number of rows = ";

cin >> row;

int\*\* matrix;

matrix= new int\* [row]; //used DMA for array as said in mannual

int\* a= new int[row]; // why i did this ? because now this array will keep a record of entered columns respect

// to every row number

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

{

cout << "enter the number elements that you want to enter in row " << i + 1 << " = "; // will tell user how much elements he needs in a row

cin >> col;

matrix[i] = new int[col]; // doing what i explained in line no 38 and 39 xD

a[i] = col;

inputMatrix(matrix, i, col);

}

cout << "the matrix is equal to = " << endl;

displayMatrix(matrix, row, col, a);

system("pause");

return 0;

}

