Metropolitan University
Sabbir Ahmed Talukdar
222-115-100

Summery:
1.GCD
2 NCR
3. MOD POWER
4. nth Prime
5. Number Divisor
6. Multiply String and number

bfs

```
//SNIPPET GCD Start-------------1------------

//call gcd(long long a, long long b)

long long gcd(long long a, long long b){
  if(!b) return a;
  return gcd(b,a%b);
}


//SNIPPET GCD End

//SNIPPET NCR START------------2--------------

//need to define MOD, call factorial() just after main function, call nCr(long long n, long long r)

long long fact[MOD];

//bigmod

long long bigmod(long long b, long long p)
{
    long long r = 1;
    while (p)
    {
        if (p & 1) r = (r * b) % MOD; //when the first bit number is 1 (means odd number) then add the power.
        b = (b * b) % MOD;
        p >>= 1;
    }
    return r;
}

void factorial() {
    fact[0] = 1;
    for (long long i = 1; i < MOD; i++) fact[i] = (((i % MOD) * (fact[i - 1] % MOD)) % MOD);
}

//nCr
long long nCr(long long n, long long r) {

    return ((fact[n] % MOD) * (bigmod(((fact[n - r] % MOD) * (fact[r] % MOD)) % MOD, MOD - 2) % MOD)) % MOD;
}
```

Metropolitan University
Sabbir Ahmed Talukdar
222-115-100

```
57.
58. //SNIPPET NCR END
59.
60. //SNIPPET  MOD POWER START----------------------------3---------------------
61.
62. //need to define MOD, call modPow(int b, int p)
63.
64. // Tips: modInverse of b is modPow(int b, int p=MOD-2)
65.
66. int modPow(int b, int p)
67. {
68.     int r = 1;
69.     while(p)
70.     {
71.         if(p&1) r = (r*b)%MOD;
72.
73.         //when the first bit number is 1(means the if the number is odd) then add the power.
74.
75.         b = (b*b)%MOD;
76.         p >>= 1;
77.     }
78.     return r;
79. }
80.
81. //SNIPPET  MOD POWER END
82.
83. //SNIPPET nth prime START--------------4--------------
84.
85. //need to edit nn, run nthPrime(), prime[3] means 3rd prime number
86.
87. #define nn 1000010
88.
89. long long int notprime[nn]={}, prime[nn];
90. void nthPrime(){
91.
92. //nth prime is cout<<prime[n]<<endl;
93.
94.     long long int c=1, i, j;
95.     for(i=3;i*i<=nn;i+=2){
96.         if(!notprime[i]){
97.             for(j=i*i;j<=nn;j+=2*i) notprime[j]=1;
98.         }
99.     }
100.    prime[c++]=2;
101.    for(i=3;i<=nn;i+=2){
102.        if(!notprime[i]){
103.            prime[c++]=i;
104.        }
105.    }
106. }
107.
108. //SNIPPET nth prime END
109.
110. //SNIPPET Number of divisor START-----------------------5-------------------
111.
112. //need to edit nn, call numberofDivisor(long long n)
113.
114. #define nn 1000010
```

Metropolitan University
Sabbir Ahmed Talukdar
222-115-100

```
115.
116. long long int notprime[nn]={}, prime[nn];
117. long long numberofDivisor(long long n){
118.     long long int c=1, i, j, ans=1;
119.     for(i=3;i*i<=nn;i+=2){
120.         if(!notprime[i]){
121.             for(j=i*i;j<=nn;j+=i) notprime[j]=1;
122.         }
123.     }
124.     prime[c++]=2;
125.     for(i=3;i<=nn;i+=2){
126.         if(!notprime[i]){
127.             prime[c++]=i;
128.         }
129.     }
130.
131.         for(i=1;i<=nn && prime[i]*prime[i]<=n;i++){
132.             if(n%prime[i]==0){
133.                 int cnt=1;
134.                 while(n>1 && n%prime[i]==0){
135.                     n/=prime[i];
136.                     cnt++;
137.                 }
138.                 ans*=cnt;
139.             }
140.         }
141.         if(n!=1) ans*=2;
142. }
143.
144. //SNIPPET Number of divisor END
145.
146. //SNIPPET MULTIPLY string and Number Start----------------------6-----------------------
147.
148. //call multyply(string a, int b)
149.
150. string multyply(string a, int b){
151.   int carry = 0, l=a.size();
152.   string ans = "";
153.   for(int i =l-1; i>=0; i--){
154.     carry = ((a[i] - '0') * b + carry);
155.     ans += carry % 10 + '0';
156.     carry /= 10;
157.   }
158.   while(carry != 0){
159.     ans += carry % 10  + '0';
160.     carry /= 10;
161.   }
162.   reverse(ans.begin(), ans.end());
163.   return ans;
164. }
165. //SNIPPET MULTIPLY string and Number END
166.
167. //SNIPPET Boilerplate Start----------------------7----------------------------
168.
169. //بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ
170.
171. #include<bits/stdc++.h>
172. using namespace std;
```

Metropolitan University
Sabbir Ahmed Talukdar
222-115-100

```cpp
#define FastRead ios_base::sync_with_stdio(false); cin.tie(0),cout.tie(0);
#define pi acos(-1)
#define MOD 1000000007
#define inf 1000010
#define endl "\n"
#define ull unsigned long long
#define con (f?"YES":"NO")

#define CLR(a) memset(a, -1, sizeof(a))
#define CLN(a) memset(a, 0, sizeof(a))

#define max3(a,b,c) max(max(a,b),c)
#define min3(a,b,c) min(min(a,b),c)
#define max4(a,b,c,d) max(a,max3(b,c,d))
#define min4(a,b,c,d) min(a,min3(b,c,d))
#define max5(a,b,c,d,e) max(max4(a,b,c,d),e)
#define min5(a,b,c,d,e) min(min4(a,b,c,d),e)

#define sortn(a,n,m) sort(a+m, s+m+n)
#define sortt(s) sort(s.begin(), s.end())
#define reversee(s) reverse(s.begin(), s.end())
#define reversesortt(s) sortt(s); reversee(s)
#define pb push_back
#define loj(i,j) "Case "<<i<<": "<<j
#define gap " "

// for (auto& x : a) cin >> x;


int main(){FastRead


}

//SNIPPET Boilerplate End
```

```cpp
vector<int>p[20005];

bool vis[20005]={};

void Bfs(int x){
   queue<int>q;
   q.push(x);

   vis[x]=true;

   while(!q.empty()){
      int u=q.front();
      q.pop();

      for(int i=0;i<p[u].size();i++){
      int v=p[u][i];
       if(!vis[v]){
           q.push(v);
           vis[v]=true;
       }
      }
   }
}
```