**Harshdeep Singh**

**2110990568**

C PROGRAMS --->

1.

#include <stdio.h>

int main() {

float base, height, side, radius, length, breadth, height\_p, area\_tri, area\_sqr, area\_cir, area\_rec, area\_par;

**Area of a triangle**

printf("Enter the base and height of the triangle: ");

scanf("%f%f", &base, &height);

area\_tri = 0.5 \* base \* height;

printf("Area of the triangle is: %.2f\n", area\_tri);

**Area of a square**

printf("Enter the side of the square: ");

scanf("%f", &side);

area\_sqr = side \* side;

printf("Area of the square is: %.2f\n", area\_sqr);

**Area of a circle**

printf("Enter the radius of the circle: ");

scanf("%f", &radius);

area\_cir = 3.14 \* radius \* radius;

printf("Area of the circle is: %.2f\n", area\_cir);

**Area of a rectangle**

printf("Enter the length and breadth of the rectangle: ");

scanf("%f%f", &length, &breadth);

area\_rec = length \* breadth;

printf("Area of the rectangle is: %.2f\n", area\_rec);

**Area of a parallelogram**

printf("Enter the base and height of the parallelogram: ");

scanf("%f%f", &base, &height\_p);

area\_par = base \* height\_p;

printf("Area of the parallelogram is: %.2f\n", area\_par);

return 0;

}

2.

#include <stdio.h>

#include <math.h>

int main() {

float x, radian, sine;

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

scanf("%f", &x);

radian = x \* (M\_PI / 180); // Convert degrees to radians

sine = sin(radian); // Calculate the sine value

printf("sin(%.2f) = %.4f\n", x, sine); // Print the sine value with 4 decimal places

return 0;

}

3.

#include <stdio.h>

#include <math.h>

int main() {

double angle, cosine;

printf("Enter an angle in radians: ");

scanf("%lf", &angle);

cosine = cos(angle);

printf("The cosine of %.2lf radians is %.2lf.\n", angle, cosine);

return 0;

}

4.

#include <stdio.h>

int main() {

double x, y;

// Input the coordinates of the point

printf("Enter the x-coordinate: ");

scanf("%lf", &x);

printf("Enter the y-coordinate: ");

scanf("%lf", &y);

// Determine the quadrant

if (x > 0 && y > 0) {

printf("The point (%.2lf, %.2lf) is in the first quadrant.\n", x, y);

}

else if (x < 0 && y > 0) {

printf("The point (%.2lf, %.2lf) is in the second quadrant.\n", x, y);

}

else if (x < 0 && y < 0) {

printf("The point (%.2lf, %.2lf) is in the third quadrant.\n", x, y);

}

else if (x > 0 && y < 0) {

printf("The point (%.2lf, %.2lf) is in the fourth quadrant.\n", x, y);

}

else if (x == 0 && y == 0) {

printf("The point (%.2lf, %.2lf) is at the origin.\n", x, y);

}

else if (x == 0) {

printf("The point (%.2lf, %.2lf) is on the y-axis.\n", x, y);

}

else if (y == 0) {

printf("The point (%.2lf, %.2lf) is on the x-axis.\n", x, y);

}

return 0;

}

5.

#include <stdio.h>

#include <math.h>

int main() {

double a, b, c, discriminant, root1, root2, realPart, imaginaryPart;

// Input the coefficients a, b, and c of the quadratic equation

printf("Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):\n");

printf("a = ");

scanf("%lf", &a);

printf("b = ");

scanf("%lf", &b);

printf("c = ");

scanf("%lf", &c);

// Calculate the discriminant

discriminant = b\*b - 4\*a\*c;

// Determine the roots based on the value of the discriminant

if (discriminant > 0) {

root1 = (-b + sqrt(discriminant)) / (2\*a);

root2 = (-b - sqrt(discriminant)) / (2\*a);

printf("The roots are real and distinct: %.2lf and %.2lf\n", root1, root2);

}

else if (discriminant == 0) {

root1 = -b / (2\*a);

printf("The roots are real and equal: %.2lf\n", root1);

}

else {

realPart = -b / (2\*a);

imaginaryPart = sqrt(-discriminant) / (2\*a);

printf("The roots are complex and conjugate: %.2lf + %.2lfi and %.2lf - %.2lfi\n", realPart, imaginaryPart, realPart, imaginaryPart);

}

return 0;

}

6.

#include <stdio.h>

int main() {

char operator;

double operand1, operand2, result;

// Input the operator and operands

printf("Enter an operator (+, -, \*, /): ");

scanf("%c", &operator);

printf("Enter two operands: ");

scanf("%lf %lf", &operand1, &operand2);

// Perform the arithmetic operation based on the operator

switch (operator) {

case '+':

result = operand1 + operand2;

printf("%.2lf + %.2lf = %.2lf", operand1, operand2, result);

break;

case '-':

result = operand1 - operand2;

printf("%.2lf - %.2lf = %.2lf", operand1, operand2, result);

break;

case '\*':

result = operand1 \* operand2;

printf("%.2lf \* %.2lf = %.2lf", operand1, operand2, result);

break;

case '/':

if (operand2 == 0) {

printf("Error: division by zero");

}

else {

result = operand1 / operand2;

printf("%.2lf / %.2lf = %.2lf", operand1, operand2, result);

}

break;

default:

printf("Error: invalid operator");

}

return 0;

}

7.

