

# 2023年NOC复赛–Python小高组

## 1. 计算年终奖

时间限制：1s

内存限制：128MB

(注：input()括号中不允许添加任何提示语)

某公司的员工年终奖计算规则如下，假设该公司的年度利润为M（单位：万元），若 $M \leq 10$ ，则员工年终奖为其本人当月工资的3倍；若 $10 < M \leq 20$ ，则员工年终奖为其本人当月工资的5倍；若 $M > 20$ ，则员工年终奖为其本人当月工资的8倍。

请补全下面程序，使程序实现如下功能：

- (1) 程序运行后，依次输入两个正整数M、m，表示公司的年度利润和员工当月工资；
- (2) 根据规则计算并输出员工的年终奖金额。

```
1 M = int(input())
2 m = int(input())
3
4 if M <= 10:
5     -----
6     -----
7     m *= 5
8     -----
9     -----
10
11 print(m) # 输出员工年终奖金额
```

参考代码：

```
1 M = int(input())
2 m = int(input())
3
4 if M <= 10:
5     m *= 3
6 elif M <= 20:
7     m *= 5
8 else:
9     m *= 8
10
11 print(m) # 输出员工年终奖金额
```

## 2. 最大公约数

时间限制：1s

内存限制：128MB

(注：input()括号中不允许添加任何提示语)

欧几里得算法又称辗转相除法，定义是：两个整数的最大公约数等于其中较小的那个数和两数相除余数的最大公约数。因此只要对除数和余数反复做除法运算，当余数为0时，当前算式的除数就是最初两个数的最大公约数。

例如：

48除以18，余数为12；

接下来，18除以12，余数为6；

接下来，12除以6，余数为0。

所以，6就是48和18的最大公约数。

请补全下面程序，使程序实现如下功能：

- (1) 程序运行后，依次输入两个正整数a、b；
- (2) 使用欧几里得算法计算a和b的最大公约数，并输出结果。

```
1 a = int(input())
2 b = int(input())
3
4 def gcd(a, b):
5     if a < b:
6         a, b = b, a
7     while _____:
8         r = _____
9         a = b
10        b = r
11    return _____
12
13 print(_____) # 输出a和b的最大公约数
```

参考代码：

方法一：

```
1 a = int(input())
2 b = int(input())
3
4 def gcd(a, b):
5     if a < b:
6         a, b = b, a
7     while a % b != 0:
8         r = a % b
9         a = b
10        b = r
11    return b
12 print(gcd(a, b)) # 输出a和b的最大公约数
```

方法二：

```
1 a = int(input())
2 b = int(input())
3
4 def gcd(a, b):
5     if a < b:
6         a, b = b, a
7     while b != 0:
8         r = a % b
9         a = b
10        b = r
11    return a
12 print(gcd(a, b)) # 输出a和b的最大公约数
```

### 3.最少纸币数

时间限制：1s

内存限制：128MB

(注：input()括号中不允许添加任何提示语)

小新使用现金购买一些商品，商品总价值 $w$  ( $1 \leq w \leq 10000$ )，他有100元、50元、10元、5元和1元的纸币，每种纸币张数足够多。

请编写一段程序，计算并输出要买到这些商品，最少需要的纸币张数。

输入描述：

输入1个正整数 $w$  ( $1 \leq w \leq 10000$ )，表示要购买的商品总价值

输出描述：

第一行输出1个正整数，表示最少需要多少张纸币

第二行依次输出5个正整数，数字之间以空格隔开，分别表示100元至1元纸币所需张数

输入样例：

38

输出样例：

7

0 0 3 1 3

参考代码：

```
1 s = int(input())
2 ls = [0, 0, 0, 0, 0]
3 ls[0] = s // 100
4 ls[1] = s % 100 // 50
5 ls[2] = s % 50 // 10
6 ls[3] = s % 10 // 5
```

```
7 ls[4] = s % 5
8 print(sum(ls))
9 for i in ls:
10     print(i, end = ' ')
11
```

#### 4. 单词变复数

时间限制：1s

内存限制：128MB

(注：input()括号中不允许添加任何提示语)

英语单词在变成复数形式时，有以下几种常见情况（不完全）：

- (1) 常规情况下结尾直接加s；
- (2) 以s、sh、ch、x结尾的单词，加es；
- (3) 以辅音字母加上y结尾的单词，去掉y加ies；

(注：英文字母中，除了a、e、i、o、u这5个元音字母外，其他都是辅音字母)

请编写一段程序，输入一个英文单词（单数，仅包含小写字母），并按以上规则变成复数形式。

输入描述：

输入一个英文单词（单数，仅包含小写字母）

输出描述：

输出这个单词按以上规则变成的复数形式

输入样例：

teacher

输出样例：

teachers

参考代码：

```
1 lsw = input().split()
2 def plural(word):
3     if word[-1] in ['s', 'x'] or word[-2:] in ['sh', 'ch']:
4         return word + 'es'
5     elif word[-1] == 'y' and word[-2] not in ['a', 'e', 'i', 'o', 'u']:
6         return word[:-1] + 'ies'
7     else:
8         return word + 's'
9 for w in lsw:
10     print(plural(w), end=' ')
```

