**HackerRank Algorithms**

**Birthday Cake Candles**

#!/bin/python3

import math

import os

import random

import re

import sys

# Complete the birthdayCakeCandles function below.

def birthdayCakeCandles(ar):

    return ar.count(max(ar))

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    ar\_count = int(input())

    ar = list(map(int, input().rstrip().split()))

    result = birthdayCakeCandles(ar)

    fptr.write(str(result) + '\n')

    fptr.close()

**Smart Number**

import math

def prime\_factors(num):

    index, factors = 2, []

    while index\*\*2 <= num:

        if num % index: index += 1

        else:

            num //= index

            factors += [index]

    if num > 1: factors += [num]

    return factors

def is\_smart\_number(num):

    return not len(prime\_factors(num)) % 2

for \_ in range(int(input())):

    num = int(input())

    ans = is\_smart\_number(num)

    if ans:

        print("YES")

    else:

        print("NO")

**Grading Students**

#!/bin/python3

import math

import os

import random

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):

    gradeslist = []

##    print(grades)

    # Write your code here

    for grade in grades:

        #if grade < 40: gradeslist += [grade]

        #else:

        #gradefloor = int(math.floor(grade\*0.2)/0.2)

        gradeceil = int(math.ceil(grade\*0.2)/0.2)

        #gradeslist += [[gradefloor, gradeceil][gradeceil-grade < grade-gradefloor]]

        if gradeceil-grade > 2 or gradeceil < 40: gradeslist += [grade]

        else: gradeslist += [gradeceil]

        #elif grade-gradefloor < 3: gradeslist += [gradefloor]

    return gradeslist

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.close()