1. Absolute Value

int absolute(int I)

{

if(I<0) return -I;

else return I;

}

1. Convert Celsius to Fahrenheit

double cToF(int C)

{

return C\*9.0/5+32;

}

1. Quadratic Equation roots

vector<int> quadraticRoots(int a, int b, int c) {

vector<int> arr;

double D = (b\*b-4\*a\*c);

if(D>=0)

{

D=sqrt(D);

double x1 = (floor((-b+D)/2.0/a));

double x2 = (floor((-b-D)/2.0/a));

if(x1>x2)

arr = {x1,x2};

else

arr = {x2,x1};

}

else

arr.push\_back(-1);

return arr;

}

1. Factorial of number

long long factorial(int N) {

long long f = 1;

for(int i = 1; i<=N; i++) f\*=i;

return f;

}

1. Digits in factorial

int digitsInFactorial(int N)

{

double f = 0;

for(int i = 1; i<=N; i++) f+=log10(i);

return floor(f+1);

}

1. GP term

double termOfGP(int A,int B,int N)

{

double r = 1.0\*B/A;

return A\*pow(r, N-1);

}

1. Primality test

bool isPrime(int N)

{

bool found = true;

for(int i = 2; i\*i<=N;i++)

{

if(N%i==0)

{

found = false;

break;

}

}

return found;

}

1. Exactly 3 divisors

int exactly3Divisors(int N)

{

int count = 0;

for(int j=2;j\*j<N;j++)

{

bool found = true;

for(int i = 2; i\*i<=j;i++)

{

if(j%i==0)

{

found = false;

break;

}

}

if(found) count++;

}

return count;

}

1. Addition under modulo

long long sumUnderModulo(long long a,long long b)

{

long long lim = 1000000007;

long long ans = (a%lim)+(b%lim);

return ans%lim;

}

1. Multiplication under modulo

int multiplicationUnderModulo(long long a,long long b)

{

long long lim = 1000000007;

a = a%lim;

b = b%lim;

long long M =0, p = 1;

while(b)

{

M+=((a\*(b%10)%lim)\*(p%lim))%lim;

M = M%lim;

p\*=10;

b/=10;

}

return M;

}

1. Modular multiplicative inverse

int modInverse(int a, int m)

{

for(int i = 1; i<m;i++)

{

if(a\*i%m==1)

return i;

}

return -1;

}