1. 2D Vector Input

```
#include <iostream>
#include <vector>
using namespace std;
#define pb push_back
int main (void)
{
vector <vector <int>
>v; int i,j,val;
for (i=0; i<3; i++)
{
vector <int> f;
for (j=0; j<3; j++)
{
cin >> val;
f.pb(val);
}
v.pb(f);
}
for (i=0; i<3; i++)
{
for (j=0; j<3; j++)
{
cout << v[i][j];
if (j == 2) cout
<< endl; else

cout << " ";
}
}
return 0;
}
```

2. Map

```
#include <iostream>
#include <map>
using namespace std;
int main ()
{
   ios_base :: sync_with_stdio(0);
   cin.tie(0);
   cout.tie(0);
   pair <char,int> p;
   map <char,int> mp;
   map <char,int> :: iterator i;
   for (char ch='a'; ch<='z'; ++ch)
   {
      p.first = ch;
      p.second = (int)(ch);
```

```
        mp.insert(p);
    }
    for (i=mp.begin(); i!=mp.end(); i++)
    {
        cout << (*i).first << " " << (*i).second << endl;
        /***
        EQUIVALENT
        cout << i->first << " " << i->second << endl;
        ***/
    }
    i = mp.find('k');
    cout << i->second << endl;
    return 0;
}
```

3. <u>Next Permutation</u>

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
#define pb
push_back int main ()
{
    int choose,n,i;
    char val; vector
    <char> v;
    cout << "Enter the number of elements : ";
    cin >> n;
    cout << "Enter " << n << " elements
    :\n"; for (i=0; i<n; i++)
    {
        cin >> val;
        v.pb(val);
    }
    //sort (v.begin(),v.end());
    cout << "\nPermutation of " << n << " elements :\n";
    do
    {
        for (i=0; i<n; i++)
        {
            cout << v[i];
            if (i == n-1)
                cout << '\n';
            else
                cout << " ";
        }
    }while(next_permutation(v.begin(),v.end()));
    return 0;
}
```

4. Prev Permutation

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
#define pb
push_back int main ()
{
   vector <char> v;
   char ch;
   int n,i;
   cout << "Enter the number of elements : ";
   cin >> n;
   cout << "Enter " << n << " elements : ";
   for (i=0; i<n; i++)
   {
      cin >> ch;
      v.pb(ch);
   }
   sort (v.begin(),v.end());
   reverse (v.begin(),v.end());
   cout << "Permutation of " << n << " elements :\n";
   do
   {
      for (i=0; i<n; i++)
      {
         cout << v[i];

         if (i == n-1)
            cout << '\n';
         else
            cout << " ";
      }
   }while (prev_permutation(v.begin(),v.end()));
   return 0;
}
```

5. Time

```
    #include <bits/stdc++.h>
   #include <time.h>
   using namespace std;
   int main()
   {
   clock_t begin,end;
   double time;
   int n;
   while (cin>>n)
   {
      begin=clock();
```

```
for(int i=1;i<=n;i++)
{
printf("Shawon\n");
}
end=clock();
time=(double)(end-begin)/CLOCKS_PER_SEC;
cout<<"Execution Time="<<time<<endl;
}
return 0;
}
```

6. **Binary to Decimal**

```
string str;
int len,i,dec,k;
k = len = str.size()-1;
dec = 0;
for (i=0; i<=len; i++)
   dec += (str[i]-'0') * pow(2,k--);
return dec;
```

7. **Appearance of a prime in a factorial**

```
   while (n)
   {
      n /= p;
      count += n;
   }
```

8. **Digits of a Factorial and Trailing Zeroes**

```
#include <bits/stdc++.h>
using namespace
std; int main (void)
{
   string str;
   int len,i,num,p,res,count;
   double sum;
   while (cin >> str)
   {
      len = str.size(), num = 0, p = len-1;
      /// Converting String into Int
      for (i=0; i<len; i++)
         num += (str[i] - '0') * pow(10.0,p--);
      /// Digits of the
      Factorial sum = 0;
      for (i=2; i<=num; i++)
         sum += log10(i);
      sum = (int)(floor(sum)+1);
      /// Calculating Trailing
      Zeroes p = 1;
      count = 0;
      while (1)
      {
            res = num/pow(5.0,p);
            count += res;
            p++;
            if (res <= 0)
                break;
        }
        cout << num << "! has " << sum << " digits and " << count << " trailing zeroes." << endl;
    }
    return 0;
}
```

