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

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

Change Theme Language Python 3

```python
import re
import sys

#
# Complete the 'gradingStudents' function below.
#
# The function is expected to return an INTEGER_ARRAY.
# The function accepts INTEGER_ARRAY grades as parameter.
#

def gradingStudents(grades):
    # Write your code here
    result = [grade for grade in grades]

    for idx, marks in enumerate(grades):
        if marks >= 38:
            next_multiple = ((marks // 5) + 1 ) * 5
            if next_multiple - marks < 3:
                marks = next_multiple

        result[idx] = marks

    return result

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    grades_count = int(input().strip())

    grades = []

    for _ in range(grades_count):
        grades_item = int(input().strip())
        grades.append(grades_item)

    result = gradingStudents(grades)

    fptr.write('\n'.join(map(str, result)))
    fptr.write('\n')
    fptr: TextIOWrapper

    fptr.close()

```

Line: 28 Col: 18

Upload Code as File Test against custom input Run Code Submit Code

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6

Compiler Message

Success

Input (stdin) Download

```
4
73
67
38
33
```

Expected Output Download

```
75
```