



## سوال A: 520A

❖ این سوال را میتوان با استفاده از ساختمان داده set حل کرد.

## C++:

```
#include <bits/stdc++.h>
using namespace std;
typedef long long int lli;
int main() {
    int n;
    string word;
    set<int> s;
    cin >> n >> word;

    for (int i = 0; i < n; i++) {
        if (word[i] >= 'A' && word[i] <= 'Z') {
            s.insert(word[i] - 'A');
        } else if (word[i] >= 'a' && word[i] <= 'z') {
            s.insert(word[i] - 'a');
        }
    }

    if (s.size() == 26) {
        cout << "YES" << '\n';
    } else {
        cout << "NO" << '\n';
    }
    return 0;
}
```

## Python:

```
n, wrd, alph = int(input()), input(), set(chr(i) for i in range(ord('a'), ord('z') + 1))
print("YES" if set(wrd.lower()) == alph else "NO")
```



❖ برای حل این سوال کافی است تا زمانی که نمیتوانیم با استفاده از منهای ۱۰ مقدار  $x$  را صفر یا کمتر کنیم، از  $x/2 + 10$  استفاده کنیم.

C++:

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

int main() {
    int t, x, n, m;
    cin >> t;
    while(t--){
        cin >> x >> n >> m;
        while(x>0){
            if (x > m*10){
                if (n > 0){
                    x = (x/2) + 10;
                    n--;
                }
                else break;
            }
            else{
                x = x-(m*10);
                m = 0;
            }
        }
        if (x <= 0) cout<<"YES\n";
        else cout<<"NO\n";
    }
}
```

Python:

```
for i in range(int(input())):
    x, n, m = map(int, input().split())
    while x > m*10 :
        if n > 0:
            x = int(x/2) +10
            n -= 1
        else: break
    if x-(m*10) <= 0: print("YES")
    else: print("NO")
```



❖ جواب این سوال با تعداد پرانتر هایی که قبل از باز شدن، بسته شده اند (یا برعکس) برابر است.

C++:

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

int main() {
    int t, n, opened, unopened; string s;
    cin >> t;
    while(t--){
        opened = 0, unopened = 0;
        cin >> n >> s;
        for (int i = 0; i < n; i++)
        {
            if (s[i] == '(' && opened == 0) unopened++;
            else if (s[i] == ')' && opened > 0) opened--;
            else if (s[i] == '(') opened ++;
        }

        cout<<unopened<<endl;
    }
}
```

Python:

```
for i in range(int(input())):
    n = int(input())
    s = input()
    while '()' in s:
        s = s.replace('()', '')
    print(len(s)//2)
```



❖ برای حل این سوال ابتدا باید دو آرایه را sort کرد. سپس میتوان روی آرایه بزرگتر پیمایش کرد و با نگه داشتن دو پوینتر، هر بار شرط وجود داشتن pair را بررسی کرد. اگر شرط برقرار نبود باید پوینتر ها را جلو ببریم، چون آرایه ها صعودی مرتب شده اند، پوینتر عنصر کوچکتر را جلوتر میبریم و دوباره شرط را بررسی میکنیم.

C++:

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

int main() {
    int n, m, l, g_pointer = 0, b_pointer = 0, pairs = 0;
    int boys[101];
    int girls[101];
    cin >> n;
    for (int i = 0; i < n; i++) cin >> boys[i];
    cin >> m;
    for (int i = 0; i < m; i++) cin >> girls[i];
    l = max(n, m);
    sort(boys, boys+n);
    sort(girls, girls+m);

    while (b_pointer < n && g_pointer < m){
        if( abs(boys[b_pointer] - girls[g_pointer]) <= 1 ) {g_pointer++; b_pointer++; pairs++; }
        else if (boys[b_pointer] > girls[g_pointer]) g_pointer++;
        else b_pointer++;
    }

    cout<<pairs<<endl;
}
```



## Python:

```
m, valM = int(input()), list(map(int, input().split()))
f, valF = int(input()), list(map(int, input().split()))

valM.sort()
valF.sort()
totalMatches = 0

for i in range(len(valM)):
    for j in range(len(valF)):
        if valM[i] >= (valF[j] - 1) and valM[i] <= (valF[j] + 1):
            # match found
            totalMatches += 1
            del valF[j]
            break
        elif valF[j] > valM[i] + 1:
            break

print(str(totalMatches))
```



- ❖ هر دنباله براکت balanced باید با '(' شروع و با ')' پایان یابد. بنابراین،  $a$  و  $b$  باید در موقعیت های اول و آخر با هم برابر باشند، وگرنه راه حلی وجود ندارد.
- ❖ تعداد کل براکت های باز در  $a$  و  $b$  باید  $n$  باشد، که زوج است. هر بیت  $i$ ، تعداد زوج براکت باز ایجاد می کند و هر بیت  $o$  تعداد فرد براکت باز ایجاد می کند. بنابراین، باید تعداد بیت های  $o$  زوج باشد در غیر این صورت راه حلی وجود ندارد. توجه داشته باشید که تعداد بیت های  $i$  نیز باید زوج باشد.
- ❖ برای ساخت  $a$  و  $b$ ، فرض میکنیم  $k$  تعداد،  $i$  وجود دارد که  $s_i = 1$  باشد. در  $a$  و  $b$  در همان اندیس های  $i$ ،  $k/2$  بار براکت باز و  $k/2$  بار براکت بسته قرار میدهیم. سپس برای بیت های صفر، تناسب با اینکه کدام یک براکت باز میگیرند، دیگری براکت بسته میگیرد.

C++:

```
#include <bits/stdc++.h>
using namespace std;
void solve()
{
    int n;
    string s;
    cin >> n >> s;
    int cnt = 0;
    for (int i = 0; i < n; i++)
        cnt += (s[i] == '1');

    if (cnt % 2 == 1 || s[0] == '0' || s.back() == '0'){
        cout << "NO\n";
        return;
    }
    string a, b;
    int k = 0;
    bool flip = false;
    for (int i = 0; i < n; i++){
        if (s[i] == '1'){
            a.push_back(2 * k < cnt ? '(' : ')');
            b.push_back(a.back());
            k++;
        }
        else{
            a.push_back(flip ? '(' : ')');
            b.push_back(flip ? ')' : '(');
            flip = !flip;
        }
    }
    cout << "YES\n"
         << a << '\n'
         << b << '\n';
}
```



```

int main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    int te;
    cin >> te;
    while (te--)
        solve();
}

```

## Python:

```

cases = int(input())
while cases:
    cases -= 1
    num = int(input())
    s = list(map(int, (list(input()))))

    if len(s) % 2 != 0 or s[0] == 0 or s[-1] == 0 or s.count(0) % 2 != 0:
        print("NO")

    else:
        ones = s.count(1)//2
        zeros = s.count(0)
        ans1 = ""
        ans2 = ""

        for n in s:
            if n == 1 and ones > 0:
                ans1 += "("
                ans2 += "("
                ones -= 1
            elif n == 1:
                ans1 += ")"
                ans2 += ")"

            elif n == 0 and zeros % 2 == 0:
                ans1 += "("
                ans2 += ")"
                zeros -= 1
            else:
                ans1 += ")"
                ans2 += "("
                zeros -= 1

        print("YES")
        print(ans1)
        print(ans2)

```

