

题目描述：

双十一众多商品进行打折销售，小明想购买自己心仪的一些物品，但由于受购买资金限制，所以他决定从众多心仪商品中购买三件，而且想尽可能的花完资金，现在请你设计一个程序帮助小明计算尽可能花费的最大资金数额。

输入描述：

输入第一行为一维整型数组  $M$ ，数组长度小于  $100$ ，数组元素记录单个商品的价格，单个商品价格小于  $1000$ 。

输入第二行为购买资金的额度  $R$ ， $R$  小于  $100000$ 。

输出描述：

输出为满足上述条件的最大花费额度。

注意：如果不存在满足上述条件的商品，请返回  $-1$ 。

补充说明：

输入格式是正确的，无需考虑格式错误的情况。

示例 1

输入：

23,26,36,27

78

输出：

76

说明：

金额 23、26 和 27 相加得到 76，而且最接近且小于输入金额 78

示例 2

输入：

23,30,40

26

输出：

-1

说明：

因为输入的商品，无法组合出来满足三件之和小于 26.故返回 -1

```
#include <algorithm>
#include <iostream>
#include<vector>
#include<string>
#include<sstream>
using namespace std;

int main() {
    int a;
    vector<int> nums;
    string str;
    getline(cin,str);
    stringstream ss(str);
    string token;

    while (getline(ss,token',')) { // 注意 while 处理多个 case
        int number = stoi(token);
        nums.push_back(number);
    }
    cin >> a;

    int maxsum = -1;

    for(int i = 0; i < nums.size(); ++i){
        for(int j = i+1;j<nums.size();++j){
            for(int k = j+1;k<nums.size();++k){
                if(nums[i]+ nums[j] + nums[k] <= a){
                    int sum = nums[i] + nums[j] + nums[k];
                    maxsum = max(maxsum,sum);
                }
            }
        }
    }

    cout << maxsum << endl;
    return 0;
}

// 64 位输出请用 printf("%lld")
```