سوال A: 1504A

- اگر رشته ما فقط از کارکتر 'a' که یک یا چند بار تکرار شده است تشکیل شده باشد ، آنگاه **پاسخی وجود ندارد** در غیر اینصورت می توانیم ادعا کنیم که جواب یا 'a' + s یا s + 'a' است یا هر دو (چرا؟)

Let us prove it. Assume for contradiction that 'a' + s and s + 'a' are both palindromes. Then the first and last characters of s are 'a'. Then the second and second to last characters of s are 'a'. Repeating this, we see that all characters of s are 'a', but we assumed we are not in this case. Therefore, the claim is true.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;

bool palindrome(const string &s)
{
    int n = s.length();
    for (int i = 0; i < n; i++)
        if (s[i] != s[n - i - 1])
            return false;
    return true;
}

int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);


    int te;
    cin >> te;
    while (te--)
    {
        string s;
        cin >> s;
        if (!palindrome(s + 'a'))
            cout << "YES\n"
                 << s << 'a' << "\n";
        else if (!palindrome('a' + s))
            cout << "YES\n"
                 << 'a' << s << '\n';
        else
            cout << "NO\n";
    }
}
```

سوال B: 1003A

- کافیست تعداد تکرار هر عدد را حساب کرده و بیشترین آن را چاپ کنیم یا به بیانی دیگر تعداد تکرار مد را در مجموعه به دست آوریم . **عدد به دست آمده جواب مسئله است** (چرا؟)

Because if there are no more than k elements with the same value in the array it is obvious that we cannot use less than k pockets, but we also do not need to use more than k pockets because of the other values can be also distributed using k pockets.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;
    cin >> n;

    vector<int> cnt(101);
    for (int i = 0; i < n; ++i)
    {
        int x;
        cin >> x;
        ++cnt[x];
    }

    cout << *max_element(cnt.begin(), cnt.end()) << endl;
    return 0;
}
```

Python:

```python
from statistics import mode

n = int(input())
ln = list(map(int, input().split()))
print(ln.count(mode(ln)))
```

سوال C: 1417A

- صورت سوال از ما **ماکسیمم تعداد باری که میتوان این عملیات را انجام داد** خواسته است. برای این که بتوانیم عملیات جمع را تعداد بار بیشتری انجام دهیم نیاز است که هربار با عدد کوچک‌تری این جمع را انجام دهیم. به عنوان مثال اگر این عملیات را بخواهیم روی دو عدد x و y انجام دهیم که x<=y، میتوانیم به یکی از دو زوج (x,y+x) و یا (x+y,y) برسیم که زوج اول برای ما بهتر است. چرا که هنوز عضو کوچک‌تری در اعداد ما باقی مانده و به کمک آن میتوانیم تعداد بار بیشتری این عملیات را تکرار کنیم.
- از نکته بالا به این نتیجه میرسیم که **کوچک‌ترین عضو آرایه را انتخاب و تا جای ممکن این روی این عضو و دیگر عضو های آرایه عملیات را انجام دهیم**. به اینصورت **حداکثر تعداد بار عملیات** را انجام داده ایم.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef pair<int, int> pii;
const int MAXN = (int)1e3 + 5;
int n, k;
int arr[MAXN];

void solve()
{
    scanf("%d %d", &n, &k);

    for (int i = 1; i <= n; ++i)
        scanf("%d", &arr[i]);

    int mn = min_element(arr + 1, arr + n + 1) - arr;
    int ans = 0;

    for (int i = 1; i <= n; ++i)
    {
        if (i != mn)
        {
            while (arr[i] + arr[mn] <= k)
            {
                arr[i] += arr[mn];
                ++ans;
            }
        }
    }
    printf("%d\n", ans);
}

int main()
{
    int tt;
    scanf("%d", &tt);

    while (tt--)
        solve();

