# Lab Exercise 4

CSDC102: Intermediate Programming

# Before your codes...

```
//******************************************************************
// Filename :
// Date :
// Subject :
// Second Semester, SY 2019 - 2020
// Activity : Lab 1A
// Problem Title :
// Input :
// Output :
//
// Honor Code : *insert honor code here*
//
// Complete Name :
// ID Number :
// Year-Course : 1-BSCS
// DCS, College of Computer Studies
// Ateneo de Naga University
//******************************************************************
```

Honor Code : This is my own program. I have not received any unauthorized help in completing this work. I have not copied from my classmate, friend, nor any unauthorized resource. I am well aware of the policies stipulated in the handbook regarding academic dishonesty.
If proven guilty, I won't be credited any points for this exercise.

# Lab 4A: Divine Gigi

## Program Description:

- Gigi loves *divine* numbers. Everybody knows that *divine* numbers are positive integers whose decimal representation contains only the *divine* digits **4** and **7**. For example, numbers **47**, **744**, **4** are divine and **5**, **17**, **467** are not.

- Gigi calls a number *semi-divine* if it could be evenly divided by some divine number. Help her find out if the given number *n* is *semi-divine*.

# Lab 4A: Divine Gigi

- Specifications:
    - Filename: **Lab4A_SURNAME.cpp**
    - Input file: standard input
    - Output: standard output

# Lab 4A: Divine Gigi

- Formats:
  - Input:
    - Input starts with *n*, the number of test cases. *n* test cases follow, *m* (1 ≤ *m* ≤ 1000) - the number that needs to be checked.
  - Output:
    - Print "YES" (without the quotes), if number *n* is *semi-divine*. Otherwise, print "NO" (without the quotes)

# Lab 4A: Divine Gigi

Sample Input

```
3
47 16 78
```

Sample Output

```
YES
YES
NO
```

***Note:***

*Note that all divine numbers that are semi-divine as any number is evenly divisible by itself.*

*In the first sample 47 is a divine number.*

*In the second sample 16 is divisible by 4.*

# Lab 4B: Baller’s Bananas

## Program Description:

- Kobe wants to buy *y* bananas in the shop. He has to pay *k* dollars for the first banana, 2*k* dollars for the second one and so on (in other words, he has to pay $i \cdot k$ dollars for the *i*-th banana).
- He has *n* dollars. How many dollars does he have to borrow from his friend Shaq to buy *y* bananas?

# Lab 4B: Baller’s Bananas

- Formats:
  - Input:
    - The first line contains three positive integers $k$,$n$,$y$ ($1 \le k, y \le 1000$, $0 \le n \le 109$), the cost of the first banana, initial number of dollars Kobe has and number of bananas he wants.
  - Output:
    - Output one integer — the amount of dollars that Kobe must borrow from Shaq. If he doesn't have to borrow money, output *0*.

# Lab 4B: Baller's Bananas

- Specifications:
    - Filename: **Lab4B_SURNAME.cpp**
    - Input file: **mamba.in**
    - Output: **legend.out**

# Lab 4B: Baller’s Bananas

Sample Input

```
3 7 4
```

Sample Output

```
13
```

# Lab 4C: Casino

## Program Description:

- Suppose we play a game where you are given *N* number of turns and *T* amount of starting money. On each turn you may only choose between two moves:
    - Move A: you lose P1 (1 peso) when you choose this move
    - Move B: you count how much money you have left. If it’s an even amount, you win P3, otherwise, you lose P5.

# Lab 4C: Casino

## Program Description:

- Output the optimal sequence of moves you need to play for each turn to arrive at the maximum possible earning or minimum possible loss from your starting money. Output as well the total number of earnings you get given *N* and *T*. It is required that you choose a move for each turn (you can’t pass on a turn).

# Lab 4C: Casino

- Specifications:
  - Filename: **Lab4C_SURNAME.cpp**
  - Input file: standard input
  - Output: standard output

# Lab 4C: Casino

- Formats:
  - Input:
    - The input consists of several integer pairs *N* and *T* where
      ($1 \leq N, T \leq 10^6$).
  - Output:
    - Print “strategy: “ followed by the sequence of moves you will take.
      On a separate line, print “total earned: “ followed by the total earnings you get *N* after turns.
  - Constraints:
    - Maximum earnings will not exceed $10^{30}$. To express a loss, follow the format:
      total earned: P-2.00

# Lab 4C: Casino

Sample Input

```
1 2
2 1
```

Sample Output

```
strategy: B
total earned: P3.00
strategy: AB
total earned: P2.00
```