

**Batch: SY-IT(B3)**

**Experiment Number: 1**

**Roll Number: 16010423076**

**Name: Ritesh Jha**

**Aim of the Experiment:**

To interpret given problem statement and identify test cases for given problem statement

---

**Program/ Steps:**

**Task 1:**

Identify the following from the given information:

1. Input values

N: Size of the array.

A: An array of N non-negative integers.

2. Constraints on input values

$$1 \leq N \leq 105$$

$$0 \leq A[i] \leq 105$$

3. Output values

A single string:

"Yes" if the number formed by the last digits of the array is divisible by 10.

"No" otherwise.

4. Constraints on output values

Only one string (Yes or No) is expected as output.

5. Specified format for input values

First line: Integer N (the size of the array).

Second line: N space-separated integers representing the array A.

6. Specified format for output values

Single string output: either "Yes" or "No".

**Task 2:**

Identify general and special test cases for given problem statements. 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
2	4 40 20 30 10	Yes	Array with 4 integers where the last digit forms a number divisible by 10.	General
3	1 50	Yes	Single-element array where the last digit is divisible by 10.	Edge Case
4	1 7	No	Single-element array where the last digit is not divisible by 10.	Edge Case
5	3 12 34 50	Yes	Array where the last digit of the last number ensures divisibility by 10.	General
6	6	No	Array where	General

	11 22 33 44 55 61		none of the numbers contribute to a divisible-by-10 number.	
7	10 0 0 0 0 0 0 0 0 0 0	Yes	Array where none of the numbers contribute to a divisible-by-10 number.	Special
8	2 100000 500001	No	Array with large numbers that form a non-divisible number.	Special
9	3 100001 100002 100000	Yes	Array with larger numbers where the last digit of the last number ensures divisibility by 10.	Special
10	100000 (100000 repeated 99999 times followed by 0)	Yes	Maximum array size $N=105N=10^5N=105$ with the last digit 000, divisible by 10.	Special(Max Size)

```
/*
// Sample code to perform I/O:
```

```

#include <iostream>

using namespace std;

int main() {
    int num;

    cin >> num;                // Reading
input from STDIN

    cout << "Input number is " << num << endl;    // Writing
output to STDOUT
}

// Warning: Printing unwanted or ill-formatted data to output
will cause the test cases to fail
*/

#include <iostream>

using namespace std;

int main() {

    int N = 0;
    cin>>N;

    long data[N];
    for(auto i=0; i<N; i++)
        cin>>data[i];

    int ld = data[N-1] % 10;
    string ans = (ld == 0) ? "Yes" : "No";

```

```
// write your code here  
// ans =  
  
cout<<ans;  
  
return 0;  
}
```

---

### Output/Result:

19:19 vscode

Test against custom input ▼

Compile & Test code

Submit code

Log ID: 269393720 / Jan 23, 2025 09:26 AM PST (America/Los\_Angeles)

RESULT: Sample Test Cases Passed ?

Refer judge environment

Note: When you **Compile & Test code**, the code is run against sample inputs. When you **Submit code**, the code is run against sample input as well as multiple hidden test cases. In order to solve the problem, your code must pass all of the test cases.

Time (sec)	Memory (KiB)	Language
0.008463	2	C++14

Input

5  
85 25 65 21 84

Output

No

Expected Correct Output

No

?

Submission ID: 105702874

**RESULT:** Accepted [Refer judge environment](#)

Score	Time (sec)	Memory (KiB)	Language
0	0.19102	912	C++14

Input	Result	Time (sec)	Memory (KiB)	Score	Your output	Correct output	Diff
Input #1	Accepted	0.025729	872	10			
Input #2	Accepted	0.02508	912	10			
Input #3	Accepted	0.009414	2	10			
Input #4	Accepted	0.017239	2	10			
Input #5	Accepted	0.026156	828	10			
Input #6	Accepted	0.01779	2	10			
Input #7	Accepted	0.017412	2	10			
Input #8	Accepted	0.017944	2	10			
Input #9	Accepted	0.017053	2	10			
Input #10	Accepted	0.0172	2	10			

## Post Lab Question-Answers:

### Task 1:

Identify the following from the given information:

#### 1. Input values

A single positive integer N.

