

二维数组

人员

韩承睿、徐浩然、牟茗、刘嘉航、辛帅辰、高健桓、李翰如、崔宸赫、崔吉诺、刘祺、方俊喆、刘峰烁、秦显森 到课

作业

noi 1.6 07:有趣的跳跃
noi 1.8 03:计算矩阵边缘元素之和

课堂表现

课上有个别同学听讲不认真容易走神，以后要集中注意力听课。

许多同学课堂上都过了 noi 1.6 07 有趣的跳跃 这个题目，但是如果下节课再写一次可能还是写不出来。

所以同学们课下一定要好好复习这个题目。

课堂内容

1333 兴趣班的排班

利用桶标记法的思想，开一个f数组，其中 $f[i]$ 代表第 i 天有多少名学生一起上课
假设有 3 名学生，3 名学生分别每 3/2/4 天上一次课
则对于第一名学生来说：
第1天上课、第4天上课、第7天上课、第10天上课、第13天上课、第16天上课、...
可以让 $f[1]++$, $f[4]++$, $f[7]++$, $f[10]++$, $f[13]++$, $f[16]++$, ...
对于第二名学生：则是 $f[1]++$, $f[3]++$, $f[5]++$, $f[7]++$, $f[9]++$, $f[11]++$, $f[13]++$, $f[15]++$, $f[17]++$, ...
对于第三名学生：是 $f[1]++$, $f[5]++$, $f[9]++$, $f[13]++$, $f[17]++$, ...
最终 $f[1]=3$, $f[2]=0$, $f[3]=1$, ..., $f[13]=3$
说明第1天有3个人，第2天有0个人，第3天有1个人，..., 第13天有3个人
所以，在第1天之后，在第13天可以有 3 个人同时上课

```

#include <bits/stdc++.h>

using namespace std;

int w[100005];

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; i++) {
        int x; cin >> x;
        for (int j = 1; j <= 100000; j += x) {
            w[j]++;
        }
    }
    for (int i = 2; i <= 100000; i++) {
        if (w[i] == n) {
            cout << i << endl;
            return 0;
        }
    }
    return 0;
}

```

1555 洗牌

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1000 + 5;
int w[maxn], a[maxn];

int main()
{
    int n, k, x; cin >> n >> k >> x;
    for (int i = 1; i <= n; ++i) w[i] = i;

    while (k -- ) {
        for (int pos = 1, i = 1, j = n/2+1; pos <= n; ++pos) {
            if (pos % 2 == 1) {
                a[pos] = w[i]; ++i;
            } else {
                a[pos] = w[j]; ++j;
            }
        }
        for (int i = 1; i <= n; ++i) w[i] = a[i];
    }

    cout << w[x] << endl;
    return 0;
}

```

noi 1.6 07:有趣的跳跃

求两个数a和b的差的绝对值:

```

int t = a-b;
if (t < 0) {
    t = -t;
}

```

一共n个数，则存在 $n-1$ 个差

那么需要判断这 $n-1$ 个差是否正好是 $1, 2, 3, \dots, n-1$

如果存在 差 $=0$ 或者 差 $>=n$ 的情况，则这 $n-1$ 个数一定凑不出 $1, 2, 3, \dots, n-1$
可以直接输出 Not Jolly

a_1 a_2 a_3 a_4 a_5
x y z t

$b[x]++$

$b[y]++$

$b[z]++$

$b[t]++$

判断b数组

$b_1 == 1$

$b_2 == 1$

$b_3 == 1$

$b_4 == 1$

```

#include <iostream>

using namespace std;

int a[3005], b[3005];

int main() {
    int n; cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }

    for (int i = 2; i <= n; i++) {
        int t = a[i] - a[i-1];
        if (t < 0) {
            t = -t;
        }
        if (t==0 || t>n-1) {
            cout << "Not jolly";
            return 0;
        }
        b[t]++;
    }

    // 检查 b[1] ~ b[n-1] 是否全都等于 1
    for (int i = 1; i <= n-1; i++) {
        if (b[i] != 1) {
            cout << "Not jolly";
            return 0;
        }
    }

    cout << "Jolly";
    return 0;
}

```

二维数组

```
// 定义、输入、输出 二维数组 的方式
#include <iostream>

using namespace std;

int a[3][4]; // 3行4列的二维数组
// 只有 第0行、第1行、第2行，没有第3行
// 只有 第0列、第1列、第2列、第3列，没有第4列

int main() {
    // 输入方式
    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            cin >> a[i][j];
        }
    }

    // 输出方式
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

noi 1.8 01 矩阵交换行

```

#include <iostream>

using namespace std;

int a[10][10];

int main() {
    for (int i = 1; i <= 5; i++) {
        for (int j = 1; j <= 5; j++) {
            cin >> a[i][j];
        }
    }
    int l, r;
    cin >> l >> r;
    for (int i = 1; i <= 5; i++) {
        // 交换 a[l][i] 和 a[r][i]
        int t = a[l][i];
        a[l][i] = a[r][i];
        a[r][i] = t;
    }
    for (int i = 1; i <= 5; i++) {
        for (int j = 1; j <= 5; j++) {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}

```

交换a和b两个变量的方式:

方式一:

```

int t = a;
a = b;
b = t;

```

方式二:

```

#include <algorithm>
swap(a, b);

```