Eliminate Number

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   int t;
   cin>>t;
   int fibo[40];
   fibo[0]=1;
   fibo[1]=2;
   for(int i=2;i<40;i++)
   fibo[i]=fibo[i-1]+fibo[i-2];
   while(t--)
   {
       int n;
       cin>>n;
       string ans="";
       int par=0;
       for(int i=39;i>=0;i--)
       {
           if(fibo[i]<=n)
           {
               par=1;
               ans+='1';
               n=n-fibo[i];
           }
           else if(par==1)
               ans+='0';
       }
       cout<<ans<<endl;
   }

}
```

Maximizing difference

```cpp
#include<bits/stdc++.h>
using namespace std;
#define endl  '\n'
#define vi  vector<int>
#include <ext/pb_ds/assoc_container.hpp>
using namespace __gnu_pbds;
typedef tree<int, null_type, less_equal<int>, rb_tree_tag,
tree_order_statistics_node_update> oset;
// order_of_key(k) : No. of elements strictly smaller than k
// find_by_order(k) : Returns iterator of second element in sorted
array (starting from 0)
long long find_maximum_difference(vector<int> a)
{
    int n = a.size();
    oset set1, set2;
    vi left(n, 0), right(n, 0);
    for (int i = 0 ; i < n ; i++) {
        int count = set1.order_of_key(a[i] + 1);
        left[i] = i - count;
        set1.insert(a[i]);
    }
    for (int i = n - 1; i >= 0; i--) {
        int count = set2.order_of_key(a[i]);
        right[i] = count;
        set2.insert(a[i]);
    }
    int ans  = INT_MIN;
    for (int i = 1 ; i < n - 1 ; i++) {
        ans = max(ans, abs(left[i] - right[i]));
    }
    return ans;
}
int32_t main()
{
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    int n; cin >> n;
    vi a(n);
    for (int i = 0 ; i < n ; i++)cin >> a[i];
    int ans = find_maximum_difference(a);
    cout << ans << endl;
    return 0;
```

```
}
```

Rahul and Substring

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   string s;
   cin>>s;
   map<char,int> mp;
   int uniq[26]={0};
   int mx=0;
   for(int i=0;i<s.size();i++)
   {
       if(uniq[s[i]-'a']==0)
       mx++;
       uniq[s[i]-'a']=1;
   }

   int len=0;
   int ans=INT_MAX;

   int low=0,high=0,find=0,temp=mx;
   while(high<=s.size())
   {
       if(find!=temp)
       {
           if(mp[s[high]]==0)
           find++;
           mp[s[high]]++;
           high++;
       }
       else
       {
           len=high-low;
           if(ans>len)
           ans=len;
           mp[s[low]]--;
           if(mp[s[low]]==0)
           find--;
           low++;
       }
   }
```

```
   cout<<ans<<endl;
}
```

Count no of alphabet

```
#include<bits/stdc++.h>
using namespace std;
const int inf=10e6;
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int n,q;
        cin>>n>>q;
        string s;
        cin>>s;
        vector<vector<int>>
l(n,vector<int>(26,-1)),S(n,vector<int>(26,INT_MAX));
        vector<int> arr(26,0);
        for(int j=0;j<s.size();j++)
        {
            int k=s[j]-'a';
            arr[k]++;
            for(int i=0;i<26;i++)
            {
                l[arr[i]][i]=max(l[arr[i]][i],j);
                S[arr[i]][i]=min(S[arr[i]][i],j);
            }
        }
        for(int i=0;i<q;i++)
        {
            char ch1,ch;
            int k;
            cin>>ch1>>ch>>k;
            if(ch1=='L')
                cout<<l[k][ch-'a']+1<<endl;
            else
                cout<<S[k][ch-'a']+1<<endl;
        }
    }
```

```
    return 0;
}
```

N bit carry adder

```
#include<bits/stdc++.h>
using namespace std;
#define int unsigned long long int
int32_t main()
{
    int q;
    cin>>q;
    while(q--)
    {
        int a,b,x=0,y=0;
        cin>>a>>b;
        int temp=a-b;
        temp/=2;
        int flag=0;
        for(int i=0;i<64;i++)
        {
            int xi=(b &(1<<i));
            int ti=(temp&(1<<i));
            if(xi==0 and ti==0)
            continue;
            if(xi>0 and ti==0)
                x=((1<<i)|x);
            else if(xi==0 and ti>0)
            {
                x=((1<<i)|x);
                y=((1<<i)|y);
            }
            else
            {
                flag=-1;
                cout<<-1<<endl;
                break;
            }
        }
        if(flag==-1)
        continue;
        cout<<min(x,y)<<" "<<max(x,y)<<endl;
    }
    return 0;
```

```
}
```

Number Selection

```
#include<bits/stdc++.h>
using namespace std;
typedef long long int lli;
#define mod 1000000007

void mul(lli a[2][2],lli b[2][2])
{
    lli w,x,y,z;
   w=(a[0][0]*b[0][0])%mod+(a[0][1]*b[1][0])%mod;
   x=(a[0][0]*b[0][1])%mod+(a[0][1]*b[1][1])%mod;
   y=(a[1][0]*b[0][0])%mod+(a[1][1]*b[1][0])%mod;
   z=(a[1][0]*b[0][1])%mod+(a[1][1]*b[1][1])%mod;
   a[0][0]=w;a[0][1]=x;a[1][0]=y;a[1][1]=z;
   return;
}
void power(lli a[2][2],lli n)
{
    if(n==0||n==1)
    return;
    lli b[2][2]={{1,1},{1,0}};
    power(a,n/2);
    mul(a,a);
    if(n%2!=0)
    mul(a,b);
}
lli fibo(lli n)
{
    lli a[2][2]={{1,1},{1,0}};
    if(n==0)
    return 0;
    power(a,n-1);
    return a[0][0]%mod;
}
int32_t main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int n;
```

```
        cin>>n;
        cout<<fibo(n+2)-1<<endl;
    }
    return 0;
}
```

Fake Palindrome

```
#include<bits/stdc++.h>
using namespace std;
#define int long long int
int solve(string s)
{
    unordered_map<int,int> mp;
    mp[0]=1;
    int ans=0,xr=0;
    for (int j=0;j<s.size();j++)
    {
       xr =xr^(1<<(s[j]-'0'));
       ans += mp[xr];
       for (int i = 0; i<26;i++)
        {
           ans+= mp[xr^(1<<i)];
       }
       mp[xr]++;
   }
   return ans;
}
int32_t main()
{
    int t;
    cin>>t;
    while(t--)
    {
         string s;
         cin>>s;
         cout <<solve(s)<< endl;
    }
    return 0;
}
```