Experiment No. 1

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

Batch:A2   **Roll No: 16010421063**   **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 scenario. 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 it's corresponding output values are also given. We refer to these values as sample test cases.

A Test Case is some sample input value and it's 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 a 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 a 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
N, A[i] where 0<=i<n

2. Constraints on input values
$1 \leq N \leq 10^5$
$0 \leq A[i] \leq 10^5$

3. Output values
Yes or No with the new number

4. Constraints on output values
string s which is "Yes" or "NO" and 0<=new number<=10^5

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 53524	array with 5 integer numbers	general
2.	5 23 98 0 55 40	Yes 38050	array with 5 integer numbers	general.
3.	4 2 3 4 5	No 2345	array with 4 integer numbers	general
4.	5 0 0 0 0 0	Yes 0	array with 5 zeros. 10*0=0 hence 0 is divisible by 10	Special case
5.	6 98 76 88 99 68 65	No 868985	array with 6	general
6.	3 99999 100000 100000	Yes 900	array with 100000 as it is the upper limit	special case
7.	3 99999 100000 9999	No 909	array with 3 integers	general
8.	3 99988 100000 100000	Yes 800	array with 3 integers	general
9.	10 11 22 33 44 55 66 77 88 99 110	Yes 1234567890	array with 10 integers	general
10.	1 20	Yes 0	array with 1 which is lower bound of n	special case
11.	1 22	No 2	array with 1 which is lower bound of n	special case
12.	4 34 56 87 92	No 3672	array with 4 integers	general

(A Constituent College of Somaiya Vidyavihar University)

Solution:

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n;
    cin>>n;
    string s="";
    while(n--){
        int y;
        cin>>y;
        string x=to_string(y);
        s+=x[x.size()-1];
    }
    if(stoi(s)%10==0)cout<<"YES";
    else cout<<"NO";
    cout<<"\n"<<stoi(s);
    return 0;
}
```

Output	Output
/tmp/kwIG7TzZNZ.o	/tmp/kwIG7TzZNZ.o
5	5
10 22 45 33 89	11 22 33 44 50
NO	YES
2539	12340

Outcomes:

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

Post Lab Questions:

Consider the given problem statement and related information:
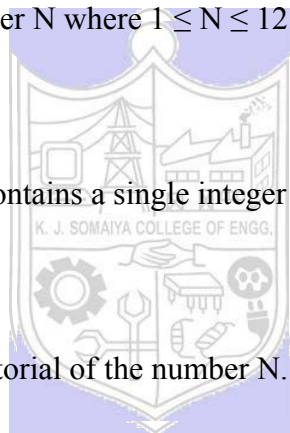
Problem

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
   N integer
2. Constraints on input values
   1<=N<=12
3. Output values
   fact integer
4. Constraints on output values
   1<=fact<=479001600
5. Specified format for input values

The first and only line of the input contains a single integer N denoting the number whose

(A Constituent College of Somaiya Vidyavihar University)

factorial you need to find.
6. Specified format for output values

Output a single line denoting the factorial of the number N

Task 2:
Identify general and special test cases for 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 (general/special)
1.	1	1	value 1 is lower bound	special
2.	12	479001600	value 12 is highest possible input	special
3.	2	2	value 2	general
4.	3	6	value 3!=3*2*1	general
5.	4	24	value 4!=4*3*2*1	general
6.	5	120	value 5!=5*4*3*2*1	general
7.	6	720	value 6!=6*5*4*3*2*1	general
8.	8	40320	array with 4 integers	general

```cpp
#include <iostream>
using namespace std;
int recurse(int n){
    if(n==0 || n==1)return 1;
    else {
        return n*recurse(n-1);
    }
}

int main()
{
    int n;
    cin>>n;
    cout<<recurse(n);
}
```

(A Constituent College of Somaiya Vidyavihar University)

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

learnt how to read the problem statement and solve the problem while keeping in mind the required input output format as well as the constraints

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