

## 7482 Association for Computing Machinery

ACM (Association for Computing Machinery) organizes the International Collegiate Programming Contest (ICPC) worldwide every year.

In the ICPC, a team of three students is presented with a problem set that contains  $N$  problems (the largest number of problems in an official ACM ICPC to date is probably the recent ACM ICPC World Finals in Marrakesh, Morocco where  $N = 13$ ) of varying types and difficulty levels. The teams are not told which problems are easier (or harder). As there is only one single computer per team, each team has to decide which one of the  $N!$  possible problem solving orders that the team wants to use. This is called the “contest strategy” and teams who are expecting to do well in an ICPC should use the optimal contest strategy for their team.

However, when a contest has ‘First to Solve Problem [ $A$ ]/ $B$ /.../ $A$ +( $N - 1$ )] award’ — like this ICPC SG Regional Contest 15 — sponsored by Kattis, then some of the teams may throw the optimal contest strategy out of the window in order to grab the (smaller) award.

### Input

The input file contains several test cases, each of them describes a hypothetical scenario of a 300 minute contest, following the statement below.

The first line contains two non-negative integers  $2 \leq N \leq 13$  and  $0 \leq p \leq N - 1$ . The integer  $N$  describes the number of problems in the problem set of this ACM ICPC and the integer  $p$  is a 0-based index that describes the problem id that your team wants to solve first before attempting the other  $N - 1$  problems.

The next line contains  $N$  integers in the range between 1 and 999, inclusive. Each integer  $i$  describes the estimated number of minutes to solve problem id  $i$  according to your team.

As an ACM ICPC duration is 5 hours, or 300 minutes, any estimation of strictly larger than 300 minutes for a certain problem  $j$  basically says that you estimate that your team cannot solve problem  $j$  during contest time.

In this problem, you can assume that all your team’s estimations are perfectly accurate, i.e. if your team estimates that your team needs 30 minutes to solve problem  $k$ , 270 minutes to solve another problem  $l$ , and have no idea how to solve the rest, and decides to solve problem  $k$  first followed by  $l$ , then after 30 minutes have elapsed from the start of contest, your team really gets an ‘Accepted’ verdict from Kattis for problem  $k$ , immediately switches to problem  $l$  for the next 270 minutes, gets another ‘Accepted’ verdict from Kattis for problem  $l$  at exactly 300 minutes (in this problem, submission at minute 300 is a valid submission (to simplify this problem a bit, we ignore seconds although the last submission must be done at 04 hours, 59 minutes, and 59 seconds (or minute 299) in order to be considered valid.). Thus you have 2 Accepted problems and the total penalty time of  $30 + 300 = 330$  minutes as per the ICPC rules.

### Output

For each case, print two integers  $Num\_AC$  and  $Penalty\_Time$  separated by a single space in one line.

$Num\_AC$  is the *highest* number of problems that your team can solve and  $Penalty\_Time$  is the *lowest* penalty minutes that your team can get in order to solve  $Num\_AC$  problems in this 300 minutes ACM ICPC **if** your team insists to solve problem  $p$  first from the start of the contest and then use the remaining time to work on the other  $N - 1$  problems.

For the example scenario above, if your team decides to solve problem  $l$  first followed by  $k$ , then your team still solves  $Num\_AC = 2$  Accepted problems, but with the total penalty of  $270 + 300 = 570$  minutes.

### Sample Input

```
7 0
30 270 995 996 997 998 999
7 1
30 270 995 996 997 998 999
7 2
30 270 995 996 997 998 999
3 0
1 300 299
```

### Sample Output

```
2 330
2 570
0 0
2 301
```