



## **Experiment No. 1**

**Title: Interpretation of problem statement and Identification of test cases for given problem statement**



**Batch: B-4****Roll No: 16010422234****Name: Chandana Ramesh Galgali****Experiment No.:1****Aim:** To interpret given problem statement and identify test cases for given problem statement

---

**Resources needed:** Text Editor, C/C++ IDE

---

**Theory:**

In competitive programming the problem statement is mostly given pertaining to real-world scenarios. Along with input and output information, constraints on input/output are also given most of the times with the problem statement. Hence in competitive programming to get started with a problem, the first step is to read and understand the problem statement and given information and find the following details from it:

1. Identify the input values
2. Identify the constraints on input
3. Identify the output values
4. Identify the constraints on output

Along with problem statement and constraints information, input format and output format information is also given. To get a clear understanding of these formats, sample input values and its corresponding output values are also given. We refer to these values as sample test cases.

A Test Case is some sample input value and its expected out value. In competitive programming, with every problem statement, some sample test cases are given most of the times. But these sample test cases do not cover all the general and special cases. Since the competitive programming platforms evaluate the solution based on test cases, it is essential to identify the general and special test cases for the problem statement under consideration. Special Cases mostly require special handling in the solution. Test Cases can be identified using following approach:

1. Random Value Test Cases – Here some random values of input and it's corresponding output within the given input/output constraints can be considered
  2. Minimal Value Test Cases – Here considering factors, such as minimum number of total input values or minimum possible value for an input, special cases can be identified
  3. Maximal Value Test Cases - Here considering factors, such as maximum number of total input values or maximum possible value for an input, special cases can be identified. These test cases also help to determine problems like integer overflow, crash point of solution, maximum running time and memory requirement of the solution and so on.
-

**Activity:**

Consider the following problem statement and other information provided along with it:

**Problem**

You are provided an array of size that contains non-negative integers. Your task is to determine whether the number that is formed by selecting the last digit of all the N numbers is divisible by 10.

**Input format**

- First line: A single integer N denoting the size of array A
- Second line: N space-separated integers.

**Output format**

If the number is divisible by 10, then print 'Yes'. Otherwise, print 'No'

**Constraints**

$$1 \leq N \leq 10^5$$

$$0 \leq A[i] \leq 10^5$$

Sample Input	Sample Output
5 45 23 65 22 74	No

**Task 1:**

Identify the following from the given information:

1. Input values  
A single non-negative integer N denoting the size of array A  
N space-separated integers
2. Constraints on input values  
 $1 \leq$  A single integer N denoting the size of array  $A \leq 10^5$   
 $0 \leq$  N space-separated integers  $\leq 10^5$
3. Output values  
Yes  
No
4. Constraints on output values  
In Yes, ('Y' - uppercase, 'e' and 's' - lowercase)  
In No, ('N' - uppercase, 'o' - lowercase)
5. Specified format for input values  
First line: A single integer N denoting the size of array A  
Second line: N space-separated integers.

## 6. Specified format for output values

If the number is divisible by 10, then print 'Yes'. Otherwise, print 'No'

**Task 2:**

Identify general and special test cases for given problem statement. List down in all 10 - 12 test cases in table format as shown:

Sr. No.	Sample Input	Sample Output	Description	Test Case Type (general/special)
1.	5 45 23 65 22 74	No	array with 5 integer numbers	general (Random Value Test Case)
2.	7 1 3 5 7 9 2 40	Yes	array with 7 integer numbers	general (Random Value Test Case)
3.	3 160 104 234	No	array with 3 integer numbers	general (Random Value Test Case)
4.	2 98 70	Yes	array with 2 integer numbers	general (Random Value Test Case)
5.	$10^5$ 70 23 342 .... $10^5$	Yes	array with $10^5$ integer numbers including $10^5$	general (Maximal Value Test Case)
6.	$10^5$ 10 23 342 .... 7	No	array with $10^5$ integer numbers	general (Maximal Value Test Case)
7.	1 0	Yes	array with 1 integer number ie 0	general (Minimal Value Test Case)
8.	1 16	No	array with 1 integer number	general (Minimal Value Test Case)
9.	4 1 22 34 0	No	Array with 4 integer numbers including 0	general (Minimal Value Test Case)
10.	6 10 4 23 34 35 0	Yes	Array with 6 integer numbers including 0	general (Minimal Value Test Case)

