```
//  i started at 1:03 A.M. 1/2/2022 Author:: Gorssorser  , nice to see you here , thnx for visiting
// somebody give me self control text me at my codeforces handle : gorssorser
// i am gorssorser <^v^>
#pragma GCC optimization ("O3")
#pragma GCC optimization ("unroll-loops")
#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <iostream>
#include <iomanip>
#include <stdio.h>
#include <limits.h>
#include <float.h>
#include <list>
#define  ll  long long  int
#include <vector>
#include <map>
#include <unordered_map>
#include <set>
#include <unordered_set>
#include <queue>
#include <sstream>
#include <stack>
#include <deque>
#define dd cout<<"print"<<endl;
#include <algorithm>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
#define mod1 998244353
#include <cmath>
#define run(a, m)  for(int i = 0 ; i <m;  i++  ) cin>>a[i];
#define run2(a, m)  for(int i = 0 ; i <m;  i++  ){ll v,u; \
                            cin>>u>>v; \
                            a[u]push_back(v);}

#define bhag(n,a) for(int i = a ; i < n+a; i++)
#define jldi(i ,  n , a) for(int i  = a ; i< n +a ; i++)
#define debmatrix(matrix){  cout<<"debuged matrix: "<<endl; for(auto i : matrix){for(auto j :
i){cout<<j<<"  "; } \
 cout<<endl;   }}
#define debpair(matrix){  cout<<"debuged pair: "<<endl; for(auto i : matrix){ cout<<i.first<<"
"<<i.second<<endl; } \
 cout<<endl; }
# define ff first
#define ss second
#define sz(a)     a.size()
```

```
#define all(a)   a.begin() , a.end()

# define debarr(arr){ cout<<"deb arr :"<<endl;  for(auto i : arr){ cout<<i<<" "; } cout<<endl;   }
# define debval(a){cout<<"var : "<<a<<endl;}
#define GNU_optimised_seg_tree vector<ll>
//////////////////////////////////////////////////
using namespace std;
using namespace __gnu_pbds;
typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> pbds;

using namespace std;
ll n, x, m, y ;
ll mod  =  1e9+7;
ll mul(ll a,ll b, ll mod = mod)
{
   return ((a%mod)*(b%mod)+mod)%mod;
}
ll add(ll a,ll b)
{
   return ((a%mod)+(b%mod)+mod)%mod;
}
ll power1(ll a, ll b, ll mod = mod)
{
   ll ans=1;
   while(b)
   {
      if(b&1)ans = mul(ans,a)%mod;
      a = mul(a,a)%mod;
      b>>=1;
   }
   return ans;

}
struct manachersAlgo
{
   string str ;
   string  preprocess(string str)
   {
      string temp  = "" ;

      for(auto i= 0;  i < str.size() ; i++)
      {
         temp+='#' ;
```

```
        temp+=str[i] ;
    }
    temp+='#'   ;
    return temp ;

}
vector<ll> manacher(string str)
{
    vector<ll>palin(str.size()+1)  ;
    ll l = 1 ;
    ll r = 1 ;
    palin[0] = 1 ;
    for(auto i = 1  ; i < str.size()  ; i++)
    {
        palin[i] = max(0ll, min(r-i, palin[l+r-i])) ;
        while(   (i-palin[i] >= 0 ) &&(i+palin[i] < n) &&str[i-palin[i]] ==  str[i+palin[i]] )
        {
            palin[i]++;
        }
        if(r < palin[i]+i)
        {
            r = palin[i]+i ;
            l = i-palin[i] ;
        }
    }
    return palin ;

}
ll get_largest_palin(vector<ll> &palin, ll index, ll parity)
{
    index-- ;
    if(parity)
    {
        return   palin[2*index+1] ;
    }
    else
    {
        return   palin[2*index+2] ;
    }


}
ll check_palin(ll l,ll r , vector<ll>&palin  )
{
```

```
        if( r-l+1 <= get_largest_palin(palin , (l+r)/2, (r-l+1)%2))return 1 ;
        else return 0 ;

    }


};

ll LCS(ll i, ll j, string &str1, string &str2, vector<vector<ll>> &dp)
{
    if(i  <= 0 || j <= 0  )
    {
        return  0 ;
    }
    else if(dp[i][j] != -1 ) return dp[i][j] ;
    else
    {

        ll case1 = LCS(i-1, j, str1, str2 , dp ) ;
        ll case2 = LCS(i, j-1, str1, str2, dp )   ;
        ll case3 = LCS(i-1, j-1, str1, str2 , dp) + (str1[i-1] == str2[j-1]) ;
        dp[i][j] =  max(case1, max(case2,case3)) ;
        return  dp[i][j] ;
    }




}
// Mancher algorithm for finding all the substrings which are palindrome
ll solve(ll bit)
{
      string str ;
      cin>> n ;
      cin>>str ;
      vector<vector<ll>>dp(str .size() +1, vector<ll>(str.size() + 1, -1 )) ;
      string str2 =  str ;
      reverse(all(str2)) ;
      fun(str.size()  , str.size()   , str  ,str2 , dp) ;
     ll max_ = 0 ;
     for(auto i= 0 ; i <  n ; i++){
         max_ =  max(dp[i][n - i ] , max_) ;
     }
```

```
//     debmatrix(dp)
     cout<<max_<<endl;
    return 0 ;
}
// u know sometime instead of trying to check for something just prebuild it in the correct order ,
// like sort or arrange the things accordinglyif
int  main()
{
    cin.tie(0)->sync_with_stdio(false);
    cout.tie(NULL);
    int t = 1 ;
    cin>>t;
    int counter =  1 ;
    while(t--)
    {
        int   bit  = 0 ;
        if(counter == 36  )bit = 1;
        solve(bit);
        counter++;
    }
    return 0;
}
/* greedy brute force dp3 binary search constructive pegion hole  */
/*some times negation is much preferred */
/*some times u will fall but a babmoo tree take 4 yrs to start */
/*
// a| b = a^b + a&b      a^(a&b) = (a|b)^b       b^(a&b) = (a|b)^a       (a&b)^(a|b) = a^b
// a+b = a|b + a&b      a+b = a^b + 2(a&b)
// a-b = (a^(a&b))-((a|b)^a)    a-b = ((a|b)^b)-((a|b)^a)   a-b = (a^(a&b))-(b^(a&b))   a-b =
((a|b)^b)-(b^(a&b))
*/
/*Type more ^v^*/
```