**DSA ASSIGNMENT-1**

**TANISHA KARMAKAR**

**21051950**

**CSE 37**

**Q1. WAP to test whether a number is prime or not.**

**Ans.**

#include <stdio.h>

int main() {

int n, i, flag = 0;

printf("Enter a positive integer: ");

scanf("%d", &n);

if (n == 0 && n == 1)

flag = 1;

for (i = 2; i <= n / 2; ++i) {

if (n % i == 0) {

flag = 1;

break;

}

}

if (flag == 0)

printf("%d is a prime number.", n);

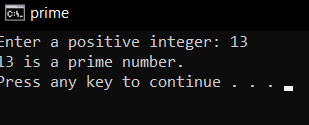
else

printf("%d is not a prime number.", n);

return 0;

}

**Output:**



**Q2. WAP to calculate x^y using functions.**

#include <stdio.h>

#include<math.h>

int power(int a,int b)

{

int ans;

ans=pow(a,b);

return ans;

printf(" %d^%d : %d", a, b, ans);

}

int main()

{

int x,y,a;

printf("Enter the value of x:");

scanf("%d",&x);

printf("Enter the value of y:");

scanf("%d",&y);

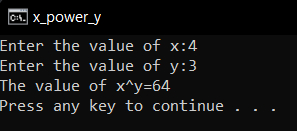
a=power(x,y);

printf("The value of x^y=%d",a);

return 0;

}

**Output:**



**Q3. WAP to find LCM of two numbers using functions**

#include <stdio.h>

int LCM(int x, int y, int max);

int main() {

int x, y, max;

printf ("Enter two positive integers: ");

scanf ("%d %d", &x, &y);

max = (x > y) ? x : y;

int ans = LCM(x, y, max);

printf ("\nThe LCM of %d and %d = %d. ", x, y, ans);

return 0;

}

int LCM ( int x, int y, int max) {

while (1) {

if (max % x == 0 && max % y == 0) {

return max;

break;

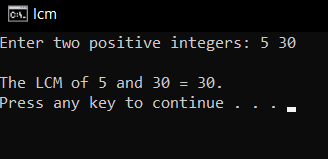
} else {

++max;

}

}

**Output:**



**Q4. Calculator program**

#include<stdio.h>

float add(float n1, float n2);

float sub(float n1, float n2);

float mult(float n1, float n2);

float div(float n1, float n2);

float div0(float n1, float n2);

int main()

{

float num1, num2;

printf("Enter two numbers: ");

scanf("%2f %2f", &num1, &num2);

printf("%2f + %2f = %2f\n", num1, num2, add(num1, num2));

printf("%2f - %2f = %2f\n", num1, num2, sub(num1, num2));

printf("%2f \* %2f = %2f\n", num1, num2, mult(num1, num2));

printf("%2f / %2f = %2f\n", num1, num2, div(num1, num2));

return 0;

}

float add(float n1, float n2)

{

float ans;

ans = n1 + n2;

return ans;

}

float sub(float n1, float n2)

{

float ans;

ans = n1 - n2;

return ans;

}

float mult(float n1, float n2)

{

float ans;

ans = n1 \* n2;

return ans;

}

float div(float n1, float n2)

{

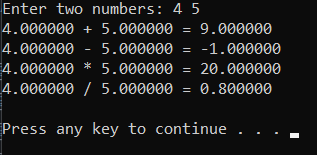
float ans;

ans = n1 / n2;

return ans;

}

**Output:**



**Q5. WAP to create, initialize, assign and access a pointer variable.**

#include <stdio.h>

int main()

{

char ch;

char \*pCh;

pCh = &ch;

ch = 'T';

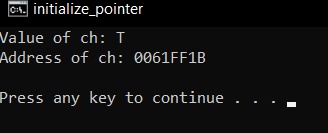
printf("Value of ch: %c\n",ch);

printf("Address of ch: %p\n",&ch);

return 0;

}

**Output:**



**Q6. WAP to add two numbers using pointers**

#include <stdio.h>

int main()

{

int n1, n2, \*p, \*q, sum;

printf("Enter 2 integers to add\n");

scanf("%d%d", &n1, &n2);

p = &n1;

q = &n2;

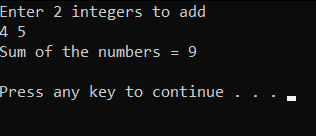
sum = \*p + \*q;

printf("Sum of the numbers = %d\n", sum);

return 0;

}

**Output:**



**Q7. Swap two numbers using call by reference.**

#include <stdio.h>

void swap(int\*, int\*);

int main()

{

int a, b;

printf("Enter a and b\n");

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

printf("Before Swapping a = %d b = %d\n", a, b);

swap(&a, &b);

printf("After Swapping a = %d b = %d", a, b);

return 0;

}

void swap(int \*c, int \*d)

{

int temp;

temp = \*d;

\*d = \*c;

\*c = temp;

}

**Output:**

