

数字游戏题目描述：

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：

$1 \leq n \leq 1000$

$1 \leq \text{牌上的整数} \leq 400000$

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 <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```

struct data{
    int * nums;
    int numsSize;
    int target;
};

void calc(struct data * arr, int arrLen){
    for(int i = 0; i < arrLen; ++i){
        int tmp[1000] = {0};
        int tmpLen = 1;
        int sum = 0;
        int res = 0;
        for(int j = 0; j < arr[i].numsSize; ++j){
            sum += arr[i].nums[j];
            int val = sum % arr[i].target;
            int flag = 0;
            for (int k = 0; k < tmpLen; ++k) {
                if(val == tmp[k]){
                    flag = 1;
                    break;
                }
            }
            if(flag == 1){
                res = 1;
                break;
            }
            else{
                tmp[tmpLen++] = val;
            }
        }
        printf("%d\n", res);
    }
}

int main(){
    int n = 0;
    struct data arr[1000] = {0};
    int arrLen = 0;
    while(scanf("%d", &n) != EOF){
        arr[arrLen].numsSize = n;
        arr[arrLen].nums = (int *)malloc(sizeof(int) * n);
        scanf("%d", &arr[arrLen].target);
        for (int i = 0; i < n; ++i) {
            scanf("%d", &arr[arrLen].nums[i]);

```

```
        }  
        arrLen++;  
    }  
    calc(arr, arrLen);  
    return 0;  
}
```