Mathematics GFG

1. Absolute Value

```
int absolute(int I) {
   if (I<0){
      return -I;
   }
   else{
      return I;
   }
}
```

2. Convert Celsius To Fahrenheit

```
double cToF(int C)
{
return (C* 9/5) + 32 ;
}
```

3. Quadratic Equation Roots

```
void quadraticRoots(int a,int b, int c)
{
  int D = (b*b) - 4*a*c;
  if (D < 0){
     printf("Imaginary");
  }
  else if (D == 0){
     int x = floor(((-b) + sqrt(D))/(2*a));
     printf("%d %d", x, x);
  }
  else if (D > 0){
     int x = floor(((-b) + sqrt(D))/(2*a));
     int y = floor(((-b) - sqrt(D))/(2*a));
     printf("%d %d", x, y);
  }
}
```

4. Factorial Of Number

```
#define ll long long
long long factorial (int N)
{
   ll res = 1;
   for (ll i = N; i > 0; i--){

      res *= i;
   }
   return res;
  }
```

5. Digits In Factorial

```
int digitsInFactorial(int N)
{
if (N < 0)
return 0;
if (N <= 1)
return 1;

double digits = 0;
for (int i=2; i<=N; i++)
digits += log10(i);
return floor(digits) + 1;
}
```

6. GP Term

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

double r=(double)B/(double)A;
return A*pow(r,N-1);
}
```

7. Primality Test

```
bool isPrime(int N)
{
   if (N <= 1)
      return false;

   for (int i = 2; i < N; i++)
      if (N % i == 0)
         return false;

   return true;

}
```

8. Addition Under Modulo

```
int sumUnderModulo(long long a,long long b)
{
   int M=1000000007;
  return (a%M+b%M)%M;
}
```

9. Multiplication Under Modulo

```
int multiplicationUnderModulo(long long a,long long b)
{
   int M=1000000007;
   return ( (a%M)*(b%M)%M);
}
```

10. Modular Multiplicative Inverse

```
int modInverse(int a, int m)
{
int num=-1;
for(int i=1;i<=m;i++){
if((i*a)%m==1){
num=i;
break;
}
}
return num;
}
```