



سوال A: 811A

❖ برای حل این سوال کافی است تا زمانی که یکی از دو عدد منفی نشده اند، پیمایش کنیم و مقدار آنها را به دست بیاوریم.

C++:

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

int main(){
    int step = 1;
    int a, b;
    cin>>a>>b;
    while(a>=0 && b>=0){
        if (step%2 == 0){
            b -= step;
        }
        else{
            a -= step;
        }
        step++;
    }
    if (a<0) cout<<"Vladik\n";
    else cout<<"Valera\n";
}
```



- ❖ برای حل این سوال کافیت به ازای هر کاراکتر در خانه i کاراکترهای $i+1$ تا n را چک کنیم.
- ❖ این راه حل از مرتبه n^2 است. (دو for تو در تو که به تعداد $1 + (n-2) + (n-1) + n$ بار اجرا میشوند یعنی $(n*(n+1))/2$ چون تعداد $t = 1000$ و $n = 50$ است مرتبه زمانی این راه حل برابر $25*10^5 = 1000*(50*50)$ است. میدانیم هر یک ثانیه 10^8 است. در نتیجه زمان اجرای کد ما برابر $0.025 = 20*10^5 / 10^8$ ثانیه است. پس time limit نمیخوریم.

C++:

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

int main(){
    int t,n;
    string s;
    cin>>t;
    while (t--){
        bool flag = true;
        cin>>n>>s;
        string res = "";
        res += s[0];
        for (int j = 0; j < n-1; j++){
            if(s[j+1] != s[j]) res += s[j+1];
        }

        for (int i = 0; i < res.size(); i++){
            for (int j = i+1; j < res.size(); j++){
                if(res[i] == res[j]){
                    flag = false;
                    break;
                }
            }
            if (!flag) break;
        }
        if (flag) cout<<"YES\n";
        else cout<<"NO\n";
    }
}
```



Python:

```
for t in range(int(input())):
    n = int(input())
    s = input()
    f = True
    res = s[0]
    for i in range(int(n)-1):
        if s[i] != s[i+1]:
            res += s[i+1]
    for r in res:
        if res.count(r) > 1:
            print("NO")
            f = False
            break
    if f:
        print("YES")
```



❖ بهترین حالت برای به دست آوردن بزرگترین عدد، pause کردن پنل شماره 2 روی عدد 8 است تا اعداد با الگوی زیر به دست بیایند:

98901234567890123456...

C++:

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

int main(){
    int t,n;
    cin>>t;
    while(t--){
        cin>>n;
        if(n==1) cout<<'9'<<endl;
        else if(n==2) cout<<"98"<<endl;
        else if(n==3) cout<<"989"<<endl;
        else{
            string res="989";
            for(int i=0;i<n-3;i++)
            {
                res+=to_string(i%10);
            }
            cout<<res<<endl;
        }
    }
}
```

Python:

```
for t in range(int(input())):
    n = int(input())
    res = "989"
    for i in range(n-3):
        res += str(i%10)
    print(res[0:n])
```



C++:

```

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

int main(){
    int t,n,a,b;
    cin>>t;
    while(t--){
        string s="";
        int sq_1=0,sq_0=0;
        cin>>n>>a>>b>>s;
        if(b>=0) cout<<n*(a+b)<<endl;
        else{
            for(int i=1;i<n;i++){
                if( (s[i]=='1' && s[i-1]=='0')) sq_0++;
                else if((s[i]=='0' && s[i-1]=='1')) sq_1++;
            }
            if(s[n-1]=='1') sq_1++;
            else sq_0++;
            cout<<n*a+b*(min(sq_0,sq_1)+1)<<endl;
        }
    }
}

```

Python:

```

for _ in range(int(input())):
    n, a, b = map(int, input().split())
    s = input()
    print(n*a+n*b if b>=0 else n*a+b*(max(s.count('01'), s.count('10'))+1))

```



- ❖ برای اینکه دور وجود نداشته باشد، فقط باید یک قله داشته باشیم. برای چیدن اعداد به صورتی که این اتفاق بیفتد، دو بتوان ان منهای یک روش داریم (چرا؟)
- ❖ کل حالات نیز $n!$ است بنابراین جواب برابر است با $n! - 2^{n-1}$

C++:

```
#include <bits/stdc++.h>

using namespace std;

#define int Long Long
const int MOD = 1e9+7;

int n;
int res, fact;

signed main(){
    cin >> n;
    res = 1;
    fact = 1;
    for(int i = 1; i <= n-1; i++){
        res *= 2;
        fact *= i;
        fact %= MOD;
        res %= MOD;
    }
    fact *= n;
    fact %= MOD;
    fact -= res;
    fact %= MOD;
    if(fact < 0) fact += MOD;
    cout << fact;
    return 0;
}
```