#### 5. 毕业旅行

时间限制：1s

内存限制：128MB

(注：input()括号中不允许添加任何提示语)

小王要从一个城市出发前往另一个城市进行一场毕业旅行，在此期间会经过 $m$  ( $1 \leq m \leq 100$ ) 个中转城市。小王给每个城市按顺序设置了一个对他们的喜爱度，分别用  $a_1, a_2, \dots, a_m$  ( $1 \leq a_i \leq 100$ ) 表示，不同的城市可能拥有相同的喜爱度。现在，小王需要选择 $k$  ( $1 \leq k \leq m$ ) 个编号连在一起的中转城市作为重点游览的城市，并希望重点游览的城市喜爱度之和最大。

请编写一段程序，计算并输出小王重点游览的城市喜爱度之和可以获得的最大值。

输入描述：

第一行依次输入两个正整数 $m$ 和 $k$ ，依次表示会经过的中转城市数量和重点游览城市数量，正整数之间以空格隔开

第二行依次输入 $m$ 个正整数，依次表示每个城市的喜爱度  $a_i$ ，正整数之间以空格隔开

输出描述：

输出一个整数，表示小王重点游览的城市喜爱度之和可以获得的最大值

输入样例：

4 2

10 20 15 30

输出样例：

45

参考代码：

```
1 m, k = [int(i) for i in input().split()]
2 cities = [int(i) for i in input().split()]
3 maxn = 0
4 for i in range(k, m + 1):
5     sum_v = sum(cities[i - k: i])
6     maxn = max(maxn, sum_v)
7 print(maxn)
```

## 6. 报数游戏

时间限制：1s

内存限制：128MB

(注：input()括号中不允许添加任何提示语)

有 $n$  ( $6 \leq n \leq 100$ ) 个小伙伴围成一圈做游戏，分别编号为 $1 \sim n$ 。游戏规则如下：

从第一个人开始依次说出N、O、C三个字母，第一个人说N，第二个人说O，第三个人说C，第四个人再说N，以此类推。如果一个人说到了两次O，那么他就被淘汰，下一个人从C开始继续往后说，直到最终剩下五个小伙伴，这五人即获得游戏的胜利。

例如：有六个小伙伴做游戏，那么1号说N，2号说O，3号说C，4号说N，5号说O，6号说C。

接下来再从1号继续，1号说N，2号说O，此时2号已经是第二次说O，那么2号淘汰，剩下的5个小伙伴的座位号是：1、3、4、5、6。

请编写一段程序，输入参与游戏的人数 $n$  ( $6 \leq n \leq 100$ )，输出最终获得游戏胜利的五位小伙伴的座位编号。

**输入描述：**

输入一个正整数 $n$ ，代表一共有 $n$ 个小伙伴参加游戏 ( $6 \leq n \leq 100$ )

**输出描述：**

由小到大输出获得游戏胜利的五位小伙伴的座位编号，编号之间以空格隔开

**输入样例：**

6

**输出样例：**

1 3 4 5 6

**参考代码：**

```
1 n = int(input())
2
3 idx = [i + 1 for i in range(n)]
4 hp = [2] * n
5 num = 0
6 p = -1
7
8 while len(hp) > 5:
9     p += 1
10    p %= len(hp)
11    num += 1
12    if num % 3 == 2:
13        hp[p] -= 1
14        if hp[p] == 0:
15            hp.pop(p)
16            idx.pop(p)
17            p -= 1
18 for i in idx:
19     print(i, end = ' ')
```

## 7. 两个背包

时间限制：1s

内存限制：128MB

(注：input()括号中不允许添加任何提示语)

现有两个背包和 $n$  ( $2 \leq n \leq 500$ ) 个物品， $n$ 个物品的重量由输入确定，并且均为正整数。

请编写一段程序，判断能否将这n个物品分到两个背包中，使得两个背包中的物品重量相等。如果可以则输出True，不可以则输出False。

**输入描述:**

一行输入n ( $2 \leq n \leq 500$ ) 个正整数，正整数均不超过100，中间以空格隔开，代表每个物品的重量

**输出描述:**

输出判断结果， True或者False

**输入样例:**

11 5 1 5

**输出样例:**

True

**参考代码:**

```
1 nums = [int(i) for i in input().split()]
2 total = sum(nums)
3 if total % 2 != 0:
4     print(False)
5 else:
6     target = total // 2
7     n = len(nums)
8     dp = [False] * (target + 1)
9     dp[0] = True
10    for i in range(n):
11        for j in range(target, nums[i] - 1, -1):
12            dp[j] = dp[j] or dp[j - nums[i]]
13    print(dp[target])
```