

57. 某市举办科技竞赛，已有预赛成绩，计划从  $n$  所学校选拔  $tot$  名选手参加决赛。选拔规则是：各校排名前  $st$  名的选手直接入选；各校排名  $st+1$  至  $ed$  名的选手进入备选池，剩余名额按预赛成绩由高到低从备选池中挑选，成绩相同的选手一并入选，一旦人数达到或超过  $tot$ ，选拔结束。现给定所有选手的预赛成绩，每名选手的数据包括学校编号  $[0 \sim (n-1)]$ 、选手编号、成绩，计算各选手的校内排名。将所有学校的数据合并到一起并按成绩由高到低排序，生成汇总表，如图所示。再按上述规则选拔，按汇总表的顺序输出入选选手编号。

(1) 如下图所示，若  $n, tot, st, ed$  的值依次为 3, 8, 1, 4，则实际入选决赛的选手数是\_\_\_\_\_。

学校编号	1	1	1	0	0	1	1	0	0	2	2
选手编号	1052	1073	1049	45	75	1079	1029	6	60	2028	2022
成绩	98	93	92	91	91	90	90	90	86	86	85
校内排名	1	2	3	1	1	4	4	3	4	1	2

(2) 校内排名的计算方法是：若选手所在学校有  $m$  个选手成绩高于该选手，则该选手的校内排名为  $m+1$ 。实现校内排名的函数如下所示。

```
def ranking(sdata):
    a = [0] * 101 # 满分 100 分，统计各分数人数
    b = [0] * 101 # 存储各分数对应的排名
    for i in range(len(sdata)):
        score = sdata[i][2]
        a[score] += 1
    mc = 1
    for i in range(100, -1, -1):
        if a[i] != 0:
            b[i], mc = mc, mc + a[i]
    for i in range(len(sdata)):
        sdata[i][3] = b[sdata[i][2]]
```

①若将方框处代码改为语句： $b[i], mc = mc, mc+a[i]$ ，则排名结果将\_\_\_\_\_（填字母：A. 不变/B. 改变）

②若将画线处代码改为： $range(len(sdata)-1, -1, -1)$ ，则排名结果将\_\_\_\_\_（填字母 A. 不变 /B. 改变）

(3) 实现选拔的部分 Python 程序如下所示，请在画线处填入合适的代码。

```
def proc(data, tot, st, ed):
    hs=[-1, -1]
    ts=[-1, -1]
    cnt = 0
    for i in range(len(data)):
        if data[i][3] <= ed:
            k = 1
            ① _____
            cnt += 1
            k = 0
            if hs[k] == -1:
                hs[k] = i
            else:
                ② _____
            ts[k] = i
    p, q = hs[0], hs[1]
```

```

while cnt < tot and q != -1:
    cj = data[q][2]
    while q != -1 and cj == data[q][2]:
        while p != -1 and p < q:
            pre = p
            p = data[p][4]
        cnt += 1
        tmp = data[q][4]
        data[q][4] = p
        ③
        pre = q
        q = tmp
p = hs[0]
while p != -1: # 输出入选选手编号
    print(data[p][1], end='')
    p = data[p][4]
data = []
# 读取 n, tot, st, ed 的值, 代码略
for i in range(n):
    ''' 读取 i 学校的选手成绩数据, 存入 sdata, 每个元素形如 [2, '2050', 84, 5, -1], 前四项依次为
    学校编号、选手编号、成绩、校内名次, 代码略'''
    ranking(sdata)
# 将 sdata 合并到 data, 代码略
# 将 data 中数据按照成绩由高到低排序, 代码略
proc(data, tot, st, ed)

```

58. 某数据序列 data 中的元素均为小于 127 的正整数。现要对 data 进行加密处理，过程分为“变换”和“重排”两步。

“变换”处理方法是使用指定的 n 组序列  $R_0, R_1, \dots, R_{n-1}$  依次对 data 进行变换得到“变换后序列”。利用  $R_i$  对 data 进行变换的过程是：在 data 中查找所有与  $R_i$  相同的子序列，将找到的每个子序列中的元素值加上  $R_i$  的长度值  $L_i$ ，并在各子序列前插入一个标记元素（值为  $127+L_i$ ），这些子序列及标记元素不再参与后续的变换。

对“变换后序列”进行“重排”的方法为：先以 m (分组长度,  $2 \leq m \leq 6$ ) 个元素为 1 组，将 data 中的元素从前往后分成若干组，不足 m 个元素的组末尾用 0 补足，再将各组元素进行组内逆序排列，得到“重排后序列”，即为密文。

例如用于变换的两组序列为 [5, 1]、[3, 8, 7]，m 为 4，对“原始序列”进行变换与重排的结果如下图所示。

原始序列	3	5	1	6	3	8	7	5	1	8	7					
变换后序列	3	129	7	3	6	130	6	11	10	129	7	3	8	7		
重排后序列	3	7	129	3	11	6	130	6	3	7	129	10	0	0	7	8

现要对密文进行解密，过程是：先将“重排后序列”恢复为“变换后序列”，再将“变换后序列”恢复为“原始序列”。请回答下列问题。

- 若变换后序列为 [4, 129, 10, 11, 14, 129, 7, 11, 11, 130, 12, 13, 11, 12, 129, 7, 11, 3, 10, 130, 12, 13, 11, 5]，则用于变换处理的序列组数为\_\_\_\_\_。
- 将“重排后序列”恢复为“变换后序列”的函数如下所示，代码有误，应改正为\_\_\_\_\_