9. <u>fibonacci series upto n</u>

```
#include <stdio.h>
int main ()
{
    unsigned long long n,a,b,sum;
    while (scanf ("%llu",&n) != EOF)
    {
        a = 0, b = 1;
        printf ("Fibonacci series upto %llu is\n",n);
        while (a <= n)
        {
            printf ("%llu",a);
            sum = a + b;
            a = b;
            b = sum;
            if (a > n)
                printf
            ("\n"); else
                printf (" ");
        }
    }
    return 0;
}
```

10. <u>fibonacci series upto nth term</u>

```
#include <stdio.h>
int main ()
{
    unsigned long long n,i,a,b,sum;
    while (scanf("%llu",&n) != EOF)
    {
        a = 0, b = 1;
        printf ("Fibonacci series upto %dth term\n",n);
        for (i=1; i<=n; i++)
        {
            printf ("%llu",a);
            sum = a + b;
            a = b;
            b = sum;
            if (i == n)
                printf ("\n");
            else
                printf (" ");
        }
    }
    return 0;
}
```
11. nth fibonacci number

```
#include <stdio.h>
int main ()
{
    unsigned  long  long  n,i,a,b,sum;
    while (scanf ("%llu",&n) != EOF)
    {
        a = 0, b = 1;
        for (i=1; i<=n; i++)
        {
            if (i == n)
                printf ("%lluth fibonacci number is %llu\n",n,a);
            sum = a + b;
            a = b;
            b = sum;
        }
    }
    return 0;
}
```
12. Sum of Big Nums

```
#include <bits/stdc++.h>
using namespace std;
#define pb push_back
int main ()
{
    string s1,s2;
    int l1,l2,i,sum,carry;
    while (cin >> s1 >> s2)
    {
        if (s1.size() > s2.size())
            swap (s1,s2);
        l1 = s1.size();
        l2 = s2.size();
        string res;
        reverse (s1.begin(),s1.end());
        reverse (s2.begin(),s2.end());
        carry = 0;
        for (i=0; i<l1; i++)
        {
            sum = (s1[i]-'0') + (s2[i]-'0') + carry;
            res.pb(sum % 10 + '0');
            carry = sum/10;
        }
        for (i=l1; i<l2; i++)
        {
            sum = (s2[i]-'0') + carry;
            res.pb(sum % 10 + '0');
            carry = sum/10;
        }
        if (carry)
            res.pb(carry + '0');
        reverse (s1.begin(),s1.end());
        reverse (s2.begin(),s2.end());
        reverse (res.begin(),res.end());
        cout << s1 << " + " << s2 << " = " << res << endl;
    }
    return 0;
}
```

13. Str to Num

```
/*** Works upto 11999999999999999999 ***/

#include <bits/stdc++.h>
using namespace std;
unsigned long long power (unsigned long long p)
{
    unsigned long long res = 1;
    while (p--)
        res *= 10;
    return res;
}
int main ()
{
    string str;
    unsigned long long n,len,i,val,p;
    while (cin >> str)
    {
        n = 0, len = str.size(), p = len-1;
        for (i=0; i<len; i++)
        {
            val = str[i] - '0';
            ///n = n * 10 + val;
            n += val * power (p--);
            /*** :D Both are Correct :D ***/
        }
        pf ("%s in STRING == %llu in INT. :)\n",str.c_str(),n);
    }
    return 0;
}
```

14. Num to Str

```
/*** Works upto 11999999999999999999 ***/
  unsigned long long n,rem;
  char ch;
  while (cin >> n)
  {
    string str;
    while (n)
    {
      rem = n % 10;
      //ch = rem+'0';
      str.pb(rem+'0');
      n /= 10;
    }
    reverse (str.begin(),str.end());
    return str;
```

15. Reverse Number

```
    temp = n, rev =
    0; while (n != 0)
    {
      rem = n % 10;
      rev = rev * 10 +
      rem; n /= 10;
    }
```

