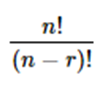
**PROGRAMMING EXERCISES**

1. The formula for permutation with no repetition is given as follows:



where n is the number of things to choose from, and we choose r from n. Develop a C program source code to compute the given formula using recursion. Prompts the user to enter an integer for both n and r, respectively. Make sure that r <= n, n >= 0 and r >=0.

**Danish**

#include<stdio.h>

int fact(int n);

int main()

{

int x, y;

repeat:

printf("Condition available : r <= n, n >= 0 and r >=0\n\n");

printf("The Value of n and r :\n");

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

if((y>x) || (y<0) || (x<0))

{

printf("Math ERROR\n");

goto repeat;

}

else

{

printf("The Number of Possible Permutation When n is %d and r is %d =====> %d", x, y, (fact(x)/fact(x-y)));

}

return 0;

}

5!=5x4x3x2x1

n!=nx(n-1)x(n-2)x..x1

n!=1x2x…xn

int fact(int n)

{

int ans;

if(n==1)

ans=1;

else

ans=n\*fact(n-1);

return ans;

}

ELVIS’ ANSWER

#include <stdio.h>

int fact(int n);

void main()

{

int n,r,npr;

printf("Enter a number n:");

scanf("%d",&n);

printf("Enter a number r:");

scanf("%d",&r);

npr=fact(n)/fact(n-r);

if((r>n) || (n<0) || (r<0))

{

printf("MATH ERROR\n");

}

else

{

printf("Value of %dP%d = %d\n",n,r,npr);

}

}

int fact(int n)

{

int i,f=1;

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

{

f=f\*i;

}

return f;

}

KHAIRI

#include<stdio.h>

int fact(int x);

int main()

{

int x, y, prob, z;

printf("Please ENTER The Value of n and r\n");

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

if((y>x) || (y<0) || (x<0))

{

printf("cant solve \n");

}

else

{

printf("The Number of Possible Permutation When n is %d and r is %d :%d", x, y, (fact(x)/fact(x-y)));

}

return 0;

}

int fact(int x)

{

int ans;

if(x==0)

ans=1;

