

INTRODUCTION TO COMPUTING – ASSIGNMENT 1

BSDS I year 2025–2026

Deadline: 11 November, 2025

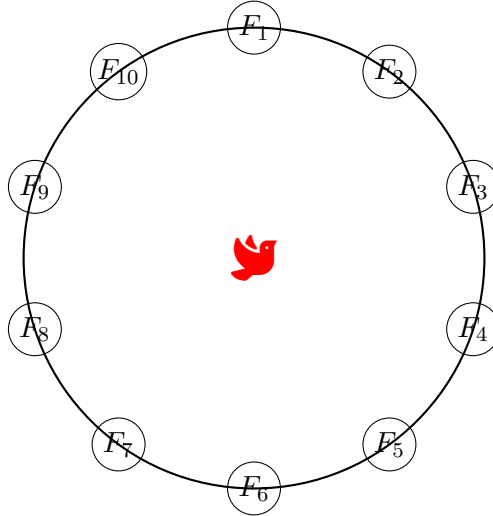
Total: 5 marks

SUBMISSION INSTRUCTIONS

1. Naming convention for your programs: `BSD-xx-25yy-progz.py` (assuming `BSD-xx-25yy` denotes your roll number, where `xx` denotes CC/DH/BG, and `progz` denotes the program name, where `z` denotes 1/2).
2. To submit the solution files (`.py` only), ensure that they are not password protected and upload them (within the stipulated time) through the link: <https://forms.gle/xcQA9U3WgZj8GF6p9>.

NOTE: The programs are to be written in Python and should be well commented. All programs should take the required inputs from standard input file and print the desired outputs to standard output file, until otherwise stated.

- Q1. Consider a lake in which n fishes labeled as $F_1, \dots, F_n$ are statically floating in a circular orientation as shown below. There is also a kingfisher flying over the lake that starts catching the fishes one by one. It starts from a particular position $p \leq n$ and catches every m^{th} fish on the circle. After a fish is caught, the counting starts from the clockwise succeeding fish to identify the next m^{th} one. For example, for $n = 10$, $p = 3$ and $m = 6$, the kingfisher first catches the fish F_8 , then the fish F_4 , and so on. Note that there is no change of position of the fishes during the process through which they are being caught by the kingfisher.



Write a program to take the values of n , p and m as inputs and return the sequence in which the fishes will be caught one after the other corresponding to their ordered label given as $F_1, \dots, F_n$. [3 marks]

Input Format

The input (to be taken from the user) is the three integers n , p and m given on a single line and separated by spaces.

Output Format

The output shows the sequence in which the fishes will be caught ordered as per their label $F_1, \dots, F_n$. This means it first prints the sequence at which the fish F_1 is caught, then the sequence at which the fish F_2 is caught, and so on.

Sample Input 1:

1 1 1

Sample Output 1:

1

Sample Input 2:

3 1 2

Sample Output 2:

2 1 3

Sample Input 3:

8 1 4

Sample Output 3:

5 4 6 1 3 8 7 2

Sample Input 4:

10 3 6

Sample Output 4:

3 8 7 2 10 9 5 1 4 6

Sample Input 5:

5 6 3

Sample Output 5:

INVALID

Q2. Let us define a string, consisting of letters from the English alphabet, as **exponentially shrinkable** if it can be written as follows.

- It is spread over one or more consecutive lines with no letter skipped or no extra letter included.
- Each line has a non-zero number of letters that follow the same order of letters appearing in the original string.
- The number of letters in a particular line is the multiplication of the letters appearing in the next pair of lines.
- The line with no two following lines (i.e., the previous to the last line) has exactly two letters.
- The line with not even a single following line (i.e., the very last line) has exactly one letter.

Note that a string is not necessarily **exponentially shrinkable**. Given a string as input, write a program to verify whether it is **exponentially shrinkable** or not. If so, print it in a **exponentially shrinkable** way. [2 marks]

Input Format

The input (to be taken from the user) is a string that contains letters from the sets [a-z] and [A-Z].

Output Format

The output is the representation of the input following the **exponentially shrinkable** pattern. The output prints **INVALID** if the input string is not **exponentially shrinkable**.

Sample Input 1:

i

Sample Output 1:

i

Sample Input 2:

ISI

Sample Output 2:

IS
I

Sample Input 3:

ahead

Sample Output 3:

ah
ea
d

Sample Input 4:

interconnectivity

Sample Output 4:

intercon
nect
iv
it
y

Sample Input 5:

Python

Sample Output 5:

INVALID