16. Sum of Div

```
#include <bits/stdc++.h>
using namespace std;
llu power (llu n,llu p)
{
  llu res = 1;

  while (p--)
    res *= n;

  return res;
}
int main ()
{
  llu n,i,sum,k;
  while (cin >> n)
  {
    sum = 1, k = 0;
    if (!(n & 1))
    {
      while (!(n & 1))
      {
        n >>= 1;
        k++;
```

```
            }
            sum *= power(2,++k)-1;
        }
        for (i=3; i*i<=n; i+=2)
        {
            if (n % i == 0)
            {
                k = 0;
                while (n % i == 0)
                {
                    n /= i;
                    k++;
                }
                sum *= (power(i,++k)-1)/(i-1);
            }
        }
        if (n > 1)
            sum *= (power(n,2)-1)/(n-1);
        cout << sum << endl;
    }
    return 0;
}
```

17. <u>Ways of Consecutive Sum</u>

```
/***
If N can be represented as sum of consecutive integers, so,
We can write,
n = a + (a+1) + (a+2) + ..... + (a+l)
=> n = (a+a+....+a) + (1+2+3+....+l)
=> n = (l+1)*a + (l*(l+1))/2
=> (l+1)*a = n - (l*(l+1))/2
=> a = (n-(l*(l+1))/2) / (l+1) ______ (i)
Here, if we let X = (l*(l+1))/2), the max value of X can be N-1.
If X be N, a will be 0.
So, we will run a loop from 1 to (l*(l+1))/2) and check everytime
if a be a whole number. If so, we will count a way to represent N
as sum of consecutive integers.
***/
#include <bits/stdc++.h>
using namespace std;
int main ()
{
    long long tc,n,l,count;
    double a;
    scanf ("%lld",&tc);
    while (tc--)
    {
        scanf ("%lld",&n);
        count = 0;
        for (l=1; l*(l+1)<2*n; l++)
        {
            a = (1.0*n-(l*(l+1))/2.0)/(l+1);
            if (a == (long long)(a))
                count++;
        }
        printf ("%lld can be represented as consecutive sum of integers in %lld ways.\n",n,count);
    }
}
```

<u>18. Tot func</u>

```
#include <bits/stdc++.h>
using namespace std;
long long totient_function (long long n)
{
    long long tot,i;
    tot = n;
    if (!(n & 1))
    {
        tot -= tot >> 1;

        while (!(n & 1))
            n >>= 1;
    }
    for (i=3; i*i<=n; i+=2)
    {
        if (n % i == 0)
        {
            tot -= tot/i;
            while (n % i == 0)
                n /= i;
        }
    }
    if (n > 1)
        tot -= tot/n;
    return tot;
}
```

19. <u>Modified Tot</u>

```
#include <bits/stdc++.h>
using namespace std; lld
power (lld n, lld p)
{
    lld res = 1;
    string  bin;
    while (p)
    {
        if (p & 1) bin
            += "1";
        else
```

```
            bin += "0";
        p >>= 1;
    }
    reverse (bin.begin(),bin.end());
    int len=bin.size(),i;
    for (i=0; i<len; i++)
    {
        res *= res;
        if (bin[i] == '1')
            res *= n;
    }
    return res;
}
bool isPrime (lld n)
{
    if (n == 2)
        return true;
    if (n < 2 || !(n & 1))
        return false;
    for (int i=2; i*i<=n; i++)
        if (n % i == 0)
            return false;
    return true;
}
lld totient (lld n)
{
    if (isPrime (n))
        return n-1;
    else
    {
        lld tot=1,i,fre;
        if (!(n & 1))
        {
            fre = 0;
            while (!(n & 1))
            {
                n >>= 1;
                ++fre;
            }
            tot *= power (2,fre-1);
        }
        for (i=3; i*i<=n; i+=2)
        {
            if (n % i == 0)
            {
                fre = 0;
                while (n % i == 0)
                {
```

```
                n /= i;
                ++fre;
            }
            tot *= power (i,fre-1) * (i-1); } }
        if (n > 1)
            tot *= (n-1);
        return tot;
    }
}
```

