

题目描述：小明玩一个游戏。系统发1+n张牌，每张牌上有一个整数。第一张给小明，后n张按照发牌顺序排成连续的一行。需要小明判断，后n张牌中，是否存在连续的若干张牌，其和可以整除小明手中牌上的数字。

输入描述：输入数据有多组，每组输入数据有两行，输入到文件结尾结束。
第一行有两个整数n和m，空格隔开。m代表发给小明牌上的数字。
第二行有n个数，代表后续发的n张牌上的数字，以空格隔开。

输出描述：对每组输入，如果存在满足条件的连续若干张牌，则输出1；否则，输出0

补充说明：1 <= n <= 1000
1 <= 牌上的整数 <= 400000
输入的组数，不多于1000
用例确保输入都正确，不需要考虑非法情况。

示例1

输入：6 7
2 12 6 3 5 5
10 11
1 1 1 1 1 1 1 1 1 1

输出：1
0

说明：两组输入。第一组小明牌的数字为7，再发了6张牌。第1、2两张牌数字和为14，可以整除7，输出1。第二组小明牌的数字为11，再发了10张牌，这10张牌数字和为10，无法整除11，输出0。

```
#include <iostream>
#include <vector>
using namespace std;

bool isAccept(vector<int>& vn, int num);
int main() {
    int n;
    int num;
    while(cin >> n >> num){
        vector<int> vn(n, 0);
        for(auto &e: vn){
            cin >> e;
            e = e%num;
        }
        if (isAccept(vn, num)){
            std::cout << "1" << std::endl;
        }else{
            std::cout << "0" << std::endl;
        }
    }
}

bool isAccept(vector<int>& vn, int num){
    for(int i=0;i<vn.size();i++){
        int sum=0;
        for(int j=i;j<vn.size();j++){
            sum += vn[j];
            if (sum >= num){
                sum -= num;
            }
            if (sum == 0){
```

```
        return true;
    }
}
return false;
}
// 64 位输出请用 printf("%lld")
```