Name : ***Harsh Tiwari***

Subject Code : CS 291

Assignment : H

Roll : 10800119098

**//H.1**

#include <stdio.h>

#include <stdlib.h>

void display( float a, float b, char ch, float r); void add (float a, float b); void subtract (float a, float b);

void multiply (float a, flaot b);

void divide ( float a, float b);

int main()

{

float a, b;

int ch;

do{

printf("Enter two numbers : ");

scanf("%f %f", &a, &b);

printf("\n1.Addition");

printf("\n2.Subtraction");

printf("\n3.Multiplication");

printf("\n4.Division");

printf("\n5.Exit");

printf("\nEnter your choice: ");

scanf("%d", &ch);

switch(ch){

case 1: add(a, b);

break;

case 2: subtract(a, b);

break;

case 3: multiply(a, b);

break;

case 4: divide(a, b);

break;

case 5: exit(0);

default: printf("\nInvalid input");

}

} while(1); return 0;

}

void add (float a, float b){

float r = a+b;

display (a, b, '+', r);

}

void subtract (float a, float b){

float r = a-b;

display (a, b, '-', r);

}

void multiply (float a, float b){

float r = a\*b;

display (a, b, '\*', r);

}

void divide (float a, float b){

float r = a/b;

display (a, b, '-', r);

}

void display( float a, float b, char ch, float r){ printf("%.2f %c %.2f = %.2f\n", a, ch, b, r);

}

**//H.2**

#include <stdio.h>

#include <stdlib.h>

int factors (int n);

int primefactors (int n);

int factorial (int n);

int isPrime (int n);

int isFibonacci (int n);

int count (int n);

int isArmstrong (int n);

int isPerfect (int n);

int main()

{

int i, n, ch;

printf("Enter any integer: ");

scanf("%d", &n);

printf ("\n1. Factors of the number"); printf ("\n2. Prime factors of the number"); printf ("\n3. Factorial of the number"); printf ("\n4. Number is prime or not ?");

printf ("\n5. Number is in Fibonacci series or not ?"); printf ("\n6. Count the number of digits");

printf ("\n7. Number is Armstrong or not ?"); printf ("\n8. Number is perfect or not");

printf("Enter your choice");

scanf("%d", &ch);

do{

switch(ch){

case 1: factors (int n);

break;

case 2: primeFactors (int n);

break;

case 3: factorial (int n);

break;

case 4: isPrime (int n);

break;

case 5: isFibonacci (int n);

break;

case 6: count (int n);

break;

case 7: isArmstrong (int n);

break;

case 8: isPerfect (int n);

break;

case 9: exit(0);

default: printf("\nInvalid Input !!");

}

} while (1); return ;

}

int factors (int n) {

int i;

printf("Factors of %d are :", n);

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

if(n % i == 0);

{

printf("%d\t", i);

}

}

}

int factorial (int n) {

int i, fact = 1, n;

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

fact = fact\*i;

}

printf("\nFactorial of the given number is : %d", fact); return fact;

}

int isPrime (int n) {

int i, ifPrime;

for( i=2; i<=n-1; i++)

{

if(n % i ==0)

{ ifPrime = 0; } else { ifPrime = 1 ;}

}

return (ifPrime);

}

int count (int n) {

int cnt = 0;

if(n> 0){

cnt++;

cnt(n/10);

}

else { return count;}

}

int isPerfect (int n) {

int s = sqrt(n);

if(s\*s == n)

return 1;

else

return 0;

}

int primeFactors (int n) {

int n;

factors( isPrime( n));

}

**//H.3**

#include <stdio.h>

#include <stdlib.h>

int sort(int a[], int n)

{

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

{

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

{

if([j] > a[j+1])

{

int temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

}

int isEqual (int a1[], int a2[], int n, int m)

{

sort (a1,n);

sort (a2,m);

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

{

if(a1[i] != a2[i])

{

return 0;

}

}

}

int main()

{

int n, m;

printf ("Enter the size of the First and the Second matrix, respectively:

\n";);

scanf("%d %d", &n, &m);

int a1[n], a2[m], i;

printf("Enter the elements of first array: ");

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

{

scanf("%d",&a1[i]);

}

printf("Enter the elements of second array: ");

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

{

scanf("%d",&a2[i]);

}

if( isEqual( a1, a2,n,m) == 0)

{

printf("Arrays are NOT eqaul.");

}

else("Arrays are same.");

return 0;

}

**//H.4**

#include <stdio.h>

#include <stdlib.h>

void makeDaigonalZero(int mat[5][5]);

int main()

{

int mat[5][5],i,j, n;

printf("Enter the order N of the matrix : "); // taking order of N\*N matrix scanf("%d", &n);

printf("Enter the elements of a matrix of order %d : \n", n);

for(i=0;i<n;i++) // scanning for elements of matrix { for(j=0;j<n;j++)

{

scanf("%d",&mat[i][j]);

}

}

printf("\n\n The original matrix is: \n");

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

{

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

{

printf("%d\t",mat[i][j]);

}

printf("\n");

}

makeDaigonalZero(mat);

printf("The matrix after making diagonals elements equal to 0 : \n")

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

{

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

{

printf("%d\t",mat[i][j]);

}

printf("\n");

}

return 0;

}

void makeDaigonalZero()

{

int i, j, n;

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

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

{

if(i==j || (i+j+1) == n) mat[i][j] = '0';

}

printf("\n");

}

}

**//H.5**

#include <stdio.h>

#include <stdlib.h>

void doSort (int a[], int n);

int main()

{

int [10], n, i, e, j;

printf("Enter the value of N\n");

scanf("%d", &n);

printf("Enter the numbers: ");

scanf("%d", &a[i]);

doSort(a);

printf("The number arranged in ascending order are given below: \n");

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

{

printf("%d\n", a[i]);

}

return 0;

}

void doSort (int a[], int n)

{

int i, j;

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

{

for(j+i; j < n; ++j)

{

if([j] > a[j+1])

{

int e = a[j];

a[j] = a[j+1];

a[j+1] = e;

}

}

}

}