20. Tot with Sieve

```
/***
We know, The totient function of a number N can be determined,
phi(N) = N * (1-(1/p1)) * (1-(1/p2)) * ....... (1-(1/pn));
where p1,p2,...,pn are distinct prime factors of N.
So, we run a loop from 2 to n and check if i be prime or not. If i be prime, we will set phi[i]=i-1 and
then run a nested loop
to set its multiples totient value. Each time a prime number p appears, it will multiply its all
multiples (phi[i]) by (1-(1/p)).
Then after the end of the loop, we will get our all totient value upto N.
We can use Sieve() here, but it is an extra addition that will increase runtime by 0.01s.
It is enough to move in the simple way. :)
***/
#include <bits/stdc++.h>
using namespace std;
const int MAX = 1000005;
long long phi[MAX+1];
void totient_func ()
{
    int i,j;
    for (i=2; i<=MAX; i++)
    {
        if (phi[i] == 0)
        {
            phi[i] = i-1;
            for (j=2*i; j<=MAX; j+=i)
            {
                if (phi[j] == 0)
                    phi[j] = j;
                phi[j] = (phi[j]/i)*(i-1);
            }
        }
    }
}
```

21. Seg Sieve

```
#include <bits/stdc++.h>
using namespace std;
const long long MAX = 1000000;
vector <long long> v,sg,P;
```

```
bool prime[MAX];
void simple_sieve ()
{
   long long i,j;
   prime[0] = prime[1] = true;
   v.pb(2);
   for (i=4; i<MAX; i+=2)
      prime[i] = true;
   for (i=3; i*i<=MAX; i+=2)
   {
      if (!prime[i])
      {
         v.pb(i);
         for (j=i*i; j<=MAX; j+=2*i)
            prime[j] = true;
      }
   }
   for (i=1001; i<=MAX; i+=2)
      if (!prime[i])
         v.pb(i);
}
void segmented_sieve (long long l,long long u)
{
   long long i,j,start,x;
   for (i=l; i<=u; i++)
      sg.pb(i);
   if (l == 0)
      sg[1] = 0;
   else if (l == 1)
      sg[0] = 0;
   for (i=0; v[i]*v[i]<=u; i++)
   {
      x = v[i];
      start = x*x;
      if (start < l)
         start = ((l+x-1)/x)*x;
      for (j=start; j<=u; j+=x)
         sg[j-l] = 0;
   }
}
int main (void)
{
   simple_sieve();
   int tc,l,u,i;
   sf ("%lld",&tc);
   while (tc--)
   {
      sf ("%lld %lld",&l,&u);
```

```
        segmented_sieve (l,u);
        for (i=l; i<=u; i++)
            if (sg[i-l] != 0)
                pf ("%lld\n",sg[i-l]);
        sg.clear ();
    }
    return 0;
}
```

22. Bit Sieve

```
#include <bits/stdc++.h>
using namespace std;
const int MAX = 100000001;
int prime[(MAX>>5)+2];
vector <int> v;
int SetBit (int n, int x) {return n | (1 << x);}
int ClearBit (int n, int x) {return n & ~(1 << x);}
int ToggleBit (int n, int x) {return n ^ (1 << x);}
bool CheckBit (int n, int x) {return (bool) (n & (1 << x));}
void sievewithbitmasking ()
{
    int i,j;
    prime[0] = SetBit (prime[0],0);
    prime[0] = SetBit (prime[0],1);
    for (i=4; i<MAX; i+=2)
        prime[i>>5] = SetBit (prime[i>>5],i&31);
    for (i=3; i*i<=MAX; i+=2)
        if (!CheckBit(prime[i>>5],i&31))
            for (j=i*i; j<MAX; j+=(i<<1))
                prime[j>>5] = SetBit (prime[j>>5],j&31);
    v.pb(2);
    for (i=3; i<MAX; i+=2)
        if (!CheckBit(prime[i>>5],i&31))
            v.pb(i);
}
int main ()
{
    sievewithbitmasking ();
    for (int i=0; i<=100; i++)
    {
        pf ("%d : ",i);
        if (!CheckBit(prime[i>>5],i&31))
            pf ("Prime");
    return 0;
}
```

