At first you include mathcode.h **(#include "mathcode.h")** then access all property.

The file mathcode.h is save in the same directory of program folder.

1. **Prime number check in O(sqrt(N)) time complexity**.

Pass one integer in **isPrime(n)** it will return number prime or not.

1. **Primality test technique in O(Nlog(logN)) time complexity**

Primality test technique which is used to generate prime numbers in a given range like from 1 to N, where N is not more than 10^7.

Call the function **primality\_test().**

Then All the prime number store in **is\_prime[]** array 1 to 1000000. If array value is 1 the array index is prime number.

You only at first call the function **primality\_test()** then directly access the array **is\_prime[]**.

1. **Prime Factorization in O(sqrt(N)).**

You pass a number through **primeFact(n)** and it return a vector pair.

You need to call the function this way **vector<pair<int,int>>v= primeFact(n).**

Number’s all prime factor is store in a vectore pair. In first element of pair store the **prime facto**r and in second element of pair **store how many times occur the prime factor**.

1. **Binary Exponentiation**((A^N)**) in O(logN) time complexity**

Pass the **base(A)** and **power(N)** through the Function **power(A,N)** it will return (A^N) value in O(logN) time complexity.

1. **Prime Factorization using Sieve in O(log(N))**

This is calculated by **Sieve** algorithm.

You call the function **sieveFun()** it will make an array name **sieve[]** and using this array value we calculate the prime factor of a number of number between 1 to 1000000.

**Example:** We calculate the factor of number 10.

After calling function the array **sieve[]** look like this,

Index= 0, 1 ,2, 3, 4, 5, 6, 7, 8, 9,10

sieve[]={0,-1 ,2, 3, 2, 5, 2, 7, 2, 3, 2}

n=10;

while(n)  
{

cout<<sieve[n];

int z= sieve[n];

n=n/z;

}

1. **Binomial Coefficient using Modulo inverse**

At first you make an array which contain all the factorial from 1 to 10^6. (Because it takes o(n) time so time complexity will increase if we calculate it every time)

Then pass value through the function **C(N,P,array).** It will return the value;