

题目描述：

有若干个文件，使用刻录光盘的方式进行备份，假设每张光盘的容量是 $500MB$ ，求使用光盘最少的文件分布方式

所有文件的大小都是整数的 MB ，且不超过 $500MB$ ；文件不能分割、分卷打包

输入描述：

一组文件大小的数据

输出描述：

使用光盘的数量

补充说明：

不用考虑输入数据不合法的情况；假设最多 100 个输入文件。

示例 1

输入：

100, 500, 300, 200, 400

输出：

3

说明：

$(100, 400), (200, 300), (500)$ 3 张光盘即可。

输入和输出内容都不含空格。

示例 2

输入：

1, 100, 200, 300

输出：

2

说明：

```
#include <iostream>
```

```
#include <string>
```

```

#include <vector>
#include <algorithm>
using namespace std;

void dealInput(vector<int> &vec, string str) {
    str.push_back(',');
    int index = 0;
    while (index < str.size()) {
        if (str[index] == ',') {
            index++;
        }
        else {
            string temp;
            while (str[index] != ',') {
                temp.push_back(str[index]);
                index++;
            }
            vec.push_back(stoi(temp));
        }
    }
}

int solution(vector<int> &vec) {
    int result = 0;
    sort(vec.begin(), vec.end());
    int left = 0, right = vec.size() - 1;
    while (right >= left) {
        if (vec[right] + vec[left] <= 500) {
            /*
            result++;
            right--;
            left++;
            */

            int curSize = vec[right] + vec[left];
            right--;
            left++;
            result++;
            while (true) {
                int nextSize = curSize + vec[left];
                if (nextSize <= 500) {
                    curSize = nextSize;
                    left++;
                }
            }
        }
    }
}

```

```
        else {
            break;
        }
    }
}
else {
    result++;
    right--;
}
}
return result;
}
```

```
int main() {
    string str;
    getline(cin, str);
    vector<int> vec;
    dealInput(vec, str);
    cout << solution(vec) << endl;
}
// 64 位输出请用 printf("%lld")
```