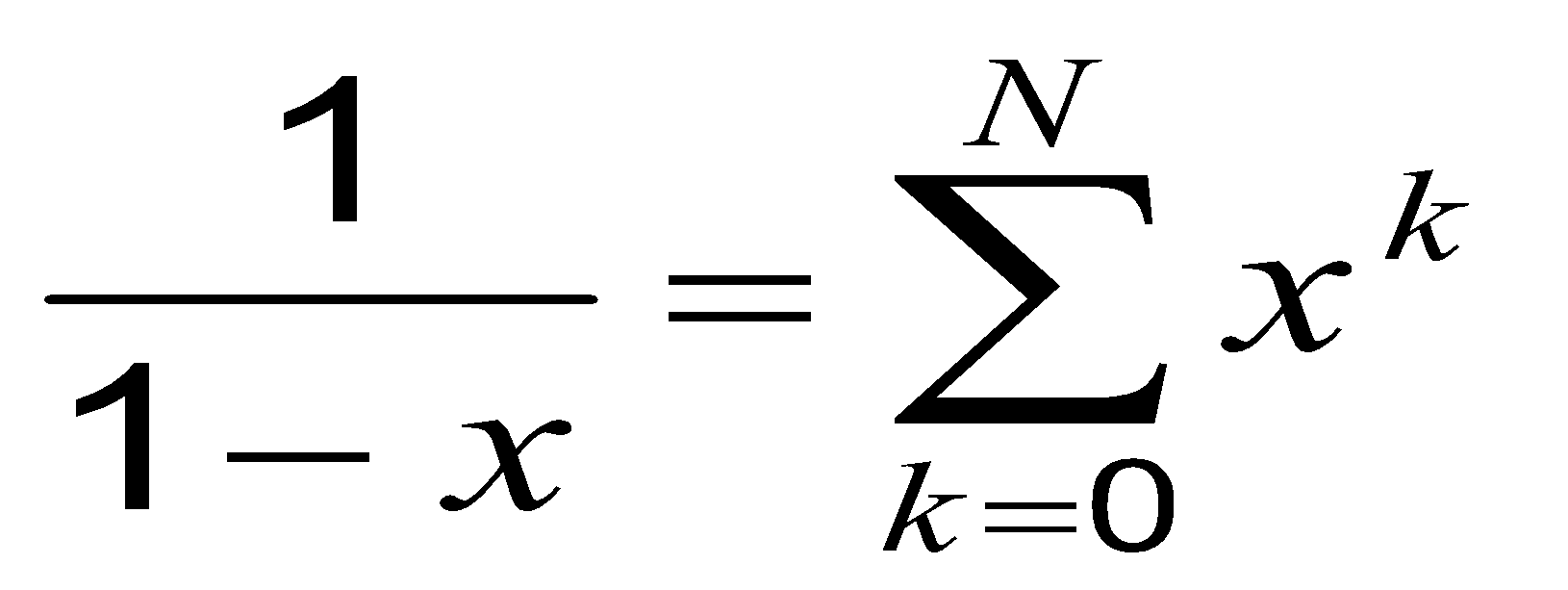
else

ans=x\*fact(x-1);

return ans;

}

1. Construct a C program source code to read N number of integers and compute the answer for the given problem below, where -1< x < 1



Apply the idea of **recursion** in your program.

Khadijah:

#include<stdio.h>

float sum(float a,int b);

int main()

{

int N;

float x;

printf("Enter integer(N) : ");

scanf("%d",&N);

range:

printf("Enter x, -1<x<1 : ");

scanf("%f",&x);

if (x<=-1 || x>=1)

{

printf("Please Try Again\n");

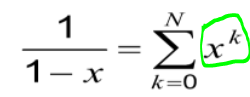
goto range;

}

printf("The sum until %d is %.3f",N,sum(x,N));

return 0;

}



=x^0 + x^1 + x^2 + … + x^n

=1 + x^1 + x^2 + … + x^n

float sum(float a,int b)

{

if (b==0)

return 1;

else

return((pow(a,b))+ (sum(a,b-1)));

}

ANIS

#include<stdio.h>

float sum(float m,int n);

int main()

{

int N;

float x;

printf("Enter number of integers,N : ");

scanf("%d",&N);

table:

printf("Enter the value of x, -1<x<1 : ");

scanf("%f",&x);

if (x<=-1 || x>=1)

{

printf("Please enter another value\n\n");

goto table;

}

printf("The sum until %d is %.3f",N,sum(x,N));

return 0;

}

float sum(float m,int n)

{

if (n==0)

return 1;

else

return((pow(m,n))+ (sum(m,n-1));

}

Ain:

#include<stdio.h>

#include<math.h>

float sum(float a,int b);

int main()

{

int N;

float x;

printf("Enter number of integers,N : ");

scanf("%d",&N);

value:

printf("Enter the value of x, -1<x<1 : ");

scanf("%f",&x);

if (x<=-1 || x>=1)

{

printf("Enter Value Again\n\n");

goto value;

}

printf("sum until %d is %.3f",N,sum(x,N));

return 0;

}

float sum(float a,int b)

{

if (b==0)

return 1;

else

return ((pow(a,b))+ (sum(a,b-1)));

}

3. The following series summation are an approximation for computing the sine and cosine

functions of a floating point

Diagram

Description automatically generated with low confidence

where x must be the angle in radian and n is a positive integer. Design and implement

an interactive program that does the following:

1. Prompts the user to choose sine or cosine function.
2. Prompts the user to enter an integer for n and reads it.
3. Prompts the user to enter a floating-point value for x and reads it.
4. You should construct your program by using **recursion** and work out the recursive functions for sine and cosine(x), respectively.

Ikhzam

#include<stdio.h>

#include<math.h>

float sine(float,float);

float cosine(float,float);

float summation(int,int,float);

float factorial (float);

int main ()

{

float x;

int n,a;

again:

printf("--------------------------------------------------------------------------------\n");

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

printf("This Programme can calculate value of Sine and Cosine\n");

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n");

printf("1: Sine(x):\n2: Cosine(x):\nEnter Here: ");

scanf("%d",&a);

if(a!=1 && a!=2)

goto again;

printf("\nEnter Positive Integer N:");

scanf("%d",&n);

printf("Enter the floating point of angle in radian:");

scanf("%f",&x);

if (a==1)

{

printf("\nsine(%.2f) = %.3f",x,summation(n,a,x));

printf("\n--------------------------------------------------------------------------------");

