Picking Numbers

Given an array of integers, find and print the maximum number of integers you can select from the array such that the absolute difference between any two of the chosen integers is ≤ 1.

Input Format

The first line contains a single integer, n, denoting the size of the array.
The second line contains n space-separated integers describing the respective values of $a_0, a_1, \ldots, a_{n-1}$.

Constraints

- $2 \leq n \leq 100$
- $0 < a_i < 100$
- The answer will be ≥ 2.

Output Format

A single integer denoting the maximum number of integers you can choose from the array such that the absolute difference between any two of the chosen integers is ≤ 1.

Sample Input 0

```
6
4 6 5 3 3 1
```

Sample Output 0

```
3
```

Explanation 0

We choose the following multiset of integers from the array: $\{4, 3, 3\}$. Each pair in the multiset has an absolute difference ≤ 1 (i.e., $|4-3| = 1$ and $|3-3| = 0$), so we print the number of chosen integers, 3, as our answer.

Sample Input 1

```
6
1 2 2 3 1 2
```

Sample Output 1

```
5
```

Explanation 1

We choose the following multiset of integers from the array: $\{1, 2, 2, 1, 2\}$. Each pair in the multiset has an absolute difference ≤ 1 (i.e., $|1-2| = 1$, $|1-1| = 0$, and $|2-2| = 0$), so we print the number of chosen integers, 5, as our answer.

Breaking the Records

HackerRank

Maria plays n games of college basketball in a season. Because she wants to go pro, she tracks her points scored per game sequentially in an array defined as $score = [s_0, s_1, \ldots, s_{n-1}]$. After each game i, she checks to see if score s_i breaks her record for most or least points scored so far during that season.

Given Maria's array of $scores$ for a season of n games, find and print the number of times she breaks her record for *most* and *least* points scored during the season.

Note: Assume her records for most and least points at the start of the season are the number of points scored during the first game of the season.

Input Format

The first line contains an integer denoting n (the number of games).
The second line contains n space-separated integers describing the respective values of $s_0, s_1, \ldots, s_{n-1}$.

Constraints

- $1 \leq n \leq 1000$
- $0 \leq s_i \leq 10^8$

Output Format

Print two space-seperated integers describing the respective numbers of times her best (highest) score increased and her worst (lowest) score decreased.

Sample Input 0

```
9
10 5 20 20 4 5 2 25 1
```

Sample Output 0

```
2 4
```

Explanation 0

The diagram below depicts the number of times Maria broke her best and worst records throughout the season:

Game	0	1	2	3	4	5	6	7	8
Score	10	5	20	20	4	5	2	25	1
Highest Score	10	10	20	20	20	20	20	25	25
Lowest Score	10	5	5	5	4	4	2	2	1

She broke her best record twice (after games **2** and **7**) and her worst record four times (after games **1**, **4**, **6**, and **8**), so we print **2 4** as our answer. Note that she *did not* break her record for best score during game **3**, as her score during that game was *not* strictly greater than her best record at the time.

Sample Input 1

```
10
3 4 21 36 10 28 35 5 24 42
```

Sample Output 1

```
4 0
```

Explanation 1

The diagram below depicts the number of times Maria broke her best and worst records throughout the season:

Game	0	1	2	3	4	5	6	7	8	9
Score	3	4	21	36	10	28	35	5	24	42
Highest Score	3	4	21	36	36	36	36	36	36	42
Lowest Score	3	3	3	3	3	3	3	3	3	3

She broke her best record four times (after games 1, 2, 3, and 9) and her worst record zero times (no score during the season was lower than the one she earned during her first game), so we print $4\ 0$ as our answer.

ngry Professor

HackerRank

crete Mathematics professor has a class of N students. Frustrated with their lack of discipline, he des to cancel class if fewer than K students are present when class starts.

ven the arrival time of each student, determine if the class is canceled.

nput Format

The first line of input contains T, the number of test cases.

Each test case consists of two lines. The first line has two space-separated integers, N (students in the class) and K (the cancelation threshold). The second line contains N space-separated integers $(a_1, a_2, \ldots, a_N)$ describing the arrival times for each student.

Note: Non-positive arrival times ($a_i \leq 0$) indicate the student arrived early or on time; positive arrival times ($a_i > 0$) indicate the student arrived a_i minutes late.

Constraints

- $1 \leq T \leq 10$
- $1 \leq N \leq 1000$
- $1 \leq K \leq N$
- $-100 \leq a_i \leq 100, where\ i \in [1, N]$

Output Format

For each test case, print the word YES if the class is canceled or NO if it is not.

Note
If a student arrives exactly on time $(a_i = 0)$, the student is considered to have entered before the class started.

