سوال ۱: ثبت نام گروهی

با تقسیم تعداد افراد و به دست آوردن باقی مانده شرط گذاری می‌کنیم.

Python:

```python
n = int(input())

if n % 3 == 0:
    print('OK')
else:
    print('HELP')
```

C++:

```cpp
#include <iostream>
using namespace std;
int main(){
    int n;
    cin >> n;

    if (n%3==0) {
        cout << "OK";
    } else{
     cout << "HELP";
    }

    return 0;
}
```

سوال ۲: برگزاری صبحگاه

با پیدا کردن حروف تکراری و حذف آن ها می‌توان به پاسخ رسید. روش اول به این صورت است که تمامی حروف های یک رشته را با رشته بعدی مقایسه کنیم و به دنبال مشابه ها بگردیم، روش دوم این است که تنها با جستوی حروف های صدادار حروف بعدی را حذف کنیم، و روش سوم جایگزین کردن aa با a است.

Python:

```python
s = input()

i = 0
res = ''

while i < len(s):
    res += s[i]
    if (i != len(s)-1) and (s[i] == s[i+1]):
        i += 1
    i += 1

print(res)
```

روش اول) با استفاده از یک حلقه تک به تک حروف ها را بررسی کرده و در صورت وجود حروف مشابه از حروف بعدی آن پرش می‌کنیم. (این راه حل به شرطی است که دو حروف بی صدا پشت هم قرار نگیرند)

```python
s = input()

s = s.replace('aa', 'a')
s = s.replace('oo', 'o')
s = s.replace('uu', 'u')
s = s.replace('ii', 'i')
s = s.replace('ee', 'e')

print(s)
```

روش دوم با جایگزین کردن حروف.

بهترین حل توسط، *مرتضی نظرزاده*:

```python
word=list(input())
vowel=['a','i','o','u','e']
x=[]
for i in range(0,len(word)-1):
    if word[i]==word[i+1] and word[i] in vowel:
        continue
    else:
        x.append(word[i])
x.append(word[len(word)-1])
for i in range(0,len(x)):
    print(x[i],end='')
```

C++:

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

using namespace std;

int main(){
    string s;
    cin >> s;

    s.erase(unique(s.begin(), s.end()), s.end());

    cout << s;
}
```

با استفاده از دستور خط ۹، می‌توان تمامی حروف های مشابه را پاک کنیم، (این راه حل به شرطی است که دو حروف بی صدا پشت هم قرار نگیرند)

سوال ۳: استخدام معلم

این سوال با روش و پیچیدگی زمان های مختلفی قابل حل است، به صورت کلی حروف های مشابه باید بین دو عبارت پیدا شوند، یک روش نوشتن شرط برای تمامی حالت ها در هر دو حروف است، روش دیگر مقایسه حروف های عبارت اول با حروف های عبارت دوم است که می‌توان با پیچدگی زمان O(n) و O(n2) به آن رسید. روش اول با پیچیدگی زمانی O(n) از ارسال توسط *مرتضی نظرزاده:*

```
word1=input()
word2=input()
x=[]
y=[]
if len(word1)<=len(word2):
    x=word1
    y=word2
else:
    x=word2
    y=word1
result=[]
for i in x :
    if i in y:
        if not i in result:
            result.append(i)
print(len(result))
```

توضیح به این صورت است که اگر حروف کلمه اول در حروف دوم وجود داشت، و این حروف قبلا به متغیر result اضافه نشده بود یعنی این یک حروف مشابه و جدید است. در نتیجه ما یک متغیر از تمام حروف های مشابه خواهیم داشت.

روش دیگر نیاز به آشنایی با ASCII دارد. هر حروف طبق استانداردی عدد گذاری شده است که با استفاده از دستور chr() می‌توان اعداد را به کاراکتر مربوط تبدیل کرد.

```python
a, b= input(), input()
cnt = 0

for i in range(97, 123):
        letter = chr(i)
        if (letter in a) and (letter in b):
                cnt += 1

print(cnt)
```

C++:

ابتدا حروف های تکراری را پاک کردیم تا از هر حروف یکی داشته باشیم، سپس تمامی حروف های کلمه اول را با تمامی حروف های کلمه دوم مقایسه کردیم تا مشترک ها را پیدا کنیم

پیچیدگی زمان این روش O(n2) است.