}

else

{

printf("\nCosine(%.2f) = %.3f",x,summation(n,a,x));

printf("\n--------------------------------------------------------------------------------");

}

}

float summation (int n,int a,float x)

{

float sum = 0;

float k;

if(a==1)

{

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

sum = sum + sine(k,x);

}

else

{

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

sum = sum + cosine(k,x);

}

return sum;

}

float sine(float k, float x)

{

float sin;

sin =(pow(x,(2\*k+1)\*(pow(-1,k))/factorial(2\*k+1)));

return sin;

}

float cosine(float k,float x)

{

float cosin;

cosin = ((pow(x,(2\*k))\*pow(-1,k)/factorial(2\*k)));

return cosin;

}

float factorial(float z)

{

double i,fact=1;

for(i=z;i>=1;i--)

{

fact = i \* fact;

}

return fact;

}

\*\*\*\*\*Comment\*\*\*\*\*\*\*\*

if (a==1)

{

printf("\nsine(%.2f) = %.3f",x,sine(n,x));

printf("\n--------------------------------------------------------------------------------");

}

else

{

printf("\nCosine(%.2f) = %.3f",x,cosine(n,x));

printf("\n--------------------------------------------------------------------------------");

}

}

float sine(float k, float x)

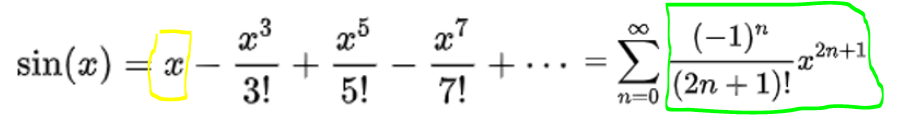
{

float sin;

sin =(pow(x,(2\*k+1)\*(pow(-1,k))/factorial(2\*k+1)));

return sin;

}



float sine(float n, float x)

{

if(n==0)

return 0;

else

return (pow(x,2\*n+1)\*pow(-1,n))/factorial(2\*k+1)+sine(n,x);

}

float cosine(float k,float x)

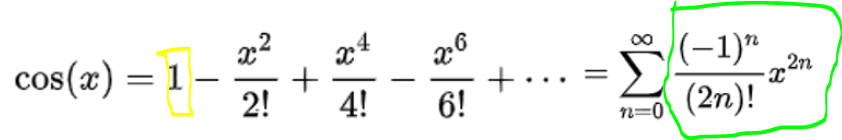
{

float cosin;

cosin = ((pow(x,(2\*k))\*pow(-1,k)/factorial(2\*k)));

return cosin;

}



float cosine(float n, float x)

{

if(n==0)

return 1;

else

return ((pow(x,(2\*k))\*pow(-1,k)/factorial(2\*k)))+cosine(n,x);

}

float factorial(float z)

{

double i,fact=1;

for(i=z;i>=1;i--)

{

fact = i \* fact;

}

return fact;

}

4. Write a program in C to find the greatest common divisor (GCD) of two numbers using ‘ recursion.

Text

Description automatically generated

IKMAL:

#include <stdio.h>

int gcd(int n1, int n2);

int main()

{

int n1, n2;

printf("Enter two positive integers:\n");

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

printf("G.C.D of %d and %d is %d.", n1, n2, gcd(n1, n2));

return 0;

}

gcd( n1, n2)

{

if (n1>n2)

return gcd(n1 - n2, n2);

if (n2>n1)

return gcd(n2 - n1,n1);

else

return n1;

}

HAIQAL :

#include <stdio.h>

int gcd(int n1, int n2);

int main()

{

int n1, n2;

printf("Enter two positive integers:\n");

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

printf("G.C.D of %d and %d is %d.", n1, n2, gcd(n1, n2));

return 0;

}

int gcd(int n1, int n2)

{

if (n1>n2)

return gcd(n1 - n2, n2);

else if (n2>n1)

return gcd(n2 - n1,n1);

else

return n1;

}

Rahmah

#include <stdio.h>

int gcd(int n1, int n2);

int main()

{

int n1, n2;

printf("Enter two positive integers: ");

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

printf("G.C.D between two numbers %d and %d is %d.", n1, n2, gcd(n1, n2));

return 0;

}

int gcd(int n1, int n2)

{

if (n1>n2)

return gcd(n1 - n2, n2);

if (n2>n1)

return gcd(n2 - n1,n1);

else

return n1;

}