657div2

A. Acacius and String

题意:给定一个字符串，可以把?换成任意字符，使得一个给定的字符串只能出现一次,

题解：暴力即可，考试时手滑了被卡了一个小时

```
string s;
int n;
string T = "abacaba";
bool check(string &a)
{
    int cnt = 0;
    rep(i, 0, n - 7)
    {
        if (a.substr(i, 7) == T)
            cnt++;
    }
    return cnt == 1;
}
int main()
{
    int t;
    sd(t);
    while (t--)
    {
        cin >> n >> s;
        int flag = 0;
        rep(i, 0, n - 7)
        {
            string ss = s;
            bool ok = 1;
            rep(j, 0, 6)
            {
                if (ss[i + j] != T[j] && ss[i + j] != '?')
                {
                    ok = 0;
                    break;
                }
                ss[i + j] = T[j];
            }
            if (ok && check(ss))
            {
                rep(j, 0, n - 1)
                {
                    if (ss[j] == '?')
                        ss[j] = 'z';
```

```
                }
                puts("Yes");
                flag = 1;
                cout << ss << endl;
                break;
            }
        }
        if (!flag)
            puts("No");
    }
    return 0;
}
```

B. Dubious Cyrpto **题意：** 有三个整数 a,b,c 满足$l\Leftarrowa,b,c\Leftarrowr$还有一个整数$m = n*a+b-c$，n 是正整数。$给定l,r,m$求出$a,b,c$的一组可能值

题解

可以求出$m-b+c$的范围，然后暴力枚举a，判断是否存在n即可。

```
#include <algorithm>
#include <cstdio>
#include <cstring>
#include <iostream>
using namespace std;
typedef long long  ll;
int main()
{
    int t;
    cin >> t;
    while(t--)
    {
        ll l, r, m;
        cin >> l >> r >> m;
        ll LL = max(0ll, m + l - r), RR = m + r - l;
        ll a;
        ll n;
        for (a = l; a <= r;a++)
        {
            ll tem = (LL / a) * a;
            if( tem >=LL && tem <= RR && LL/a!=0)
            {
                n = LL / a;
                break;
            }
            tem += a;
            if( tem >=LL && tem <= RR)
            {
                n = LL / a + 1;
```

```
                break;
            }
        }
        ll d = a * n - m;
        ll b, c;
        if(d <= 0)
        {
            b = r;
            c = d + b;
        }
        else
        {
            c = r;
            b = c - d;
        }
        cout << a << " " << b << ' ' << c << endl;
    }
    return 0;
}
```

C. Choosing flowers

题意：有 m 种花，每种花第一次购买获益a[]之后在购买购买获益b[]问买n朵花，最大收益。

题解[]

发现只有一种花会被购买2朵以及以上，否则必定获益相等于是也可转化为前一种情况，那只要枚举哪一种花购买多次即可。

```
#include<bits/stdc++.h>
using namespace std;
#define pll pair<long long, long long>
pair<long long, long long>x[100005];
long long sum[100005];
int upper(int l,int r,long long num)
{
    int ans = -1;
    while (l <= r)
    {
        int mid = (l + r) >> 1;
        if (x[mid].first > num)
        {
            ans = mid;
            l = mid + 1;
        }
        else r = mid - 1;
    }
    return ans;
}
```

```
int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        int n, m;
        cin >> n >> m;
        int i;
        for (i = 0; i < m; i++)
        {
            cin >> x[i].first >> x[i].second;
            sum[i] = 0;
        }
        sort(x, x + m, greater<pair<long long, long long>>());
        for (i = 0; i < m; i++)
        {
            //cout << x[i].first << ' ' << x[i].second << endl;
            if (i)sum[i] = x[i].first + sum[i - 1];
            else sum[i] = x[i].first;
        }
        long long ans = 0;
        for (i = 0; i < m; i++)
        {
            int p = upper(0, m-1, x[i].second);
            if (p != -1)
            {
                ans = max(ans, sum[min(p, n-1)] + (p < i ? max(0, n - p -
2) * x[i].second + (n>p+1?x[i].first:0) : max(0, n - p-1) * x[i].second));
            }
            else
            {
                ans = max(ans,  (n-1) * x[i].second + x[i].first);
            }
        }
        long long ans2 = 0;
        for (i = 0; i < m && n; i++)
        {
            ans2 += x[i].first;
            n--;
        }
        cout << max(ans2, ans) << endl;
    }
    return 0;
}
```