

class Solution {

public String solution(String video\_len, String pos, String op\_start, String op\_end, String[] commands) {

String answer = "";

int posInt = transformMinSecToSec(pos);

posInt = checkBetweenOp(op\_start, op\_end, posInt);

for (String command : commands) {

switch (command) {

case "prev" :

posInt = posInt - 10;

break;

case "next" :

posInt = posInt + 10;

break;

}

posInt = returnValidTime(video\_len, posInt);

posInt = checkBetweenOp(op\_start, op\_end, posInt);

}

return transforSecToMinSec(posInt);

}

private int transformMinSecToSec(String minSecStr) {

String[] minSecArr = minSecStr.split(":");

int returnSec = Integer.parseInt(minSecArr[0]) \* 60 + Integer.parseInt(minSecArr[1]);

return returnSec;

}

private String transforSecToMinSec(int targetSecond) {

if (targetSecond == 0) {

return "00:00";

}

int minute = targetSecond / 60;

int second = targetSecond % 60;

String returnStr = "";

if (minute < 10) {

returnStr = "0" + minute;

} else {

returnStr = minute + "";

}

returnStr += ":";

if (second < 10) {

returnStr += "0" + second;

} else {

returnStr += second;

}

return returnStr;

}

private int checkBetweenOp(String opStartStr, String opEndStr, int pos) {

int opStart = transformMinSecToSec(opStartStr);

int opEnd = transformMinSecToSec(opEndStr);

if ((pos > opStart || pos == opStart) && (pos < opEnd || pos == opEnd)) {

pos = opEnd;

}

return pos;

}

private int returnValidTime(String videoLenStr, int pos) {

int videoLen = transformMinSecToSec(videoLenStr);

int resultPos = 0;

if (pos > videoLen) {

resultPos = videoLen;

} else if (pos < 0) {

resultPos = 0;

} else {

resultPos = pos;

}

return resultPos;

}

}