سوال A: 1436A

❖ باید در نظر داشته باشیم که هر a_i/i، i بار جمع میشود؛ پس اگر جمع مجموعه ارقام وارد شده با m برابر باشد، آنگاه باید YES چاپ شده و در غیر این صورت NO.

C++

```cpp
#include <iostream>
using namespace std;
int main(void)
{
    int t;
    cin >> t;
    while(t--)
    {
        int n,num,sum=0;
        long long m;
        cin >> n >> m;
        for(int i=0 ; i<n ; i++)
        {
            cin >> num;
            sum += num;
        }
        if(sum == m)
            cout << "YES" << endl;
        else
            cout << "NO" << endl;
    }
    return 0;
}
```

Python:

```python
for i in range(int(input())):
    n, m = map(int, input().split())
    lst = list(map(int, input().split()))
    if sum(lst) == m:
        print("YES")
    else:
        print("NO")
```

سوال B: 1092B

- در ابتدا آرایه مهارت هارا به صورت صعودی سورت میکنیم.
- حال میتوانیم مینیمم تعداد سوالی که باید حل شود تا سطح مهارت‌ها برابر شود را پیدا کنیم برای مثال a2-a1 و a4-a1 و غیره.
- فرمول کلی به این شکل میشود:

$$\sum_{i=1}^{\frac{n}{2}} a_{2i} - a_{2i-1}$$

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin >> n;
    vector<int> a;
    a.resize(n);

    for (int i = 0; i < n; ++i)
        cin >> a[i];

    sort(a.begin(), a.end());

    int res = 0;
    for (int i = 0; i < n; i += 2)
        res += a[i + 1] - a[i];

    cout << res << endl;
    return 0;
}
```

Python:

```python
n = int(input())
lst = list(map(int, input().split()))
lst.sort()
print(sum([lst[i+1] - lst[i] for i in range(0,n-1,2)]))
```

سوال C: 1272A

- با توجه به اینکه در یک دقیقه افراد تنها مجاز به یک حرکت اند بنابراین تنها مقادیر ممکن برای a، a+1 یا a-1؛ برای b، b+1 یا b-1؛ و برای c، c+1 یا c-1 می‌باشد.
- همه این مقادیر را حساب کرده و مینیمم را چاپ میکنیم.

C++:

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

int calc(int a, int b, int c) {
    return abs(a - b) + abs(a - c) + abs(b - c);
}

int main()
{
    int q;
    cin >> q;
    for (int i = 0; i < q; ++i)
    {
        int a, b, c;
        cin >> a >> b >> c;
        int ans = calc(a, b, c);
        for (int da = -1; da <= 1; ++da)
        {
            for (int db = -1; db <= 1; ++db)
                for (int dc = -1; dc <= 1; ++dc)
                {
                    int na = a + da;
                    int nb = b + db;
                    int nc = c + dc;
                    ans = min(ans, calc(na, nb, nc));
                }
        cout << ans << endl;
        }
    }
    return 0;
}
```

Python:

```python
for i in range(int(input())):
    lst = list(map(int, input().split()))
    lst.sort()
    x = sum([lst[1]-lst[0], lst[2]-lst[1], lst[2]-lst[0]]) - 4
    print( x if x > 0 else 0)
```

سوال D: 1493B

- برای حل مسئله باید با شروع از زمان داده شده، همه ی زمان ها را بررسی کنیم که ایا معکوس پذیر هستند یا نه.
- تنها نکته ی حائز توجه این است که اعداد ۰ و ۱ و ۲ و ۵ و ۸ معکوس پذیرند و دیگر اعداد با معکوس شدن نتیجه ی معتبری ندارند.
- همچنین باید بررسی کرد معکوس ساعت های داده شده، در بازه‌ی زمانی درستی باشد.

C++:

```
#include<bits/stdc++.h>
using namespace std;
int can_reflect(int input, int * reflected){
    if(reflected[input%10]== -1 || reflected[input/10]== -1)
        return -1;
    return reflected[input%10]*10+reflected[input/10];
}
int main(){
    int t;
    cin>>t;
    int reflected[10]={0,1,5,-1,-1,2,-1,-1,8,-1};
    while(t--){
        int h,m;
        cin>>h>>m;
        string input;
        cin>> input;
        int currenth = stoi(input.substr(0,2)), currentm = stoi(input.substr(3));
        while(1){
            if(can_reflect(currenth, reflected) != -1 &&
                can_reflect(currenth, reflected) <m &&
                can_reflect(currentm, reflected) != -1 &&
                can_reflect(currentm, reflected) <h){
                if(currenth<10) cout<<0;
                cout<<currenth<<":";
                if(currentm<10) cout<<0;
                cout<<currentm<<endl;
                break;
            }
            else {
                currentm++;
                if(currentm>=m){
                    currentm =0;
                    currenth++;
                }
                if(currenth>=h) currenth = 0;
            }
        }
    }
    return 0;
}
```

Python:

```python
def check(curr):
    ones,tens = curr%10,curr//10
    if ones in d and tens in d:
        return d[ones]*10 + d[tens]
    return 10**5
d = {1:1,2:5,5:2,8:8,0:0}

for _ in range(int(input())):
    h,m = map(int,input().split())
    x,y = map(int,input().split(":"))
    while True:
        if check(x)<m and check(y)<h:
            break
        y += 1
        if y==m:
            y=0
            x = (x+1)%h
    print(str(x).rjust(2,"0") + ":" + str(y).rjust(2,"0"))
```

سوال E: 611B

- برای احل این سوال تمامی اعداد باینری که تنها یک صفر دارند را تولید میکنیم و انهایی که در بازه ی [a,b] قرار دارند را میشماریم.

Python:

```python
a,b=list(map(int,input().split()))
ans=0
for i in range(1,62):
    for j in range(62):
        if a<=int("1"*i+"0"+"1"*j,2)<=b:
            ans+=1
print(ans)
```

C++:

First way:

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
int main(){
    ll a,b,ans=0;
    cin >> a >> b;
    for(ll i=0;i<64;i++){
        for(ll j=0;j<i;j++){
            ll one = (2LL<<i)-1-(1LL<<j);
            ans = one>=a && one <=b ? ans+1:ans;
        }
    }
    cout<<ans<<endl;
    return 0;
}
```

Second way:

```cpp
#include <bits/stdc++.h>
#include<bitset>
using namespace std;
typedef  unsigned long long ull;
int num_bit(ull x)
{
    int res = 0;
    while(x!=0){
        x/=2;
        res++;
    }
    return res;
}
ull BinToDec(string s){
    ull res = 0;
    int n = s.size();
    for (int i = 0 ; i <n; i++){
        int tavan = n-i-1;
        if(s[i]=='1'){
            ull power = (pow(2,tavan));
            res +=power;
        }
    }
    return res;
}
int main(){
    ull a , b, res = 0;
    cin>>a>>b;
    int bnum = num_bit(b);
    int anum = num_bit(a);
    for (int i =  anum; i<=bnum; i++)
    {
        string s;
        s.clear();
        for (int j = 0; j< i; j++)
            s+="1";
        for (int j = 1; j< i; j++){
            s[j] = '0';
            ull dec = BinToDec(s);
            if(dec>=a && dec <= b)
                res++;
            s[j] = '1';
        }
    }
        cout<<res<<endl;
    return 0;
}
```