#### 2. Constraints on input values

$$1 \leq N \leq 12$$

#### 3. Output values

The factorial of the input integer N.

## 4. Constraints on output values

The output must be a single number representing  $N!$  within the range of a 64-bit integer.

## 5. Specified format for input values

A single integer  $N$  on one line.

## 6. Specified format for output values

A single integer (the factorial of  $N$ ) on one line.

**Task 2:**

Identify general and special test cases for given problem statements. 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 (general/special)
1	1	1	Minimum value of $N$ . Tests edge case at the lower bound.	Special
2	12	479001600	Maximum value of $N$ . Tests edge case at the upper bound.	Special
3	2	2	Small valid input, testing correctness of low factorial values.	Special

4	6	720	Mid-range input to verify correctness of the loop.	General
---	---	-----	---	---------

**Code :**

```
#include <iostream>

using namespace std;

int main() {
    int N;

    cout<<"Enter number : ";

    cin >> N;

    int factorial = 1;

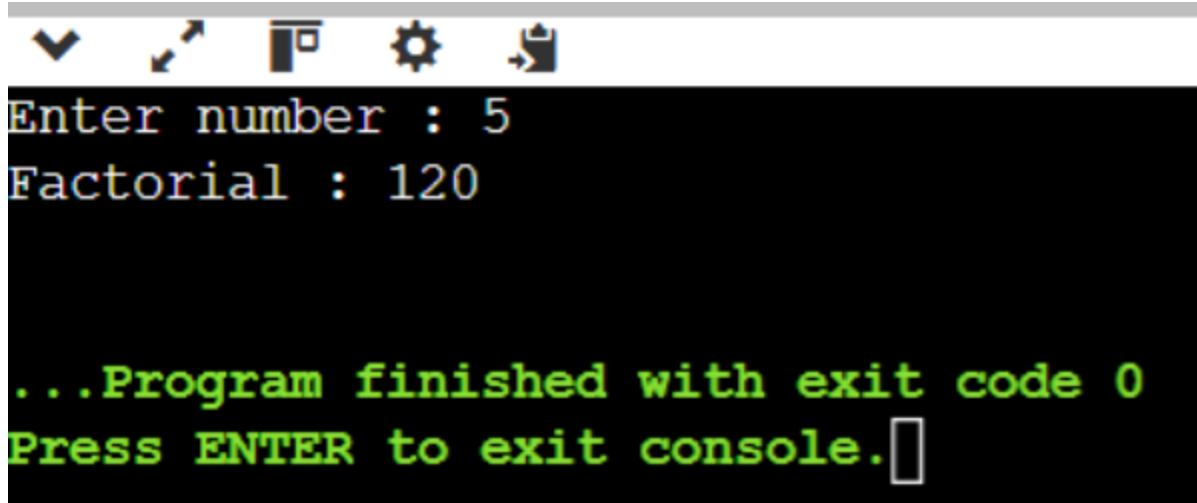
    for (int i = 1; i <= N; ++i) {
        factorial = factorial * i;
    }

    cout << "Factorial : "<< factorial << endl;

    return 0;
}
```

**Output :**



A screenshot of a console window with a dark background. At the top, there is a toolbar with icons for a dropdown menu, a cursor, a window, a gear, and a clipboard. The console text is as follows: 'Enter number : 5' in white, 'Factorial : 120' in white, '...Program finished with exit code 0' in green, and 'Press ENTER to exit console.' in green with a white cursor at the end.

```
Enter number : 5
Factorial : 120

...Program finished with exit code 0
Press ENTER to exit console.
```

---

**Outcomes:** CO1. Inculcate the best practices that are essential for competitive programming

---

**Conclusion (based on the Results and outcomes achieved):**

From this experiment, I learned how to approach competitive programming problems by carefully interpreting the problem statement, identifying input/output values, and recognizing constraints. I gained an understanding of how to break down problems into general and special test cases to ensure that all edge cases are covered. Additionally, I practiced writing code to implement solutions efficiently, focusing on handling different scenarios such as small and large inputs.

---

**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 ICPC