#include <stdio.h>

int main() {

double principle, rate, time, simple\_interest;

// Input the principle amount, rate of interest, and time period

printf("Enter the principle amount: ");

scanf("%lf", &principle);

printf("Enter the rate of interest: ");

scanf("%lf", &rate);

printf("Enter the time period (in years): ");

scanf("%lf", &time);

// Calculate the simple interest

simple\_interest = (principle \* rate \* time) / 100;

// Print the simple interest

printf("The simple interest for the principle amount %.2lf at a rate of %.2lf%% for %.2lf years is: %.2lf", principle, rate, time, simple\_interest);

return 0;

}

8.

#include <stdio.h>

int main() {

double principle, rate, time, simple\_interest;

// Input the principle amount, rate of interest, and time period

printf("Enter the principle amount: ");

scanf("%lf", &principle);

printf("Enter the rate of interest: ");

scanf("%lf", &rate);

printf("Enter the time period (in years): ");

scanf("%lf", &time);

// Calculate the simple interest

simple\_interest = (principle \* rate \* time) / 100;

// Print the simple interest

printf("The simple interest for the principle amount %.2lf at a rate of %.2lf%% for %.2lf years is: %.2lf", principle, rate, time, simple\_interest);

return 0;

}

9.

#include <stdio.h>

#include <math.h>

int main() {

double principle, rate, time, compound\_interest;

int n;

// Input the principle amount, rate of interest, time period, and number of times interest is compounded

printf("Enter the principle amount: ");

scanf("%lf", &principle);

printf("Enter the rate of interest: ");

scanf("%lf", &rate);

printf("Enter the time period (in years): ");

scanf("%lf", &time);

printf("Enter the number of times the interest is compounded per year: ");

scanf("%d", &n);

// Calculate the compound interest

compound\_interest = principle \* pow((1 + (rate / (n \* 100))), (n \* time)) - principle;

// Print the compound interest

printf("The compound interest for the principle amount %.2lf at a rate of %.2lf%% for %.2lf years compounded %d times per year is: %.2lf", principle, rate, time, n, compound\_interest);

return 0;

}

10.

#include <stdio.h>

#include <math.h>

int main() {

int degree, i;

float x, coefficient, result = 0;

printf("Enter the degree of the polynomial: ");

scanf("%d", &degree);

for(i = degree; i >= 0; i--) {

printf("Enter the coefficient of x^%d: ", i);

scanf("%f", &coefficient);

result += coefficient \* pow(x, i);

}

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

scanf("%f", &x);

printf("The value of the polynomial equation for x = %f is %f\n", x, result);

return 0;

}

11.

#include <stdio.h>

int factorial(int n);

int main() {

int n, r;

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

scanf("%d", &n);

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

scanf("%d", &r);

int nPr = factorial(n) / factorial(n - r);

printf("%dP%d = %d\n", n, r, nPr);

return 0;

}

int factorial(int n) {

if(n == 0) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

12.

int factorial(int n);

int main() {

int n, r;

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

scanf("%d", &n);

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

scanf("%d", &r);

int nCr = factorial(n) / (factorial(r) \* factorial(n - r));

printf("%dC%d = %d\n", n, r, nCr);

return 0;

}

int factorial(int n) {

if(n == 0) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

13.

#include <stdio.h>

int findGCD(int a, int b);

int findMultiplicativeInverse(int a, int m);

int main() {

int num, mod;

printf("Enter a number: ");

scanf("%d", &num);

printf("Enter a modulus: ");

scanf("%d", &mod);

int gcd = findGCD(num, mod);

if(gcd != 1) {

printf("Multiplicative inverse does not exist\n");

} else {

int inverse = findMultiplicativeInverse(num, mod);

printf("The multiplicative inverse of %d modulo %d is %d\n", num, mod, inverse);

}

return 0;

}

int findGCD(int a, int b) {

if(b == 0) {

return a;

} else {

return findGCD(b, a % b);

}

}

int findMultiplicativeInverse(int a, int m) {

int m0 = m;

int y = 0, x = 1;

if(m == 1) {

return 0;

}

while(a > 1) {

int q = a / m;

int t = m;

m = a % m;

a = t;

t = y;

y = x - q \* y;

x = t;

}

if(x < 0) {

x += m0;

}

return x;

}

14.

#include <stdio.h>

int isHappy(int num);

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if(isHappy(num)) {

printf("%d is a happy number\n", num);

} else {

printf("%d is not a happy number\n", num);

}

return 0;

}

int isHappy(int num) {

int rem, sum = 0;

while(num != 1 && num != 4) { // If num is 1 or 4, then it's a happy number

sum = 0;

while(num > 0) {

rem = num % 10;

sum += rem \* rem;

num /= 10;

}

num = sum;

}

return (num == 1); // If num is 1, then it's a happy number

}

15.

#include <stdio.h>

int isHappy(int num);

int main() {

int n, i;

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

scanf("%d", &n);

printf("Happy numbers till %d:\n", n);

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

if(isHappy(i)) {

printf("%d ", i);

}

}

printf("\n");

return 0;

}

int isHappy(int num) {

int rem, sum = 0;

while(num != 1 && num != 4) { // If num is 1 or 4, then it's a happy number

sum = 0;

while(num > 0) {

rem = num % 10;

sum += rem \* rem;

num /= 10;

}

num = sum;

}

return (num == 1); // If num is 1, then it's a happy number

}