**PROBLEM STATEMENT 1**: - Write a C program to calculate the factorial of a given number using recursion.

**OBJECTIVE** : - To understand the implementation of recursion method in a program.

**PROGRAM**

#include <stdio.h>

int fact(int);

int main()

{

int n, f;

printf("Enter number : ");

scanf("%d", &n);

f = fact(n);

printf("Factorial = %d", f);

return 0;

}

int fact(int n)

{

if (n==0)

{

return 0;

}

else if(n == 1)

{

return 1;

}

else

{

return n\*fact(n-1);

}

}

**OUTPUT**

Enter number : 5

Factorial = 120

**PROBLEM STATEMENT 2**: - Write a C program to calculate power a given number using recursion.

**OBJECTIVE** : - To understand the implementation of recursion method in a program.

**PROGRAM**

#include <stdio.h>

int power(int, int);

int main()

{

int base, powi, result;

printf("Enter the base number : ");

scanf("%d",&base);

printf("Enter the power number : ");

scanf("%d",&powi);

result = power(base, powi);

printf("%d^%d = %d", base,powi,result);

return 0;

}

int power(int base, int powi)

{

if(powi!=0)

{

return base\*power(base,powi-1);

}

else

{

return 1;

}

}

**OUTPUT**

Enter the base number : 5

Enter the power number : 3

5^3 = 125

**PROBLEM STATEMENT 3**: - Write a C program to calculate the root of an quadratic equation ax²+bx+c=0 by calculating D.

**OBJECTIVE** : - To understand how to find the roots through programming.

**PROGRAM**

#include <math.h>

#include <stdio.h>

int main()

{

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

printf("Enter coefficients a, b and c: ");

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

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

if (discriminant > 0)

{

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

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

printf("root1 = %f and root2 = %f", root1, root2);

}

else if (discriminant == 0)

{

root1 = root2 = -b / (2 \* a);

printf("root1 = root2 = %f;", root1);

}

else

{

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

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

printf("root1 = %f+%fi and root2 = %f-%fi", realPart, imagPart, realPart, imagPart);

}

return 0;

}

**OUTPUT**

Enter coefficients a, b and c: 52

10

28

root1 = root2 = -0.491046;

**PROBLEM STATEMENT 4**: - Write a menu driven program to design a calculator for-

1. Addition, Subtraction, multiplication and division of two numbers
2. Exponent of a number
3. Logarithm of a number
4. Absolute of a number

**OBJECTIVE** : - To understand the switch case method in the program.

**PROGRAM**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

void asmd();

void expo();

void loga();

void absolute();

int main()

{

char ch;

while (1)

{

printf("Press 1 for asmd\nPress 2 for exponent\nPress 3 for log\nPress 4 for absolute\nPress 5 for exit\n");

scanf("%c", &ch);

switch (ch)

{

case '1':

asmd();

break;

case '2':

expo();

break;

case '3':

loga();

break;

case '4':

absolute();

break;

case '5':

exit(1);

break;

default:

printf("Wrong choice");

break;

}

}

return 0;

}

void asmd()

{

float a, b;

printf("Enter two numbers on which you want to perform asmd : ");

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

float res;

char ch;

while (1)

{

printf("Press 1 for add\nPress 2 for sub\nPress 3 for multi\nPress 4 for dic\nPress 5 for exit\n");

scanf("%c", &ch);

switch (ch)

{

case '1':

res = a+b;

printf("Result is : %f", res);

break;

case '2':

res = a-b;

printf("Result is : %f", res);

break;

case '3':

res = a\*b;

printf("Result is : %f", res);

break;

case '4':

res = a/b;

printf("Result is : %f", res);

break;

case '5' :

exit(1);

break;

default:

printf("Wrong choice");

break;

}

}

}

void expo()

{

float num;

printf("Enter number : ");

scanf("%f", &num);

int powi, res;

printf("Enter power : ");

scanf("%d", &powi);

res = pow(num, powi);

printf("Result is : %f", res);

}

void loga()

{

float num;

printf("Enter number : ");

scanf("%f", &num);

float res;

res = log(num);

printf("Result is : %f", res);

}

void absolute()

{

float num;

printf("Enter number : ");

scanf("%f", &num);

float res;

res = abs(num);

printf("Result is : %f", res);

