2019周赛二标程

A

```
#include<iostream>
using namespace std;
int n,k,a,sum;
int main()
{
    cin>>n>>k;
    for(int i=0;i<n;i++){
        cin>>a;
        sum+=a;
    }
    if(sum>k)cout<<"YES"<<endl;
    else cout<<"NO"<<endl;
    return 0;
}
```

B

```
#include<iostream>
using namespace std;
typedef long long ll;
ll sum,a,b;//注意int只有1e9！开long long才能AC
int main()
{
    while(cin>>a>>b){
        cout<<a*b<<endl;
    }
    return 0;
}
```

C

```
#include <iostream>
#include <string>
using namespace std;
string s;
int main(){
    while(cin>>s){
        if(s=="Sunny"){
            cout<<"Cloudy"<<endl;
        }
        else if(s=="Cloudy"){
            cout<<"Rainy"<<endl;
        }
        else cout<<"Sunny"<<endl;
    }
    return 0;
}
```

D

复杂度$O(N^2 * logN)$

```
#include <iostream>
#include <algorithm>
#include <iomanip>
#include <cstring>
using namespace std;
int n,m,a[6005];
int sum[3005];
int mmin;
int main(){
    while(cin>>n>>m){
        memset(sum,0,sizeof(sum));//memset是一个将数组清零的函数，详细用法请参考百度！
        memset(a,0,sizeof(a));
        mmin=0x3f3f3f3f;
        for(int i=1;i<=n;i++){
            cin>>a[i];
            a[i+n]=a[i];
        }
        for(int i=1;i<=n;i++){
            for(int j=1;j<=m;j++){
                sum[i]+=a[i+j-1];
            }
            if(sum[i]<mmin)mmin=sum[i];
        }
        cout<<mmin<<endl;
    }
}
```

E

方法一 复杂度$O(2 * N * M)$:

```
#include <iostream>
#include <algorithm>
#include <iomanip>
#include <cstring>
using namespace std;
int n,m,a[6005];
int sum[3005];
int mmin;
int main(){
    while(cin>>n>>m){
        //memset(void *s,int ch,size_t n);
        memset(sum,0,sizeof(sum));//memset是一个将数组清零的函数，详细用法请参考百度！
        memset(a,0,sizeof(a));
        mmin=0x3f3f3f3f;
        for(int i=1;i<=n;i++){
            cin>>a[i];
            a[i+n]=a[i];
        }
        for(int i=1;i<=n;i++){
            for(int j=1;j<=m;j++){
                sum[i]+=a[i+j-1];
            }
            if(sum[i]<mmin)mmin=sum[i];
        }
        cout<<mmin<<endl;
    }
}
```

方法二 复杂度$O(2*N)$:

前缀和优化

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef unsigned int UINT;
typedef unsigned long long ull;

int const maxn = 3000 + 10;
ll sum[maxn << 1], a[maxn << 1];
int const INF = 0x3f3f3f3f;

int main(void) {
    int n, m;
    while (cin >> n >> m) {
        for (int i = 1; i <= n; i++) {
            cin >> a[i];
            a[n + i] = a[i];
        }
        for (int i = 1; i <= 2 * n; i++) {
            sum[i] = sum[i - 1] + a[i];
        }
        ll ans = INF;
        for (int i = m; i <= 2 * n; i++) {
            ans = min(ans, sum[i] - sum[i - m]);
        }
        cout << ans << endl;
    }
    return 0;
}
```

F

方法一：

```
#include <iostream>
using namespace std;
int n, m;
string s[1005];
int ans;
int main() {
    cin >> n >> m;
    for (int i = 0; i < n; i++) cin >> s[i];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            if (s[i][j] - '0' != 0) ans++;
    ans *= 2;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (i + 1 < n && s[i][j] > s[i + 1][j]) {
                ans += s[i][j] - s[i + 1][j];
            }
            if (i - 1 >= 0 && s[i][j] > s[i - 1][j]) {
                ans += s[i][j] - s[i - 1][j];
            }
            if (j + 1 < m && s[i][j] > s[i][j + 1]) {
                ans += s[i][j] - s[i][j + 1];
            }
            if (j - 1 >= 0 && s[i][j] > s[i][j - 1]) {
                ans += s[i][j] - s[i][j - 1];
            }
            if (i + 1 == n || i - 1 == -1) {
                ans += s[i][j] - '0';
            }
            if (j - 1 == -1 || j + 1 == m) {
                ans += s[i][j] - '0';
            }
        }
    }
    cout << ans << endl;
    return 0;
}
```

方法二：

