

**A****B****C****D**

贪心

n个人共m道菜，循环选取k道，每个人对不同菜的喜爱程度给出，每个人都希望这k道菜的喜爱程度的和最大

感性地，预知前人的选择是没有意义的，没人能对干预过去，轮到他们的时候前人的结果就已经确定了

只会考虑后人的选择，那么，最后一个人一定有能力选择自己最喜欢的菜，以此类推

每个人只需要点后人点过的菜以外自己最喜欢的菜，就可以进行回推了

**AC代码**

```
#include <bits/stdc++.h>
using namespace std;
typedef pair<int,int> PII;
const int N = 2010;
int a[N][N];
PII b[N][N];
int vis[N];
int main(){
    int t;
    cin>>t;
    while(t--){
        int n,m,k;
        cin>>n>>m>>k;
        for(int i=0;i<n;++i) for(int j=0;j<m;++j) scanf("%d",&a[i][j]);
        int count = 0;
        for(int i=0;i<k;++i){
            if(count==n) count=0;
            for(int j=0;j<m;++j){
                b[i][j] = {a[count][j],j};
            }
            count++;
        }
        for(int i=k-1;i>=0;--i){
            sort(b[i],b[i]+m);
            for(int j=m-1;j>=0;--j){
                if(!vis[b[i][j].second]){
                    vis[b[i][j].second] = 1;
                    break;
                }
            }
        }
    }
}
```

```
    }  
  }  
  for(int i=0;i<m;++i) if(vis[i]) printf("%d ",i+1);  
  printf("\n");  
  memset(vis,0,sizeof(int)*m);  
}  
  
}
```

## E

这个数量级的 $k$ 基本一眼枚举,那么问题其实就变成了对于固定的 $x$ 和 $10$ 的幂次,如何快速找到满足条件的 $y$

由于这个函数单调,只需要找到第一个使得该函数大于等于乘积的 $y$ 即可,我的第一感觉是二分,直接让队长写了

赛后一想这个函数太简单了可以直接开方,然后稍微调整一下即可剩省一个 $\log$ 的复杂度,对于复杂的单调函数可以考虑二分

## AC代码

```
#include <bits/stdc++.h>  
typedef long long ll;  
using namespace std;  
int main(){  
    int t;  
    cin>>t;  
    while(t--){  
        int flag = 0;  
        ll ans = 0;  
        ll tmp = 1;  
        ll x;  
        cin>>x;  
        for(int k=0;k<18;++k){  
            ans = (ll)sqrt(x*tmp);  
            if(ans>=(ll)(1e9)) break;  
            while(ans*ans<x*tmp) ans++;  
            if(ans*ans/tmp==x){  
                flag = 1;  
                break;  
            }  
            tmp*=10;  
        }  
        if(flag) cout<<ans<<endl;  
        else cout<<"-1"<<endl;  
    }  
}
```

**F****G****H****I****签到题**

给定 $n*m$ 大小的棋盘,按五子棋规则构造两人和棋情况

结论很显然

如果行数为偶数,对每一行构造一个连续四个断一个即可,下一行和上一行的情况完全相反

如果行数为奇数,考虑前 $n-1$ 行采取偶数行情况的构造,最后一行两个棋子交替即可

本来第一直觉以为是考察五子棋的日字八卦阵,结果发现我属实想多了(虽然正解是这么做的,但我觉得很没必要)

八卦阵是和棋局面的充分条件,并非必要条件

**AC代码**

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int t;
    cin>>t;
    while(t--){
        int n,m;
        cin>>n>>m;
        int flag = 1;
        for(int i=0;i<(n%2?n-1:n);++i){
            int count = 1;
            for(int j=0;j<m;++j){
                if(flag>0){
                    if(count<=4){
                        printf("x");
                        count++;
                    }
                    else{
                        count = 1;
                        printf("o");
                    }
                }
                else{
                    if(count<=4){
                        printf("o");
                    }
                }
            }
        }
    }
}
```

```
        count++;
    }
    else{
        count = 1;
        printf("x");
    }
}
flag = -flag;
printf("\n");
}
if(n%2){
    for(int j=0;j<m;++j){
        if(j%2==0) printf("x");
        else printf("o");
    }
    printf("\n");
}
}
```

J

K

L

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: [https://wiki.cvbbacm.com/doku.php?id=2023-2024:teams:ikun\\_is\\_coding:2023\\_%E7%89%9B%E5%AE%A2%E6%9A%91%E6%9C%9F%E5%A4%9A%E6%A0%A1%E8%AE%AD%E7%BB%83%E8%90%A5\\_2&rev=1690094114](https://wiki.cvbbacm.com/doku.php?id=2023-2024:teams:ikun_is_coding:2023_%E7%89%9B%E5%AE%A2%E6%9A%91%E6%9C%9F%E5%A4%9A%E6%A0%A1%E8%AE%AD%E7%BB%83%E8%90%A5_2&rev=1690094114)

Last update: 2023/07/23 14:35