}

**OUTPUT**

Press 1 for asmd

Press 2 for exponent

Press 3 for log

Press 4 for absolute

Press 5 for exit

1

Enter two numbers on which you want to perform asmd : 9

3

Press 1 for add

Press 2 for sub

Press 3 for multi

Press 4 for dic

Press 5 for exit

Wrong choicePress 1 for add

Press 2 for sub

Press 3 for multi

Press 4 for dic

Press 5 for exit

2

Result is : 6.000000Press 1 for add

Press 2 for sub

Press 3 for multi

Press 4 for dic

Press 5 for exit

Wrong choicePress 1 for add

Press 2 for sub

Press 3 for multi

Press 4 for dic

Press 5 for exit

3

Result is : 27.000000Press 1 for add

Press 2 for sub

Press 3 for multi

Press 4 for dic

Press 5 for exit

Wrong choicePress 1 for add

Press 2 for sub

Press 3 for multi

Press 4 for dic

Press 5 for exit

4

Result is : 3.000000Press 1 for add

Press 2 for sub

Press 3 for multi

Press 4 for dic

Press 5 for exit

Wrong choicePress 1 for add

Press 2 for sub

Press 3 for multi

Press 4 for dic

Press 5 for exit

5

**PROBLEM STATEMENT 5**: - Develop a program to evaluate sin(x) using the series.

sin x=s-x3/3! + x5/5! - ………. Up to ᶯth term

**OBJECTIVE** : - To understand the concept of moving the series through programming.

**PROGRAM**

#include <stdio.h>

#include <math.h>

int fac(int x)

{

int i,fac=1;

for(i=1;i<=x;i++)

fac=fac\*i;

return fac;

}

int main()

{

float x,Q,sum=0;

int i,j,limit;

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

scanf("%f",&x);

printf("Enter the limit upto which you want to expand the series: ");

scanf("%d",&limit);

Q=x;

x = x\*(3.1415/180);

for(i=1,j=1;i<=limit;i++,j=j+2)

{

if(i%2!=0)

{

sum=sum+pow(x,j)/fac(j);

}

else

sum=sum-pow(x,j)/fac(j);

}

printf("Sin(%0.1f): %f",Q,sum);

return 0;

}

**OUTPUT**

Enter the value of x of sinx series: 1

Enter the limit upto which you want to expand the series: 3

Sin(1.0): 0.017452

**PROBLEM STATEMENT 6**: - Write a C program to enter a floating point number and print the integer and fractional part of that number.

**OBJECTIVE**: - To understand the concept of numbers in the programming.

**PROGRAM**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

float num, floating;

int integr;

printf("Enter number");

scanf("%f", &num);

integr = abs(num);

floating = num - integr;

printf("Integer part is %d\nFloating part is %f", integr, floating);

}

**OUTPUT**

Enter number52

Integer part is 52

Floating part is 0.000000

**PROBLEM STATEMENT 7**: - Write a C program to find the range of the given equation.

**OBJECTIVE**: - To understand how to find range in a program.

**PROGRAM**

#include<stdio.h>

#include<math.h>

float fun(float x)

{

return((x\*x\*x)-2\*x-5);

}

void main()

{

int i=0,pos=0;

float ai=0,bi=0;

while(1)

{

if(fun(i)==0)

printf("exact root found at %d",i);

if(fun(i)\*fun(i+1)<0)

{

pos++;

break;

}

i++;

}

if(pos)

{

ai=i;

bi=i+1;

}

else

printf("Root not found");

if(pos)

{

printf("\n\nrange for the function is=%.0f and %.0f\n",ai,bi);

}

}

**OUTPUT**

Range for the function is=2 and 3

**PROBLEM STATEMENT 8**: - Write a C program to find the factor of the given equation using Bisection Method.

**OBJECTIVE**: - To understand the implementation of Bisection Method in a program.

**PROGRAM**

#include<stdio.h>

#include<math.h>

float fun(float x)

{

return((x\*x\*x)-2\*x-5);

}

void main()

{

int i=0,pos=0;

float ai=0,bi=0,xi=0,fxi=0;

printf("\nBISECTION METHOD\n");

while(1)