23. BigInt Div

```
        vector <int> v;
        l = str.size(), res = 0;
        for (i=0; i<l; i++)
        {
           div = (res*10+(str[i]-'0'))/m;
           res = ((res*10)%m+(str[i]-'0')%m)%m;
           if (v.empty() && div == 0)
              continue;
           v.push_back(div);
        }
        l = v.size();
        cout << str << " / " << m << " = ";
        for (i=0; i<l; i++)
           cout << v[i];
```

<u>24. BigInt Mod</u>

```
        len = str.size(), res = 0;
        for (i=0; i<len; i++)
           res = ((res*10)%m+(str[i]-'0')%m)%m;
     return res;
```

<u>25. BigMod</u>

```
lld bigmod (lld base, lld power, lld mod)
{
   lld p1,p2;
   if (power == 0)
      return 1;
   else if (power & 1)
   {
      p1 = base % mod;
      p2 = (bigmod (base, power-1, mod)) % mod;
      return (p1 * p2) % mod;
   }
   else if (power % 2 == 0)
   {
      p1 = (bigmod (base, power/2, mod)) % mod;
      return (p1 * p1) % mod;
   }
}
```

26. <u>Dec to</u>
<u>Double</u> Hints :

Suppose, you are given a number a and another number L. You are to find a least number b, that lcm (a,b) = L.
Now, how to do this? :)

Well, first check if a divides l or not, if no, it is impossible to find another integer beacuse a should divide L if L is the lcm. If yes, now we will follow the rules below.

(i) First divide l by a. (c = l/a)
(ii) Then get the gcd of c and a.

(iii) Run a loop while gcd (c,a) != 1 and multiply c by the gcd and divide a by the gcd.

(iv) After gcd being 1, loop will break and you have got your answer, c!!!
long long gcd (long long a,long long b)

```
{
   if (b == 0)
      return a;
   else
      return gcd (b,a%b);
}
#include <bits/stdc++.h>
using namespace std;
int main ()
{
   long long a,l,c,d;
   while (cin >> a >> l)
   {
      if (l % a != 0)
         cout << "impossible" << endl;
      else
      {
         c = l/a;
         d = gcd (c,a);
         while (d != 1)
         {
            c *= d;
            a /= d;
            d = gcd (c,a);
         }
         cout << c << endl;
      }
   }
}
```

27. Find a least number while given LCM and another number

Suppose, you are given a number a and another number L. You are to find a least number b, that lcm (a,b) = L.

Now, how to do this? :)

Well, first check if a divides l or not, if no, it is impossible to find another integer beacuse a should divide L if L is the lcm. If yes, now we will follow the rules below.

(i) First divide l by a. (c = l/a)

(ii) Then get the gcd of c and a.

(iii) Run a loop while gcd (c,a) != 1 and multiply c by the gcd and divide a by the gcd.

(iv) After gcd being 1, loop will break and you have got your answer, c!!!

```
long long gcd (long long a,long long b)
{
   if (b == 0)
      return a;
   else
      return gcd (b,a%b);
}
```

```
#include <bits/stdc++.h>
using namespace std;
int main ()
{
   long long a,l,c,d;
   while (cin >> a >> l)
   {
      if (l % a != 0)
         cout << "impossible" << endl;
      else
      {
         c = l/a;
         d = gcd (c,a);
         while (d != 1)
         {
            c *= d;
            a /= d;
            d = gcd (c,a);
         }
         cout << c << endl; } } }
```

28. <u>LCM of n Nums</u>

```
int arr[10055]; int
GCD (int a,int b)
{
   if (b == 0)
      return a;
   else
      GCD (b,a%b);
}
int main ()
{
   int n,i,gcd; lld
   lcm; while (cin
   >> n)
   {
      for (i=0; i<n; i++)
         cin >> arr[i];
      lcm = arr[0]; for
      (i=1; i<n; i++)
      {
         gcd = GCD (lcm,arr[i]);
         lcm = (lcm / gcd) * arr[i];
      }
      pf ("LCM of %d numbers :
      ",n); cout << lcm << endl;
   }
   return 0;
}
```

29. Power (BigMod)

```
#include <bits/stdc++.h>
using namespace std; lld
power (lld n, lld p)
{
    if (p == 0)
        return 1;
    if (p == 1)
        return n;
    if (p & 1)
        return n*power(n,p-1);
    else
    {
        int x = power(n,p/2);
        return x*x;
    }
}
int main ()
{
    lld n,p;
    while (cin >> n >> p)
        cout << power(n,p) << endl; return 0; }
```

