

23、数据分类，考点 or 实现——逻辑分析

题目描述

对一个数据a进行分类，分类方法为：
 此数据a（四个字节大小）的四个字节相加对一个给定的值b取模，如果得到的结果小于一个给定的值c，则数据a为有效类型，其类型为取模的值；如果得到的结果大于或者等于c，则数据a为无效类型。
 比如一个数据a=0x01010101，b=3，按照分类方法计算 (0x01+0x01+0x01+0x01) %3=1，
 所以如果c=2，则此a为有效类型，其类型为1，如果c=1，则此a为无效类型；
 又比如一个数据a=0x01010103，b=3，按照分类方法计算 (0x01+0x01+0x01+0x03) %3=0，
 所以如果c=2，则此a为有效类型，其类型为0，如果c=0，则此a为无效类型。
 输入12个数据，第一个数据为c，第二个数据为b，剩余10个数据为需要分类的数据，
 请找到有效类型中包含数据最多的类型，并输出该类型含有多少个数据。

输入描述

输入12个数据，用空格分隔，第一个数据为c，第二个数据为b，剩余10个数据为需要分类的数据。

输出描述

输出最多数据的有效类型有多少个数据。

用例

输入	3 4 256 257 258 259 260 261 262 263 264 265
输出	3
说明	10个数据4个字节相加后的结果分别为1 2 3 4 5 6 7 8 9 10， 故对4取模的结果为1 2 3 0 1 2 3 0 1 2，c为3，所以0 1 2都是有效类型，类型为1和2的有3个数据，类型为0的只有2个数据，故输出3。

输入	1 4 256 257 258 259 260 261 262 263 264 265
输出	2
说明	10个数据4个字节相加后的结果分别为1 2 3 4 5 6 7 8 9 10， 故对4取模的结果为1 2 3 0 1 2 3 0 1 2，c为1，所以只有0是有效类型，类型为0的有2个数据，故输出2。

题目解析

逻辑题，按照题目意思写过程即可。

JavaScript算法源码

```
1  /* JavaScript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  rl.on("line", (line) => {
10    let arr = line.split(" ");
11
12    const c = arr.shift() - 0;
13    const b = arr.shift() - 0;
14
15    console.log(getResult(arr, c, b));
16  });
17
18  function getResult(arr, c, b) {
19    const count = {};
20
21    arr
22      .map((a) => {
23        let str = Number(a).toString(16);
24
25        if (str.length % 2 !== 0) {
26          str = "0" + str;
27        }
28      })
```

```
28
29     let sum = 0;
30     for (let i = 0; i < str.length - 1; i += 2) {
31         sum += parseInt(str.slice(i, i + 2), 16);
32     }
33
34     const type = sum % b;
35
36     if (type < c) {
37         return type;
38     } else {
39         return -1;
40     }
41 })
42 .forEach((type) => {
43     if (type !== -1) {
44         count[type] ? count[type]++ : (count[type] = 1);
45     }
46 });
47
48 return Object.values(count).sort((a, b) => b - a)[0];
49 }
```

Java算法源码

```

27
28         if (str.length() % 2 != 0) {
29             str = "0" + str;
30         }
31
32         int sum = 0;
33         for (int i = 0; i < str.length() - 1; i += 2) {
34             sum += Integer.parseInt(str.substring(i, i + 2), 16);
35         }
36
37         int t = sum % b;
38         if (t < c) {
39             return t;
40         } else {
41             return -1;
42         }
43     })
44     .forEach(
45         t -> {
46             if (t != -1) {
47                 count.put(t, count.getOrDefault(t, 0) + 1);
48             }
49         });
50
51     return count.values().stream().max((x, y) -> x - y).orElse(0);
52 }
53 }

```

Python算法源码

```
1  # 输入获取
2  tmp = list(map(int, input().split()))
3  c = tmp[0]
4  b = tmp[1]
5  arr = tmp[2:]
6
7
8  # 判断是否有效
9  def classify(a):
10     s = hex(a)[2:] # hex方法会将十进制转为0x开头的16进制数，这里去除0x前缀
11
12     if len(s) % 2 != 0:
13         s = "0" + s # 补足0
14
15     sumV = 0
16     for i in range(0, len(s) - 1, 2):
17         sumV += int(s[i:i + 2], 16)
18
19     t = sumV % b
20
21     if t < c:
22         return t
23     else:
24         return -1
25
26
27 # 算法入口
28 def getResult():
```

```
28 def getResult():
29     count = {}
30
31     for t in map(classify, arr):
32         if t != -1:
33             if count.get(t) is None:
34                 count[t] = 1
35             else:
36                 count[t] += 1
37
38     return max(count.values())
39
40
41 # 算法调用
42 print(getResult())
```