



سوال A: 148A

حل از طریق Brute Force

C++:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int k, l, m, n, d, count = 0;

    cin >> k >> l >> m >> n >> d;
    for (int i = 0; i < d; i++)
    {
        if ((i + 1) % k == 0 || (i + 1) % l == 0 || (i + 1) % m == 0 || (i + 1) % n == 0)
            count++;
    }

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

Python:

```
k, l, m, n, d = (int(input()) for _ in range(5))

print(sum(1 for i in range(1, d + 1) if i % k == 0 or i % l == 0 or i % m == 0 or i % n == 0))
```



از مقدار پولمان تا جایی که امکانش باشد از اسکناس با بیشترین مقدار کم کرده و تا جایی که مقدار پولمان 0 شود ادامه می دهیم و تعداد اسکناس ها را ذخیره می کنیم

C++:

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long
int main()
{
    ll int n, max = 0, count = 0;
    cin >> n;
    ll int *m = new ll int[5];
    m[0] = 1;
    m[1] = 5;
    m[2] = 10;
    m[3] = 20;
    m[4] = 100;
    max = 100;
    int i = 4;
    while (n != 0)
    {
        while (max > n)
        {
            i--;
            max = m[i];
        }
        n -= max;
        count++;
    }
    cout << count;
    return 0;
}
```

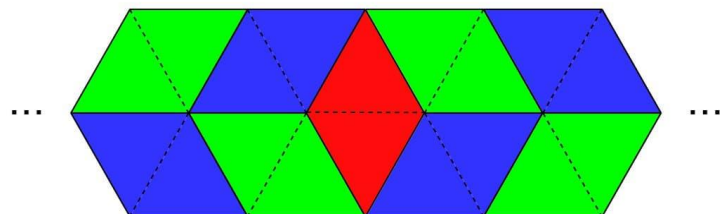


Python:

```
a = int(input())
change_list = [100, 20, 10, 5]
bills = 0
for i in change_list:
    if (a > 0):
        t = a//i
        bills += t
        a -= (t*i)
    else:
        break
bills += a
print(bills)
```



سوال C: 1339A



نکته کلیدی : هر کجا که الماس عمودی را در یک نقطه قرار دهیم ، تمامی مکان های دیگر به طور منحصر به فرد توسط یک الماس افقی مانند تصویر رو به رو قرار می گیرد .

پس جهت حل سوال ، کافست به ازای هر تست کیس عدد ورودی را چاپ کنید :

C++:

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long
int main()
{
    int n;
    cin >> n;
    while (n--)
    {
        ll int t;
        cin >> t;
        cout << t << endl;
    }
    return 0;
}
```

Python:

```
for __ in range(int(input())):
    print(int(input()))
```



سوال D: 1326B

❖ می دانیم که $b_i = a_i - x_i$ پس $a_i = b_i + x_i$ ، حال از ابتدا (از یک تا n) مقادیر a را محاسبه میکنیم:

$x = 0$ ----- $\rightarrow a_1 = a[1] = b[1] + 1$ ----- $\rightarrow x = \max(0, a_1)$ ----- $\rightarrow a[2] = b[2] + x_2$ ----- $\rightarrow x = \max(0, a[1], a[2]) = \max(\max(0, a[1]), a[2]) = \max(x_2, a[2])$ ----- \rightarrow and so on!

C++:

```
#include <bits/stdc++.h>
using namespace std;
long long n, a, b = 0;

int main()
{
    cin >> n;
    while (n--)
    {
        cin >> a;
        cout << a + b << ' ';
        b = max(b, a + b);
    }
    return 0;
}
```

Python:

```
input()
ls = list(map(int, input().split()))
b = 0
for a in ls:
    print(a + b, end=' ')
    b = max(b, a + b)
```



سوال E: 579A

با توجه به اینکه هر باکتری هر روز دو برابر می شود - < در نتیجه هر باکتری فقط می تواند توانی از دو رو تولید کند

پس تنها کفایت **تعداد بیت های 1** عدد وارد شده را محاسبه و چاپ کنیم . (چرا؟)

C++:

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

```
int main()
{
    long long x;
    cin >> x;
    cout << __builtin_popcountll(x) << endl;
    return 0;
}
```

//_builtin_popcount(x): This function is used to count the number of one's (set bits) in an integer.

Python:

```
print(bin(int(input()))).count('1'))
```



سوال F: 37A

$a[m]$: تعداد میله هایی که طول m دارند

k : تعداد میله ها با طول های متمایز است

اگر $a[m]$ صفر باشد، و سپس یکی اضافه شود، یک میله ی جدید داریم که با قبلی ها متمایز است

هر بار بررسی میکنیم بیشترین تعداد رو کدام نوع میله دارد.

C++:

```
#include <bits/stdc++.h>
using namespace std;
int n, k, s, a[10001], m;
int main()
{
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cin >> m;
        if (a[m] == 0)
            k++;
        a[m]++;
        s = max(s, a[m]);
    }
    cout << s << " " << k << endl;
    return 0;
}
```



Python:

```
input()
bars = list(map(int, input().split()))
towers = {}

for b in bars:
    towers[b] = towers.get(b, 0) + 1

print(max(towers.values()), len(towers))
```



می دانیم که جمع تعامی عناصر موجود در b برابر است با دو برابر جمع تعامی عناصر $a + x$

فرض می کنیم B جمع تعامی عناصر موجود در b باشد

اکنون با پیمایش روی عناصر مشخص می کنیم کدام عنصر به عنوان مجموع اعضای a در این آرایه قرار دارد . پس $x = B - 2 \cdot A$

حال تنها کاری که باید انجام داد ، بررسی وجود x در آرایه b است (این کار با استفاده از Binary Search Tree امکان پذیر است)



C++:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int t;
    cin >> t;
    while (t--)
    {
        int n, i, j;
        cin >> n;
        long long b[n + 2], sum = 0;
        for (i = 0; i < n + 2; i++)
            cin >> b[i];
        sort(b, b + n + 2);
        for (i = 0; i < n; i++)
            sum += b[i];
        if (sum == b[n])
            for (i = 0; i < n; i++)
                cout << b[i] << " ";
        else
        {
            sum += b[n];
            for (i = 0; i < n + 1; i++)
                if (sum - b[i] == b[n + 1])
                    break;
            if (i < n + 1)
                for (j = 0; j < n + 1; j++)
                    if (j != i)
                        cout << b[j] << " ";
            else
                cout << "-1";
        }
        cout << endl;
    }
}
```