    return 0;
}
```

سوال D: 1419A

- فرض می کنیم رقم هایی که در موقعیت فرد قرار دارند **آبی** و رقم هایی که در موقعیت زوج قرار دارند **قرمز** هستند .
- **اگر n زوج باشد** آنگاه آخرین رقم باقی مانده حتما به رنگ **قرمز** است. (چرا که بازیکن شروع کننده آبی‌ها را انتخاب میکند و تعداد کل زوج است. پس بعد از هر آبی قرمز انتخاب شده و بنابراین آخرین رقم قرمز خواهد بود.) حال که میدانیم آخرین رقم قرمز است پس اگر رقمی قرمز داشته باشیم که خود آن رقم زوج است Breach برنده خواهد بود. چرا که آن رقم را انتخاب نمی کند تا آخرین رقم باقی مانده زوج باشد و برنده شود. در غیر این صورت Raze برنده خواهد بود.
- **اگر n فرد باشد** آخرین رقم باقی مانده حتما به رنگ **آبی** است. (چرا که بازیکن شروع کننده آبی ها را انتخاب میکند و تعداد کل فرد است. پس بعد از هر آبی قرمز انتخاب شده و بنابراین آخرین رقم آبی خواهد بود.) پس اگر رقمی آبی داشته باشیم که خود آن رقم فرد است Raze برنده می شود و در غیر اینصورت Breach.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int T;
    cin >> T;

    while (T-- > 0)
    {
        int n;
        string s;
        cin >> n >> s;
        bool odd = false, even = false;
        for (int i = 1; i <= n; ++i)
        {
            if (i % 2 == 1)
                odd |= ((s[i - 1] - '0') % 2 == 1);

            else
                even |= ((s[i - 1] - '0') % 2 == 0);
        }

        if (n % 2 == 1)
            cout << (odd ? 1 : 2) << '\n';

        else
            cout << (even ? 2 : 1) << '\n';
    }

    return 0;
}
```

سوال E: 834A

- در کد C++ : با توجه به اینکه دوره تناوب این حرکات 4 است ، **باقی مانده عدد n را بر 4** محاسبه کرده و با استفاده از یک حلقه ، در هر حرکت جهت را مشخص می کنیم و در پایان با داده سوال چک می کنیم.

Python:

```python
DIRS = ['v', '<', '^', '>']

a, b = map(DIRS.index, input().split())
n = int(input())

delta = (b - a + 4) % 4

if delta == 0 or delta == 2:
    print('undefined')
elif delta == n % 4:
    print('cw')
else:
    print('ccw')
```

C++:

```cpp
#include <iostream>
using namespace std;
int main()
{
    char start, end, tmp;
    bool cw = false, ccw = false;
    long long n;
    cin >> start >> end >> n;
    n = n % 4;

    tmp = start;
    for (int i = 0; i < n; i++) // cw
    {
        switch (tmp) {
        case 118:
            tmp = 60;
            break;
        case 60:
            tmp = 94;
            break;
        case 94:
            tmp = 62;
            break;
        case 62:
            tmp = 118;
            break;
        }
    }
    if (tmp == end)
        cw = true;
    tmp = start;
    for (int i = 0; i < n; i++) // ccw
    {
        switch (tmp) {
        case 118:
            tmp = 62;
            break;
        case 60:
            tmp = 118;
            break;
        case 94:
            tmp = 60;
            break;
        case 62:
            tmp = 94;
            break;
        }
    }
    if (tmp == end)
        ccw = true;
    if (ccw && !cw)
        cout << "ccw" << endl;
    else if (cw && !ccw)
        cout << "cw" << endl;
    else
        cout << "undefined" << endl;
}
```

سوال F: 1312B

- کافیست هر آرایه رو **به صورت نزولی مرتب کرده و چاپ کنیم**.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;

void solve()
{
    int n;
    cin >> n;
    vector<int> a(n);
    for (int &x : a)
        cin >> x;
    sort(a.begin(), a.end());
    reverse(a.begin(), a.end());
    for (int b : a)
        cout << b << ' ';
    cout << '\n';
}

int main()
{
    ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    int T;
    cin >> T;
    while (T--)
        solve();
}
```

Python:

```python
for __ in range(int(input())):
    n = int(input())
    print(*sorted(map(int, input().split()), reverse=True))
```

سوال G: 237A

❖ کافیست **بیشترین تعداد حضور مشتری در یک زمان** را چاپ کنید

C++:

```cpp
#include <stdio.h>

struct collection
{
    int a, b;
} s[100000];

int main()
{
    int n, i, count = 0, max = 0;
    scanf("%d", &n);

    for (i = 0; i < n; i++)
    {
        scanf("%d %d", &s[i].a, &s[i].b);
    }

    for (i = 0; i < n - 1; i++)
    {
        if (s[i].a == s[i + 1].a && s[i].b == s[i + 1].b)
            count++;

        else
        {
            if (count > max)
                max = count;
            count = 0;
        }
    }

    if (count >= max)
        max = count;

    printf("%d", max + 1);

    return 0;
}
```

Python:

```python
visitor = dict()
for __ in range(int(input())):
    h, m = map(int, input().split())
    visitor[(h, m)] = visitor.get((h, m), 0) + 1
print(max(visitor.values()))
```