

2024 厦门市小学生 C++语言复赛试题

学校：_____ 姓名：_____ 成绩：_____

题目名称	纸牌游戏	战线巡逻	乌龟对对碰	有趣子序列
题目类型	传统型	传统型	传统型	传统型
可执行文件名	card	patrol	turtle	sub
源程序文件名	card.cpp	patrol.cpp	turtle.cpp	sub.cpp
输入文件名	card.in	patrol.in	turtle.in	sub.in
输出文件名	card.out	patrol.out	turtle.out	sub.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512MiB	512MiB	512MiB	512MiB
测试点数目	10	10	10	10
测试点是否等分	是	是	是	是
题目分值	100 分	100 分	100 分	100 分
编译选项	-O2 -std=c++14			

注意事项：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考《考试须知》的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

纸牌游戏 (card)

【题目描述】

A 和 B 手中只有一种花色的纸牌，其中的点数分别为 1,2,3,4,5,6,7,8,9,10,11,12,13。他们的大小规则如下所示：

$$2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < 11 < 12 < 13 < 1$$

A 和 B 各出手一次，一张牌定胜负。若 A 胜，则输出 Awesome，若 B 胜，则输出 Brilliant，若平局，则输出 Draw。

【输入格式】

两个整数，分别代表 A 和 B 出牌的点数

【输出格式】

若 A 胜，则输出 Awesome；若 B 胜，则输出 Brilliant；若平局，则输出 Draw。

【输入输出数据样例 1】

card.in	card.out
2 9	Brilliant

【样例解释 1】

2<9，所以 B 赢，输出 Brilliant。

【输入输出数据样例 2】

card.in	card.out
1 7	Awesome

【样例解释 2】

1>7，所以 A 赢，输出 Awesome。

【数据范围】

对于所有的测试点，恒满足： $1 \leq A \leq 13$ ， $1 \leq B \leq 13$

战线巡逻 (patrol)

【题目描述】

在一条战线中,有 n 个需要巡逻的点,为了完成巡逻任务,指挥部计划指派 k 个哨兵,每个哨兵可以自由选择起始位置 i ,不消耗体力。但哨兵每移动一个单位距离(从 i 到 $i+1$ 或 $i-1$),则消耗 1 点体力。

指挥部的目标是将 k 个哨兵合理部署到战线上,使得:

- 所有需要巡逻的点都由至少一名哨兵巡逻过。
- 哨兵总体力消耗的最小。

请你设计一个合理的方案,计算最小的体力消耗,并输出结果。

【输入格式】

第一行,两个整数,分别是 k 和 n ;

第二行, n 个整数,战线中必须巡逻的坐标 a_i 。

【输出格式】

输出消耗的最少体力。

【输入输出数据样例 1】

patrol.in	patrol.out
3 4 -10 -1 1 14	2

【样例解释 1】

哨兵 1 初始点位即为-10,接下来无需移动,消耗为 0。

哨兵 2 初始点位为-1,接下来需向右移动 2 个位置到点位 1,消耗为 2。

哨兵 3 初始点位即为 14,接下来无需移动,消耗为 0。

综上,总消耗为 $0+2+0=2$ 。

【输入输出数据样例 2】

patrol.in	patrol.out
4 9 -11 -3 0 9 -100 2 17 20 1	15

【样例解释 2】

哨兵 1 初始点位即为-100,接下来无需移动,消耗为 0。

哨兵 2 初始点位为-11,接下来无需移动,消耗为 0。

哨兵 3 初始点位即为-3,接下来需向右移动 3 个位置至点位 0,消耗为 3;接下来需向右移

动 1 个位置至点位 1，消耗为 1；接下来需向右移动 1 个位置至点位 2，消耗为 1；接下来需向右移动 7 个位置至点位 9，消耗为 7；共计消耗 $3+1+1+7 = 12$
 哨兵 4 初始点位即为 17，接下来需向右移动 3 个位置，消耗为 3。
 综上，总消耗为 $0+0+12+3 = 15$ 。

【输入输出数据样例 3】

patrol.in	patrol.out
5 3 -1000 100 200	0

【样例解释 3】

根据题意，哨兵巡逻可以做到不消耗体力。

【数据范围】

对于所有测试数据有： $-10^5 \leq a_i \leq 10^5$ ， $1 \leq k \leq 10^5$ ， $1 \leq n \leq 10^5$