```
#include <iostream>
using namespace std;
int n, m;
string ss;
int a[3005][3005];
int mmax, sum;
int main() {
    cin >> n >> m;
    sum = 0;
    for (int i = 1; i <= n; i++) {
        cin >> ss;
        for (int j = 0; j < m; j++) {
            a[i][j + 1] = ss[j] - '0';
            if (a[i][j + 1] > 0) sum++;
        }
    }
    for (int i = 1; i <= m; i++) {
        sum += a[1][i];
        for (int j = n; j > 1; j--) {
            if (a[j][i] > a[j - 1][i]) sum += (a[j][i] - a[j - 1][i]);
        }
    }
    for (int i = 1; i <= n; i++) {
        sum += a[i][1];
        for (int j = 2; j <= m; j++) {
            if (a[i][j] > a[i][j - 1]) sum += (a[i][j] - a[i][j - 1]);
        }
    }
    cout << sum * 2 << endl;
    return 0;
}
```

方法三：

```
#include <cstring>
#include <iostream>
#include <string>
using namespace std;
int book[1005][1005];
int main() {
    int a, b;
    while (cin >> a >> b) {
        memset(book, 0, sizeof(book));
        int sum = 0;
        for (int i = 1; i <= a; i++) {
            string s;
            cin >> s;
            for (int j = 1; j <= b; j++) {
                book[i][j] = s[j - 1] - '0';
                if (s[j - 1] - '0' > 0) {
                    sum += 2;
                }
            }
        }
        for (int i = 1; i <= a; i++) {
            for (int j = 1; j <= b; j++) {
                if (book[i - 1][j] < book[i][j]) {
                    sum += book[i][j] - book[i - 1][j];
                }
                if (book[i][j - 1] < book[i][j]) {
                    sum += book[i][j] - book[i][j - 1];
                }
                if (book[i + 1][j] < book[i][j]) {
                    sum += book[i][j] - book[i + 1][j];
                }
                if (book[i][j + 1] < book[i][j]) {
                    sum += book[i][j] - book[i][j + 1];
                }
            }
        }
        cout << sum << endl;
    }
    return 0;
}
```

方法四：

```
#include <bits/stdc++.h>
using namespace std;
int a[1005][1005][15], b[1005][1005];
long long s;
char ss[1005];
int main() {
    int n, m, nm;
    scanf("%d %d", &n, &m);
    int ma = -1, add = 0;
    for (int i = 1; i <= n; i++) {
        scanf("%s", ss);
        for (int j = 1; j <= m; j++) {
            for (int k = 1; k <= ss[j - 1] - '0'; k++) {
                a[i][j][k] = 1;
            }
            add += ss[j - 1] - '0';
            b[i][j] = ss[j - 1] - '0';
        }
    }
    s = 6 * add;  //先算每个方块加起来的总面积
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= m; j++)
            for (int k = 1; k <= b[i][j]; k++)
                //总结公式 重叠的减去
                s = s - a[i + 1][j][k] - a[i][j + 1][k] - a[i - 1][j][k] -
                    a[i][j - 1][k] - a[i][j][k + 1] - a[i][j][k - 1];

    printf("%lld\n", s);
    return 0;
}
```

G

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    int T;
    while (cin >> T)
    {
        while (T--)
        {
            int r = 0, book[10], n;
            memset(book, 0, sizeof(book));
            for (int i = 1; i <= 7; i++)
            {
                cin >> n;
                book[n] = 1;
                if (n == 0)
                {
                    r = 1;
                }
            }
            for (int i = 1; i <= 7; i++)
            {
                cin >> n;
                book[n] = 1;
            }
            if (r == 0)
            {
                cout << "NO" << endl;
                continue;
            }
            if ((book[1] == 1 || book[2] == 1) && book[4] != 1)
            {
                cout << "NO" << endl;
            }
            else if (book[4] == 1 && book[3] != 1)
            {
                cout << "NO" << endl;
            }
            else
            {
                cout << "YES" << endl;
            }
        }
    }
    return 0;
}
```