30. Power (Bin Expo)

```
lld power (lld n,lld p)
{
    string bin;
    while (p)
    {
        int res = p % 2;
        if (res == 0)
            bin += "0";
        else
            bin += "1";
        p /= 2;
    }
    reverse (bin.begin(),bin.end());
    lld l = bin.size(),i,res=1;
    for (i=0; i<l; i++)
    {
        res *= res;
        if (bin[i] == '1')
            res *= n;
    }
    return res;
}
```

31. Least Prime Factor upto N

```
int arr[10000001];
void lpf (int n)
```

```
{
   int i,j;
   arr[0] = 0;
   arr[1] = 1;
   for (i=2; i<=n; i++)
   {
      if (i & 1)
         arr[i] = i;
      else
        arr[i] = 2;
   }
   for (i=3; i*i<=n; i+=2)
      if (arr[i] == i)
         for (j=i*i; j<=n; j+=2*i)
            if (arr[j] == j)
               arr[j] = i;
   for (i=1; i<=n; i++)
      printf ("Least prime factor of %d is %d\n",i,arr[i]);
}
```

32. Shortest Path

```
vector <int> edges[MAX+1],cost[MAX+1];
bool vis[MAX+1];
int lv[MAX+1];
int n,e,i,x,y,u,v,a,b,l;
void BFS (int s, int d)
{
   memset (vis,false,sizeof(vis));
   queue <int> q;
   q.push(s);
   vis[s] = true;
   lv[s] = 0;
   while (!q.empty())
   {
      u = q.front();
      q.pop();
      for (i=0; i<edges[u].size(); i++)
      {
         v = edges[u][i];
         //cout << u << " " << edges[u][i] << endl;
         if (!vis[v])
         {
            vis[v] = true;
            q.push(v);
            lv[v] = lv[u]+1;
         }
      }
   }
}
```

```
pf ("Shortest path from %d to %d is %d.\n",a,b,lv[b]);
```

33. <u>BFS in 2D</u>

```
int     lv[10][10];
bool vis[10][10];
int fx[] = {1,1,-1,-1,2,2,-2,-2};
int fy[] = {2,-2,2,-2,1,-1,1,-1};
void BFS (int a,int b,int c,int d)
{
   memset (vis,false,sizeof vis);
   queue <pii> q;
   q.push(pii(a,b));
   lv[a][b] = 0;
   vis[a][b] = true;
   while (!q.empty())
   {
      pii u = q.front();
      q.pop();
      if (u.first == c && u.second == d)
      {
         pf ("%c%d needs %d moves to reach
%c%d.\n",char(b+96),a,lv[u.first][u.second],char(d+96),c);
         break;
      }
      for (int i=0; i<8; i++)
      {
         int x = u.first+fx[i]; int
         y = u.second+fy[i];
         if (x >= 1 && x <= 8 && y >= 1 && y <= 8 && !vis[x][y])
         {
            vis[x][y]  =  true;
            q.push(pii(x,y));
            lv[x][y] = lv[u.first][u.second]+1;
         }
      }
   }
}
```

34. <u>Sqrt Decompose</u>

```
2
10 3
1 5 2 4 6 1 3 5 7 10
S 1 9
U 5 25
S 5 9
9 4
1 2 3 4 5 6 7 8 9
S 4 6
U 6 -1
S 6 6
```

```
S 1 8
int arr[1000001]; int block[1001]; int blk_size,blk_ind,i,l,r,n;
void preprocess (int n) { blk_size = sqrt(n); blk_ind = -1; for (i=0; i<n; i++)
{ if (i % blk_size == 0) ++blk_ind; block[blk_ind] += arr[i]; } }
int getSum (int l, int r) { int sum = 0; while (l<r and l!=0 and l%blk_size!=0) { sum += arr[l]; ++l; }
while (l+blk_size <= r) { sum += block[l/blk_size]; l += blk_size; } while (l <= r) { sum += arr[l];
++l; } return sum; }
void update (int ind, int val) { int blocknumber = ind/blk_size; block[blocknumber] += val - arr[ind];
arr[ind] = val; }
int main ()
{ int tc,pos,m,a,b; char str[2]; sf ("%d",&tc); for (pos=1; pos<=tc; pos++) {sf ("%d
 %d",&n,&m); for (i=0; i<n; i++) sf ("%d",&arr[i]);
 preprocess (n);
 while (m--) { sf ("%s %d %d",str,&a,&b); if (strcmp (str,"S") == 0) pf ("%d\n",getSum(a,b));
else update (a,b); } memset (block,0,sizeof block); } return 0; }
```

