

1 爆破

爆破

时间限制： 3000MS

内存限制： 589824KB

题目描述：

小明当起了矿场的爆破工程师。小明工作的矿场可以看作一个 $N \times N$ 的二维网格，小明可以将一个炸弹放置到矿场的任意位置，若干时间后炸弹会爆炸，将炸弹所在处及相邻位置炸开，让工人们可以轻松获得那些位置的矿物。现在给出这个矿场的矿物数量分布，请你帮小明找出一个最佳位置，使得炸完后能获得的矿物数量最大。你只需要告诉小明这个最大数量即可。在第 i 行 j 列的矿物数量为 $a[i][j]$ (当 $1 \leq i, j \leq N$)，否则为 0 (即超出矿场边界时不会获得任何矿物)。若放置炸弹在 (i, j) ，爆炸范围为 $(i, j), (i+1, j), (i, j+1), (i-1, j), (i, j-1)$ 。注意小明只能把炸弹放置到矿场内，但可能爆炸范围超出边界。

输入描述：

第一行一个整数 N ，表示小明工作的矿场大小。

接下来 N 行，每行 N 个整数。接下来的第 i 行开 $a[i][1]a[i][2] \cdots a[i][N]$ 。表示这矿场内所有位置的矿物数量。

对于 100% 的数据， $1 \leq N \leq 800, 0 \leq a[i][j] \leq 100000$

输出描述：

输出一个整数表示小明安放一个炸弹，其爆炸范围内矿物数量之和的最大值。

样例输入 1：

```
5
1 2 3 3 4
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0
9 0 9 9 9
```

样例输出 1：

```
27
```

思路分析：

实现代码：

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm> // 用于 std::max
4
5 using namespace std;
6
7 int main() {
8     int N;
9     cin >> N;
10    vector<vector<int>> a(N, vector<int>(N)); // 二维数组的定义
11    for (int i = 0; i < N; i++) {
12        for (int j = 0; j < N; j++) {
13            cin >> a[i][j]; // 输入数组的元素
14        }
15    }
16
17    int maxMinerals = 0;
18    for (int i = 0; i < N; i++) {
19        for (int j = 0; j < N; j++) {
```

```
20     int currentMinerals = a[i][j]; // 当前炸弹位置的矿物数量
21     if (i > 0) {
22         currentMinerals += a[i - 1][j]; // 上方位置
23     }
24     if (i < N - 1) {
25         currentMinerals += a[i + 1][j]; // 下方位置
26     }
27     if (j > 0) {
28         currentMinerals += a[i][j - 1]; // 左侧位置
29     }
30     if (j < N - 1) {
31         currentMinerals += a[i][j + 1]; // 右侧位置
32     }
33     maxMinerals = max(maxMinerals, currentMinerals); // 更新最大矿物数量
34 }
35 }
36
37 cout << maxMinerals << endl; // 输出结果
38 return 0;
39 }
```

2 切分

切分

时间限制：3000MS

内存限制：589824KB

题目描述：

小明今年儿童节获得了妈妈送的一个可爱的字符串 s 。而这个字符串也是一个数字串， s 仅由数字 '0', '1', ..., '9' 构成。小明在课上记住了如下定义：记字符串 s 的长度为 $|s|$ ，其中第 i 位字符为 $s[i]$ ，一个子串由相邻不间断的字符构成，字符 $s[i], s[i+1], \dots, s[j]$ 构成其中一个子串，表示为 $s[i, j]$ ($1 \leq i \leq j \leq |s|$)。例如，'abcdefg' 有子串 'abc' 'bcde'，但 'abd' 不是其子串。小明想象着如同切蛋糕一样切分字符串。小明会确定两个端点 i, j ($1 \leq i \leq j \leq |s|$)，将子串 $s[i, j]$ 取出。小明对数字有着独特的品味，特别的，他希望切出来的这个子串能被 4 整除。小明想知道他有多少种选择端点的方案，使得他的品味得到满足。注意切出的子串允许出现前导 0，即 "0123" 数值上与 "123" 相同。两种选择端点的方案不同当前仅当在两个方案中左端点或者右端点不相同，或者两个端点均是不同的。

输入描述：

第一行一个字符串 S ，表示小明的字符串。

对于 100% 的数据， $1 \leq |S| \leq 50000$ ，其中 S 中仅包含数字。

输出描述：

输出一行一个整数表示答案。

样例输入 1：

```
104
```

样例输出 1：

```
4
```

思路分析：

实现代码：

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main() {
7     string s;
8     cin >> s;
9     int n = s.length();
10    int count = 0;
11
12    for (int i = 0; i < n; i++) {
13        // 单个字符组成的子串
14        int singleDigit = s[i] - '0';
15        if (singleDigit % 4 == 0) {
16            count++;
17        }
18
19        // 两个字符组成的子串
20        if (i > 0) {
21            int twoDigitNumber = (s[i - 1] - '0') * 10 + (s[i] - '0');
22            if (twoDigitNumber % 4 == 0) {
23                count += i; // 包含前导0的子串总数
24            }
25        }
26    }
27}
```

```
25     }  
26 }  
27  
28 cout << count << endl;  
29  
30 return 0;  
31 }
```