测试点	特殊性质 A	k	n
1, 2	否	$k=1$	$\leq 10^5$
3, 4	是	$\leq 10^5$	$\leq 10^5$
5, 6, 7	否	$\leq 10^3$	$\leq 10^2$
8, 9, 10	否	$\leq 10^5$	$\leq 10^5$

特殊性质A：保证 $k \geq n$ 恒成立

乌龟对对碰 (turtle)

【题目描述】

乌龟对对碰是网络直播平台中流行的一种盲盒玩法。主播准备了一个盲盒序列，消费者需要按盲盒序列从前到后的顺序，先购买数包树脂乌龟盲盒。消费者在下单之后，主播会现场拆开消费者的树脂乌龟盲盒进行对对碰环节。

对对碰环节 1: 在该环节会把消费者的乌龟盲盒**按照原盲盒序列的顺序**拆开放在 9 宫格当中，按照从上到下从左到右的顺序给 9 宫格编号 1 到 9，按照编号从小到大的顺序给每一个格子放一个乌龟。查看 9 宫格是否满足下方的特定条件，满足条件按照赠送机制对应地进行操作。当满足下面的特定条件时，会把 9 宫格中的部分乌龟拿走，算作要给消费者的乌龟。

当所有特定条件都不满足的时候，9 宫格里的所有乌龟全部拿走，算作要给消费者的乌龟，对对碰环节结束。最终消费者获得**自己购买的乌龟**以及对对碰**赠送的乌龟**。

然后进入到环节 2。

对对碰环节 2: 向消费者的未拆开的盲盒中，补充满足特定条件对应数量的赠送盲盒。即如果当前剩余 x 个盲盒还没开，现在满足特定条件又赠送了 y 个盲盒，那么现在剩余的盲盒数量增加到 $x+y$ ，将这 y 个盲盒按顺序补在最后面，原盲盒序列的数量保证满足游戏中的赠送数量。然后进入到环节 3。

对对碰环节 3: 用剩余还没拆开的乌龟盲盒**按照原盲盒序列的顺序**去补充 9 宫格，按照编号从小到大的顺序给空的格子放一个乌龟。然后再进入到环节 1。

特定条件	要求	赠送机制
条件 1: 对对碰	场上有两个颜色相同的乌龟 (不一定相邻)	把场上成对出现的乌龟全部成对拿走，如果拿走了 x 对乌龟那么就赠送 x 个盲盒
条件 2: 清台	当 按照条件 1 拿走乌龟之后刚好场上没有任何一只乌龟	赠送 8 个盲盒
条件 3: 全家福	当 9 宫格的各个位置 都有乌龟 并且每只乌龟的颜色 各不相同	9 宫格中的所有乌龟拿走，赠送 10 个盲盒 (按本条件拿走所有乌龟后，不能按“条件 2 清台”赠送盲盒)。

乌龟总共有 10 种颜色 (用 0~9 表示)，现在假定有一个长度为 n 的盲盒序列，序列用字符串 S 表示，其中第 i 个字符代表第 i 个乌龟盲盒中乌龟的颜色。小 Z 看到别人买了很多乌龟也想组建自己的乌龟大军，于是小 Z 下单了 a 个盲盒，即盲盒序列中的第 1 个盲盒到第 a 个盲盒，赠送盲盒从盲盒序列的第 $a+1$ 个盲盒依次往后赠送，现在小 Z 想知道自己最后能拿到多少只乌龟。

【输入格式】

输入的第一行包含一个只包含数字 0 到 9 的字符串 S 。

输入的第二行包含一个整数 a ，代表购买的树脂乌龟盲盒数量。

【输出格式】

输出一个整数，代表小 Z 最终能够拿走多少只乌龟。

样例	turtle.in	turtle.out
【输入输出数据样例 1】	012334567888999 10	12
【输入输出数据样例 2】	2012345678901234501890123 10	23