{

if(fun(i)==0)

printf("exact root found at %d",i);

if(fun(i)\*fun(i+1)<0)

{

pos++;

break;

}

i++;

}

if(pos)

{

ai=i;

bi=i+1;

}

else

printf("Root not found");

if(pos)

{

printf("\n\nrange for the function is=%.0f and %.0f\n",ai,bi);

printf("\nCA LCULATION TABLE\n ");

xi=(ai+bi)/2.0;

fxi=fun(xi);

for(i=0; ;i++)

{

printf("%d\t%f\t%f\t%f\t%f\n ",i+1,ai,bi,xi,fxi);

if((float)fabs(ai-xi)<0.0001f||(float)fabs(bi-xi)<0.0001f)

break;

if(fxi<0)

ai=xi;

else

bi=xi;

xi=(ai+bi)/2.0;

fxi=fun(xi);

}

printf("\n\n");

printf("Root of the function is=%.4f \n",ai);

}

}

**OUTPUT**

BISECTION METHOD

range for the function is=2 and 3

CALCULATION TABLE

1 2.000000 3.000000 2.500000 5.625000

2 2.000000 2.500000 2.250000 1.890625

3 2.000000 2.250000 2.125000 0.345703

4 2.000000 2.125000 2.062500 -0.351318

5 2.062500 2.125000 2.093750 -0.008942

6 2.093750 2.125000 2.109375 0.166836

7 2.093750 2.109375 2.101562 0.078562

8 2.093750 2.101562 2.097656 0.034715

9 2.093750 2.097656 2.095703 0.012862

10 2.093750 2.095703 2.094727 0.001954

11 2.093750 2.094727 2.094238 -0.003495

12 2.094238 2.094727 2.094482 -0.000771

13 2.094482 2.094727 2.094604 0.000592

14 2.094482 2.094604 2.094543 -0.000090

Root of the function is=2.0945

**PROBLEM STATEMENT 9**: - Write a C program to find the factor of the given equation using Regula Falsi Method.

**OBJECTIVE**: - To understand the implementation of Regula Falsi Method in a program.

**PROGRAM**

#include<stdio.h>

#include<math.h>

float fun(float x)

{

return((x\*x\*x)-2\*x-5);

}

void main()

{

int i=0,pos=0;

float x0=0,x1=0,x2=0,fx2=0;

printf("\nREGULA FALSI METHOD\n");

while(1)

{

if(fun(i)==0)

printf("exact root found at %d",i);

if(fun(i)\*fun(i+1)<0)

{

pos++;

break;

}

i++;

}

if(pos)

{

x0=i;

x1=i+1;

}

else

printf("Root not found");

if(pos)

{

printf("\n\nrange for the function is=%.0f and %.0f\n",x0,x1);

printf("\nCALCULATION TABLE\n ");

x2=x1-(fun(x1)\*(x1-x0)/(fun(x1)-fun(x0)));

fx2=fun(x2);

for(i=0; ;i++)

{

printf("\n%d\t%f\t%f\t%f\t%f\n ",i+1,x0,x1,x2,fx2);

if((float)fabs(x2-x0)<0.0001f||(float)fabs(x2-x1)<0.0001f)

break;

if(fx2<0)

x0=x2;

else

x1=x2;

x2=x1-(fun(x1)\*(x1-x0)/(fun(x1)-fun(x0)));

fx2=fun(x2);

}

printf("\n\n");

printf("Root of the function is=%.4f \n",x2);

}

}

**OUTPUT**

REGULA FALSI METHOD

range for the function is=2 and 3

CALCULATION TABLE

1 2.000000 3.000000 2.058824 -0.390799

2 2.058824 3.000000 2.081264 -0.147205

3 2.081264 3.000000 2.089639 -0.054677

4 2.089639 3.000000 2.092740 -0.020203

5 2.092740 3.000000 2.093884 -0.007450

6 2.093884 3.000000 2.094306 -0.002745

7 2.094306 3.000000 2.094461 -0.001010

8 2.094461 3.000000 2.094518 -0.000372

Root of the function is=2.0945

**PROBLEM STATEMENT 10**: - Write a C program to find the factor of the given equation using Bisection Method.

**OBJECTIVE**: - To understand the implementation of Bisection Method in a program.

**PROGRAM**