35. BIT implementation

```
int BIT[100001],n,i;
void update (int i, int val) { for (; i<=n; i+=i&(-i)) BIT[i] += val; }
int query (int i) { int sum = 0; for (; i>0; i-=i&(-i)) sum += BIT[i]; return sum; }
int main (void) { int val; sf ("%d",&n); memset (BIT,0,sizeof BIT);
for (i=1; i<=n; i++) { sf ("%d",&val); update (i,val); }
pf ("\ni   query(i)    BIT[i]\n");
for (i=1; i<=n; i++) pf ("%d    %2d          %2d\n",i,query(i),BIT[i]);
```

36. Largest Sum Contiguous Subarray

```
maxi = sum = arr[0]; for (i=1; i<n; i++) { sum = max (arr[i],sum+arr[i]); maxi = max (maxi,sum); }
return maxi;
```

37. BS

```
low = 0; high = n-1;
while (low <= high) { mid = (low + high) / 2; if (x == arr[mid]) break;
else if (x > arr[mid]) low = mid + 1; else high = mid - 1; }

if (low > high) return -1;
return mid+1;
```

38. Stringstream

```
string str,s;
while (getline(cin,str)) { stringstream ss(str); vector <string> v; while (ss >> s) v.push_back(s);}
```

39. ostream

```
ostringstream os; double x = 3.14159; os << x; string s = os.str(); cout << s << endl;
```

40. Modulo Inverse

```
   fact[0] = 1;

   for (i=1; i<1000001; i++)
      fact[i] = ((fact[i-1] % MOD) * (i % MOD)) % MOD;

      u = fact[n];
      d = ((fact[k] % MOD)*(fact[n-k] % MOD)) % MOD;
      res = bigmod (d,MOD-2,MOD);
      res = ((u % MOD) * (res % MOD)) % MOD;
```

<u>41. To make a vector into a set</u>
It should be sorted first._v.erase(unique(v.begin(),v.end()), v.end());

<u>42. Assigning 2D vector</u>
vector_name.assign(size_t, vi())

<u>Code Template</u>
```
/***
        Bismillahir Rahmanir Rahim
        Read in the name of Allah, who created you!!!
        Author : Shah Newaj Rabbi Shishir,
        Department of CSE, City University, Bangladesh.
***/
#include <bits/stdc++.h>
using namespace std;
#define ebar_khela_hoppe int main (void)
#define bair_ho return 0
#define sf scanf
#define pf printf
#define ssf sscanf
#define spf sprintf
#define fsf fscanf
#define fpf fprintf
#define fast ios_base::sync_with_stdio(0),cin.tie(0),cout.tie(0)
#define scase sf ("%d",&tc)
#define sn sf ("%d",&n)
#define whilecase while (tc--)
#define eof while (cin >> n)
#define forloop for (pos=1; pos<=tc; pos++)
#define arrayloop (i=0; i<n; i++)
#define cinstr cin >> str
#define getstr getline (cin,str)
#define pcase pf ("Case %d: ",pos)
#define vi vector <int>
#define pii pair <int,int>
#define mii map <int,int>
#define pb push_back
#define in insert
#define llu unsigned long long
#define lld long long
#define U unsigned int
#define endl "\n"
const int MOD = 1000000007;
const int MAX = 1000005;
int SetBit (int n, int x) { return n | (1 << x); }
int ClearBit (int n, int x) { return n & ~(1 << x); }
int ToggleBit (int n, int x) { return n ^ (1 << x); }
bool CheckBit (int n, int x) { return (bool)(n & (1 << x)); }
ebar_khela_hoppe
{
  /* freopen ("input.txt","r",stdin); freopen ("output.txt","w",stdout); */
  bair_ho;
}
```