【样例解释 1】

<p>1.盲盒序列：012334567888999 放进九宫格之前没开的盲盒：0 1 2 3 3 4 5 6 7 8 放入九宫格之后，如右图所示：</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td><td>2</td></tr> <tr><td>3</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td></tr> </table>	0	1	2	3	3	4	5	6	7	<p>2.手里没开的盲盒：8 已经拿走的盲盒：无 满足条件 1（赠送 1 个盲盒），处理之后如右图所示：</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td><td>2</td></tr> <tr><td></td><td></td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td></tr> </table>	0	1	2			4	5	6	7
0	1	2																	
3	3	4																	
5	6	7																	
0	1	2																	
		4																	
5	6	7																	
<p>3.手里没开的盲盒：8 8 已经拿走的盲盒：3 3 放入九宫格之后，如右图所示：</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td><td>2</td></tr> <tr><td>8</td><td>8</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td></tr> </table>	0	1	2	8	8	4	5	6	7	<p>4.满足条件 1（赠送 1 个盲盒），处理之后如右图所示：</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td><td>2</td></tr> <tr><td></td><td></td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td></tr> </table>	0	1	2			4	5	6	7
0	1	2																	
8	8	4																	
5	6	7																	
0	1	2																	
		4																	
5	6	7																	
<p>5.手里没开的盲盒：8 已经拿走的盲盒：3 3 8 8 放入九宫格之后，如右图所示：</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td><td>2</td></tr> <tr><td>8</td><td></td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td></tr> </table>	0	1	2	8		4	5	6	7	<p>6.不满足所有 3 个条件，游戏结束，9 宫格里的盲盒全部拿走 手里没开的盲盒：无 已经拿走的盲盒：3 3 8 8 0 1 2 8 4 5 6 7 因此输出答案为 12</p>									
0	1	2																	
8		4																	
5	6	7																	

【数据范围】

对于所有数据，满足 S 的长度 $\leq 5 \times 10^5$ ， $10 \leq a \leq 1 \times 10^5$ 。

数据点编号	S 的长度 \leq	a \leq	特殊性质
1, 2	200	20	只满足特定条件 1
3, 4	200	20	只满足特定条件 1, 2
5, 6	200	20	只满足特定条件 1, 3
7,8,9,10	5×10^5	1×10^5	无特殊限制

有趣子序列 (sub)

【题目描述】

小唐最近在研究一个长为 n 的 01 字符串 S (下标从 1 开始)。

他很喜欢其子区间和子序列, 例如: 如下表的 01 字符串 $S = "01011010"$

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
0	1	0	1	1	0	1	0

子区间 $[2,4]$ 即为: S_2, S_3, S_4 这样一个在原 S 串中连续的一段序列;

子序列即类如: S_2, S_4, S_6, S_7 这样一个按原 S 串顺序的, 可连续、可不连续的序列。

所以, 子区间肯定是子序列, 但子序列不一定是子区间。

小唐会询问你 Q 次, 每次询问给出一个 S 的子区间 $[l, r]$, 他想知道是否存在 S 的一个“有趣子序列”, 其满足:

1. “有趣子序列”和询问的子区间 $[l, r]$ 相同;

例: 如果询问的子区间是 $[2,4]$, 其中, 子序列 S_2, S_6, S_7 和询问的子区间 S_2, S_3, S_4 中的字符按顺序一一对应相同;

2. “有趣子序列”不是从 S 中选出的一段子区间。

例: 子序列 S_2, S_6, S_7 并不是原 S 串中连续的一段序列, 即不是原 S 串的一段子区间。

请对于每次询问 $[l, r]$, 输出 Yes 或 No, 分别表示存在或不存在这样的“有趣子序列”。

【输入格式】

本题有多组数据。

第一行一个非负整数 T , 表示数据组数。

接下来 T 组数据, 对于每组数据:

第一行两个非负整数 n 和 q , 表示字符串长度和询问数。

第二行一个 01 字符串 s 。

接下来 q 行, 每行两个正整数 l, r , 表示询问的子区间。

【输出格式】

对于每组数据:

对于每个询问, 输出一行 Yes 或 No 表示答案。

【输入输出数据样例】

sub.in	sub.out
2	No
6 3	Yes
011100	Yes
2 4	No

1 3 3 5 5 2 11111 1 5 2 3	Yes
--	-----

【样例解释】

对于第一组数据，可以选择的子序列下标依次为：不存在，(1,2,4)，(3,4,6)。

对于第二组数据，可以选择的子序列下标依次为：不存在，(1,3)。

【数据范围】

对于所有数据，满足 $1 \leq n \leq 1 \times 10^5$ ， $q \leq 1 \times 10^5$ ， $1 \leq l \leq r \leq n$ 。

数据点编号	$T \leq$	$n \leq$	$q \leq$	特殊性质
1	100	4	1	
2	100	10	10	
3	10	12	100	
4	10	100	100	
5, 6	10	1000	1000	
7	5	1×10^5	1×10^5	$l=1$
8, 9, 10	5	1×10^5	1×10^5	

特殊性质： $l=1$ 表示子区间左边界为 1