Sample Input

```
2
4 3
-1 -3 4 2
4 2
0 -1 2 1
```

Sample Output

```
YES
NO
```

Explanation

For the first test case, $K = 3$. The professor wants at least 3 students in attendance, but only 2 have arrived on time (-3 and -1). Thus, the class is canceled.

For the second test case, $K = 2$. The professor wants at least 2 students in attendance, and there are 2 who have arrived on time (0 and -1). Thus, the class is *not* canceled.

Mini-Max Sum

Given five positive integers, find the minimum and maximum values that can be calculated by summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers.

For example, $arr = [1,3,5,7,9]$. Our minimum sum is $1+3+5+7 = 16$ and our maximum sum is $3+5+7+9 = 24$. We would print

```
16 24
```

Function Description

Complete the *miniMaxSum* function in the editor below. It should print two space-separated integers on one line: the minimum sum and the maximum sum of 4 of 5 elements.

miniMaxSum has the following parameter(s):

- *arr*: an array of 5 integers

Input Format

A single line of five space-separated integers.

Constraints

$1 \leq arr[i] \leq 10^9$

Output Format

Print two space-separated long integers denoting the respective minimum and maximum values that can be calculated by summing exactly *four* of the five integers. (The output can be greater than a 32 bit integer.)

Sample Input

```
1 2 3 4 5
```

Sample Output

```
10 14
```

Explanation

Our initial numbers are 1, 2, 3, 4, and 5. We can calculate the following sums using four of the five integers:

1. If we sum everything except 1, our sum is $2+3+4+5 = 14$.
2. If we sum everything except 2, our sum is $1+3+4+5 = 13$.
3. If we sum everything except 3, our sum is $1+2+4+5 = 12$.
4. If we sum everything except 4, our sum is $1+2+3+5 = 11$.
5. If we sum everything except 5, our sum is $1+2+3+4 = 10$.

Hints: Beware of integer overflow! Use 64-bit Integer.

thday Cake Candles

HackerRank

in charge of the cake for a child's birthday. You have decided the cake will have one candle for each year of total age. They will only be able to blow out the tallest of the candles. Count how many candles are tallest.

ple

$idles = [4, 4, 1, 3]$

ie maximum height candles are **4** units high. There are **2** of them, so return **2**.

Function Description

Complete the function [illegible] in the editor below.

birthdayCakeCandles has the following parameter(s):

- *int candles[n]*: the candle heights

Returns

- *int*: the number of candles that are tallest

Input Format

The first line contains a single integer, n, the size of $candles[]$.
The second line contains n space-separated integers, where each integer i describes the height of $candles[i]$.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq candles[i] \leq 10^7$

Sample Input 0

```
4
3 2 1 3
```

Sample Output 0

```
2
```

Explanation 0

Candle heights are $[3, 2, 1, 3]$. The tallest candles are **3** units, and there are **2** of them.

Grading Students

HackerLand University has the following grading policy:

- Every student receives a $grade$ in the inclusive range from 0 to 100.
- Any $grade$ less than 40 is a failing grade.

Sam is a professor at the university and likes to round each student's $grade$ according to these rules:

- If the difference between the $grade$ and the next multiple of 5 is less than 3, round $grade$ up to the next multiple of 5.
- If the value of $grade$ is less than 38, no rounding occurs as the result will still be a failing grade.

Examples

- $grade = 84$ round to 85 (85 - 84 is less than 3)
- $grade = 29$ do not round (result is less than 40)
- $grade = 57$ do not round (60 - 57 is 3 or higher)

Given the initial value of $grade$ for each of Sam's n students, write code to automate the rounding process.

Function Description

Complete the function *gradingStudents* in the editor below.

gradingStudents has the following parameter(s):

- *int grades[n]*: the grades before rounding

Returns

- *int[n]*: the grades after rounding as appropriate

Input Format

The first line contains a single integer, n, the number of students.
Each line i of the n subsequent lines contains a single integer, $grades[i]$.

Constraints

- $1 \leq n \leq 60$
- $0 \leq grades[i] \leq 100$

Sample Input 0

```
4
73
67
38
33
```

Hackput 0

lanation 0

ID	Original Grade	Final Grade
1	73	75
2	67	67
3	38	40
4	33	33

1. Student 1 received a 73, and the next multiple of 5 from 73 is 75. Since $75 - 73 < 3$, the student's grade is rounded to 75.
2. Student 2 received a 67, and the next multiple of 5 from 67 is 70. Since $70 - 67 = 3$, the grade will not be modified and the student's final grade is 67.
3. Student 3 received a 38, and the next multiple of 5 from 38 is 40. Since $40 - 38 < 3$, the student's grade will be rounded to 40.
4. Student 4 received a grade below 33, so the grade will not be modified and the student's final grade is 33.