```cpp
#include <bits/stdc++.h>
using namespace std;
int main(){
    string a;
    string b;

    cin >> a >> b;

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

    a.erase(unique(a.begin(), a.end()), a.end());
    b.erase(unique(b.begin(), b.end()), b.end());

    int cnt = 0;
    for (int i=0; i<a.size();i++){
        for (int j=0;j<b.size(); j++){
            if (a[i] == b[j]){
                cnt += 1;
            }
        }
    }
    cout << cnt;
}
```

سوال ۴: صندوق مدرسه

نکته: برخی از کد های ارسال، راه حلی بسیار خوب و درست داشتند اما به دلیل استفاده از حروف کوچک در خروجی(مثل open به جای OPEN) هیچ نمره ای نگرفتند

به صورت کلی باید ارقام را یکی یکی تقسیم کنیم و بررسی کنیم آیا عدد بخش پذیر هست یا نه.

منظور این است که هر یک از ارقام عدد خط دوم که عدد آقای مهربان هست باید بررسی شود، اگر ارقام عدد آقای مهربان به صورت زیر باشند

Mehraban[0], Mehraban[1], Mehraban[2]

و اعداد گاوصندوق

Lock[0], Lock[1], Lock[2]

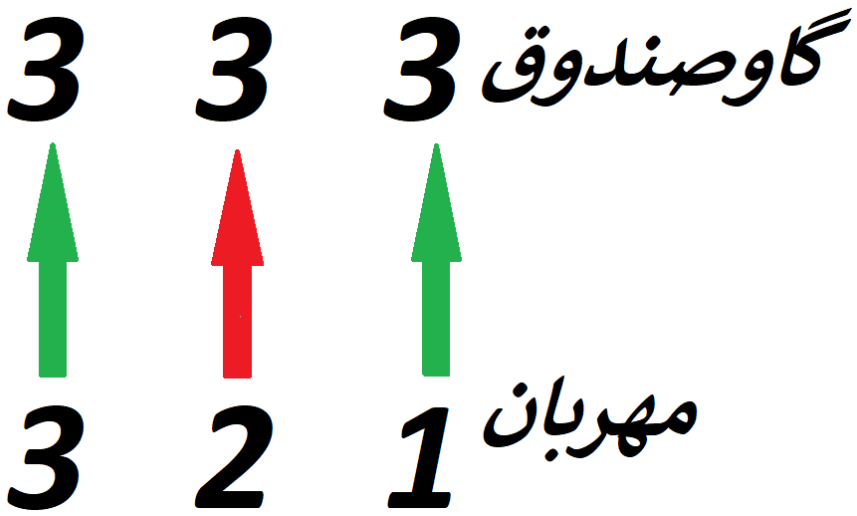
باشند. باید برای هر رقم اقای مهربان فرآیند تقسیم اعداد گاوصندوق بر آن را انجام داد تا ببینیم آیا درست یا نه

Lock[0] % Mehraban[0] = 0

Lock[1] % Mehraban[0] = 3

Lock[2] % Mehraban[0] = 1

در اینجا یکی از ارقام بخش پذیر است در نتیجه عدد آقای مهربان نیز درست است.



گام اول) جدا سازی ارقام است که می‌تواند با استفاده از تبدیل int به string انجام شود و یا با استفاده از الگوریتم ریاضی.

Python:

```python
n, m = input(), input()

res = m
for i in n:
    for j in m:
        if int(i)%int(j) == 0:
            res = res.replace(j, '')

if res == '':
    print("OPEN")
else:
    print('LOCKED')
```

در این کد، دو ورودی به صورت string از کاربر گرفته می‌شود، و تمامی ارقام عدد اول به ارقام عدد دوم تقسیم خواهند شد، اگر همگی بخش پذیر بودند(در این کد اگر res برابر خالی یعنی '' بود) دستور OPEN و در غیر این صورت دستور LOCKED نمایش داده شود.

C++:

در خط ۱۶ ما دو لیست متشکل از تمامی ارقام دو عدد داریم. حال ارقام عدد اول را به ارقام عدد دوم تقسیم می‌کنیم اگر در بین این تقسیم ها یکی از اعداد به هیچکدام از ارقام عدد دوم بخش پذیر نبود، به کاربر LOCKED را نشان می‌دهیم و برنامه را پایان می‌دهیم، در غیر این صورت عبارت OPEN را چاپ می‌کنیم.

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

    int a = n;
    int b = m;
    vector<int> v, z;
    while (a!=0 || b!=0) {
        v.push_back(a%10);
        z.push_back(b%10);
        a /= 10;
        b /= 10;
    }
    // for (int i=0;i<v.size();i++) cout << v[i] << "    " << z[i] << endl;
    for (int i=0; i<z.size();i++){
        int cnt = 0;
        for (int j=0;j<v.size();j++) {
            if (v[j] % z[i] == 0) {
                cnt++;
            }
        }
        if (cnt == 0) {
            cout << "LOCKED";
            return 0;
        }
    }
    cout << "OPEN";
}
```

سوال ۵: بحران ماژیک

به صورت خلاصه سه گام داریم

- **پیدا کردن اعداد مشابه**
- **تبدیل اعداد مشابه به صفر(به جز یکی از آن ها)**
- **مرتب کردن اعداد**

برای پیدا کردن اعداد مشابه از روش های بسیاری می‌توان عمل کرد یکی از ساده ترین آن ها تبدیل list به set است، در واقع ورودی را به شکلی بگیریم که در یک مجموعه با المان های منحصر به فرد ذخیره شوند.

Python:

```python
marks = list(map(int, input().split()))

n = len(marks)

marks = list(set(marks))
if len(marks) != n:
    tmp = n - len(marks)
    for i in range(tmp):
        marks.append(0)

marks.sort()
for x in marks:
    print(x, end=' ')
```

بهترین ارسال از *مرتضی نظرزاده:*

```python
number=list(map(int,input().split()))
for i in number:
    while number.count(i)!=1 and i!=0:
        number.remove(i)
        number.append(0)
number.sort()
for i in number:
    print(i,end=' ')
```

در خط سوم اگر چندبار از عددی وجود داشته باشد آن عدد پاک خواهد شد و به جای آن صفر قرار خواهد گرفت.

روش دیگر با استفاده از نوشتن تمامی حالت ها از *آریا صیانتی*:

```python
input01 = input()

list_input01 = []

abc = 0
abcd = 0


list_input01 = input01.split(" ")

list_input01 = list(dict.fromkeys(list_input01))

list_input01.sort()

# print(list_input01)

if len(list_input01) == 5:
    print(list_input01[0], list_input01[1], list_input01[2], list_input01[3], list_input01[4])
if len(list_input01) == 4:
    print(0, list_input01[0], list_input01[1], list_input01[2], list_input01[3])
if len(list_input01) == 3:
    print(0, 0, list_input01[0], list_input01[1], list_input01[2])
if len(list_input01) == 2:
    print(0, 0, 0, list_input01[0], list_input01[1])
if len(list_input01) == 1:
    print(0, 0, 0, 0,list_input01[0])

```

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;
int main(){
    vector<int> v;
    for (int i=0;i<5;i++){
        int tmp;
        cin >> tmp;
        v.push_back(tmp);
    }

    int len = v.size();

    sort(v.begin(), v.end());
    v.erase(unique(v.begin(), v.end()), v.end());

    if (len != v.size()) {
        len -= v.size();
        for (int i=0;i<len;i++) {
            v.push_back(0);
        }
    }
    sort(v.begin(), v.end());
    for (int i=0; i<v.size();i++){
        cout << v[i] << " ";
    }
}
```

شرکت کنندگان از زبان های php، python، C++ برای حل سوال ها استفاده کردند.

سوالات به روش های مختلفی قابل حل بودند، در اینجا تنها یک یا دو نمونه از روش های حل قرار گرفته شد.

با تشکر از وقت شما،

با امید یادگیری بیشتر برای همگان.