**Solution:**

```


N=int(input())
A=[]
A=list(map(int, input().split(" ")))
if (A[N-1]%10==0):
    print("Yes")
else:
    print("No")

```









































```

PS C:\Users\chand\Downloads\IV SEM\CPL>
M/CPL/exp1.py"
5
45 23 65 22 74
No

```

RESULT:  Accepted[Refer judge environment](#)

Score	Time (sec)	Memory (KiB)	Language
20	0.26896	12368	Python 3.8

Input	Result	Time (sec)	Memory (KiB)	Score	Your output	Correct output	Diff
Input #1	 Accepted	0.024795	10536	10			
Input #2	 Accepted	0.033132	12368	10			
Input #3	 Accepted	0.016942	2	10			
Input #4	 Accepted	0.025465	9116	10			
Input #5	 Accepted	0.042097	10756	10			
Input #6	 Accepted	0.025559	8232	10			
Input #7	 Accepted	0.025423	7164	10			
Input #8	 Accepted	0.025532	7776	10			
Input #9	 Accepted	0.025138	6864	10			
Input #10	 Accepted	0.024873	7888	10			

---

**Outcomes: Inculcate the best practices that are essential for competitive programming**

---

**Post Lab Questions:**

Consider the given problem statement and related information:

**Problem**

You have been given a positive integer  $N$  where  $1 \leq N \leq 12$ . You need to find and print the Factorial of this number.

**Input Format**

The first and only line of the input contains a single integer  $N$  denoting the number whose factorial you need to find.

**Output Format**

Output a single line denoting the factorial of the number  $N$ .

Sample Input	Sample Output
3	6

**Task 1:**

Identify the following from the given information:

1. Input values  
The input consists of a single positive integer  $N$ .
2. Constraints on input values  
 $1 \leq N \leq 12$
3. Output values  
The output is a single integer representing the factorial of  $N$ .
4. Constraints on output values  
No specific constraints on the output values are mentioned.
5. Specified format for input values  
The input is given in a single line, and it contains only one integer ( $N$ ).
6. Specified format for output values  
The output is a single line containing an integer, which is the factorial of  $N$ .


**Task 2:**

Identify general and special test cases for the given problem statement. List down in all 6-8 test cases in table format (refer activity section for table format of test cases)

















Sr. No.	Sample Input	Sample Output	Description	Test Case Type
1	3	6	Random input value	General (Random Value Test Case)
2	1	1	Minimum input value	General (Minimal Value Test Case)
3	5	120	Random input value	General (Random Value Test Case)
4	7	5040	Random input value	General (Random Value Test Case)
5	9	362880	Random input value	General (Random Value Test Case)
6	12	479001600	Maximum input value	General (Maximal Value Test Case)
7	2	2	Random input value	General (Random Value Test Case)
8	4	24	Random input value	General (Random Value Test Case)

**Solution:**

```
def factorial(N):
    if N == 0:
        return 1
    else:
        factorial_of_the_number = N * factorial(N-1)
        return factorial_of_the_number
N = int(input())
factorial_of_the_number = factorial(N)
print(factorial_of_the_number)
```

RESULT:  Accepted [Refer judge environment](#)

Score	Time (sec)	Memory (KiB)	Language
20	0.06854	2	Python 3.8

Input	Result	Time (sec)	Memory (KiB)	Score	Your output	Correct output	Diff
Input #1	 Accepted	0.017101	2	25			
Input #2	 Accepted	0.016933	2	25			
Input #3	 Accepted	0.017485	2	25			
Input #4	 Accepted	0.017023	2	25			

**Conclusion: (Conclusion to be based on the objectives and outcomes achieved)**

The experiment achieved its objectives by comprehensively interpreting the problem statement and formulating a set of test cases that effectively cover various aspects of the problem, contributing to a thorough and systematic approach to testing the solution.

**References:**

1. Antti Laaksonen, "Guide to Competitive Programming", Springer, 2018
2. Gayle Laakmann McDowell, "Cracking the Coding Interview", CareerCup LLC, 2015
3. Steven S. Skiena Miguel A. Revilla, "Programming challenges, The Programming Contest Training Manual", Springer, 2006
4. Antti Laaksonen, "Competitive Programmer's Handbook", Hand book, 2018
5. Steven Halim and Felix Halim, "Competitive Programming 3: The Lower Bounds of Programming Contests", Handbook for ACM ICP