

A题：秋季特惠

等差数列求和水题。

```
#include <bits/stdc++.h>
using namespace std;
const int maxn = 1005;
typedef long long ll;
int main()
{
    ll T, k, n, w;
    cin >> T;
    while(T--) {
        cin >> k >> n >> w;
        ll t = (k + w * k) * w / 2 - n;
        if(t <= 0) cout << 0 << endl;
        else cout << t << endl;
    }
    return 0;
}
```

B题：找工作

这题注意，当 $n = 0$ 时，输出0， $n! \neq 0$ ，系数全为0时输出为空，还有末尾不要带空格

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

int main()
{
    int T,k,n;
    cin >> T;
    while(T--){
        vector<int>vec;
        cin >> k;
        bool flag = true;
        bool zero = true;
        for(int i = k; i >= 0; --i){
            cin >> n;
            if(n)zero = false;
            if(i == 0) break;
            if(n == 0 && flag) {
                continue;
            }
            flag = false;
            vec.push_back(n * i);
        }
        if(zero){
            cout << 0 << endl;
        }
    }
}
```

```

        continue;
    }
    if(k == 0) {
        cout << 0 <<endl;
        continue;
    }
    for(int i = 0 ;i < vec.size() - 1; ++i){
        cout << vec[i] << " ";
    }
    cout <<vec[vec.size() - 1] <<endl;
}
return 0;
}

```

C题：购物计划

这题水的一批，因为任意n个相邻数的最大公因数为1，所以这题只要疯狂输出1即可。

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int T, n;
    cin >> T;
    while(T--){
        cin >> n;
        cout << n <<endl;
        for(int i = 0; i < n - 1; ++i)
            cout << 1<< " ";
        cout << "1" <<endl;
    }
    return 0;
}

```

D题：蟹黄堡的配方

这题是简单的字符串比较问题。

```

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

int main()
{
    int T, n;
    string s1,s2;
    cin>> T;
    while(T--){
        bool flag = true;
        string s;
        cin >> n;
        cin >> s1 >> s2;
    }
}

```

```

        for(int i = 0; i < s1.size(); ++i){
            if(s1[i] >= s2[i]){
                s += s2[i];
            }
            else {
                cout << -1 << endl;
                flag = false;
                break;
            }
        }
        if(flag) cout << s << endl;
    }
    return 0;
}

```

E题：大圈圈

欧拉示性数公式

著名数学家欧拉(Euler, 1707-1783)给出一个公式 $v - e + f = 2$, 其中 v 是顶点数, e 是棱数, f 是面数.

n 个圆, 两两相交, 则 $v = 2 * C(n, 2) = n(n - 1)$

其中 $C(n, 2)$ 是从 n 个元素中选两个元素的组合

$e = n * (2(n - 1)) = 2 * n * (n - 1)$, 这个式子的含义是 n 个圆, 每个圆都被其余 $n - 1$ 个圆分出 $2(n - 1)$ 条线段, 由欧拉公式, $f = e - v + 2 = 2 * n * (n - 1) - n * (n - 1) + 2 = n^2 - n + 2$

故答案是 $n^2 - n + 2$

```

#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int mod = 1000000007 ;
int main()
{
    int T, n;
    cin >> T;
    while(T--) {
        cin >> n;
        cout << (n * n - n + 2) % mod << endl;
    }
    return 0;
}

```

F题：恶龙与勇者

这题要对恶龙的实力从小到大排序, 对勇者的花费从小到大排序。

```

#include<bits/stdc++.h>
using namespace std;
const int maxn = 10000;

```

```

typedef pair<int,int> P;
typedef long long ll;
P arr[maxn + 5];
int attack[maxn + 5];
bool cmp(P a, P b)
{
    return a.second < b.second;
}
int main()
{
    int T, n, m, t;
    cin >> T;
    while(T--){
        ll sum = 0;
        bool flag = true;
        cin >> n >> m;
        for(int i = 0; i < n; ++i) cin >> attack[i];
        for(int i = 0; i < m; ++i) cin >> arr[i].first >> arr[i].second;
        sort(attack, attack + n);
        sort(arr, arr + m, cmp);
        for(int i = 0; i < n; ++i){
            int k = 0;
            while(arr[k].first < attack[i] && k < m)k++;
            if(k >= m){
                flag = false;
                break;
            }
            sum += arr[k++].second;
        }
        if(flag) cout << sum << endl;
        else cout << "kingdom fall" << endl;
    }
}

```

G题：搭积木

贪心思维题。

```

#include <cstdio>
using namespace std;
int n, tp, pre = 0, ans;
int main()
{
    scanf("%d", &n);
    for (int i = 1; i <= n; i++)
    {
        scanf("%d", &tp);
        if (tp >= pre) ans += tp - pre, pre = tp;
        else pre = tp;
    }
    printf("%d\n", ans);
}

```

```
}
```

H题：一条数学题

题目大意：列出从一个给定上界的双平方数集合中选出若干个数，组成长度为 N 的等差数列的首项和公差。

题解：首先，因为是在双平方数集合上的等差数列，而且根据题目范围可知，上界不超过 $2e5$ ，可以先打表，将符合条件的双平方数存入一个数组，并排序离散化。**在等差数列中，只要数列中的前两项确定，整个数列就会被确定下来。**因此，考虑枚举序列的前两项，至此时间复杂度为 $O(cnt^2)$ 。接着，若直接枚举后面的项来判断是否长度可以到达 N 的话，时间复杂度还要多一个 cnt 的乘积因子，难以承受。其实，在枚举的过程中，产生了大量的无效结果，最后才判断导致时间复杂度暴增，同时根据双平方数集合不大，可以直接用桶来装，因此，可以直接枚举 N ，每次判断数列的下一项是不是在双平方数集合中即可，时间复杂度为 $O(cnt^2 * N)$ 。

```
#include<iostream>
#include<cmath>
#include<cstring>
#include<cstdio>
using namespace std;
int n,m,p,q,s[125000];
bool num[125000];
int main()
{
    while(cin >> n){
        cin>>m;
        memset(num,false,sizeof(num));
        memset(s,0,sizeof(s));
        int i,j,a,b,l,cnt=0;
        for(i=0;i<=m;i++)
            for(j=0;j<=m;j++)
                num[i*i+j*j]=1;
        l=0;
        for(i=0;i<=125000;i++)
            if(num[i]) s[++l]=i;

        for(b=1;b<s[l];b++)
            for(i=1;i<l-1;i++){
                if(s[i]+(n-1)*b>s[l]) break;
                for(a=2;a<=n;a++)
                    if(!num[s[i]+(a-1)*b]) break;
                if(a==n+1&&num[s[i]+(n-1)*b])
                    printf("%d %d\n",s[i],b),cnt++;
            }
        if(!cnt) printf("NONE\n");
        cout <<endl;
    }

    return 0;
}
```