1 Assignment 2 Introduction

This assignment is due *before* 11:59pm on the listed date. Which means, submissions made *on or after* 11:59pm will be counted as a late submissions. Late submissions *before* 11:59pm on the following day will receive a 25% point deduction as penalty. Submissions made *on or after* 11:59pm on the same day will *not* be graded. We strongly recommend completing the assignment before then.

Assignments due on Mondays generally involve materials covered in lectures from the previous week, whereas assignments due on Thursdays involve materials covered in lectures from the running week. So be sure to watch the lectures and go over the reading materials before attempting the assignments.

Good luck!

2 No Strings Attached (40 points)

You are tasked with creating a simple 'Rock Paper Scissors' game, but with a little twist! You are not allowed to use `string` (other than with `std::cout`) or `char` in your code.

To help you with this problem, you are given a `RandNumGen` function that randomly generates (and returns) `1`, `2`, or `3` when called.

```
#include <iostream>
#include <random>

std::random_device rd;
std::mt19937 gen(rd());

int RandNumGen() {
    std::uniform_int_distribution<> rng(1, 3);
    return rng(gen);
}

int main() {
    //Your code here
    return 0;
}
```

Below is an example of how your program should behave:

```
Enter 1 (Rock), 2 (Paper), or 3 (Scissors): 2
You picked: Paper
The computer picked: Rock
You win!
```

General Instructions:

- When writing your program, you may only use concepts you have learned in the course thus far, with the exception of `string` and `char`.
- Your program should compile and run. You can assume the input will be `1`, `2`, or `3`.
- The formatting must match exactly as shown in the example above, including whitespaces.

3 Tic-tac-toe Validator (60 points)

Everyone knows how to play tic-tac-toe. At the end of a tic-tac-toe match, we are left with a board with `X`s and `O`s. In this problem, we will refer to the final state of a tic-tac-toe board as a *configuration*. Using `0` (resp., `1`) to characterize `O` (resp., `X`), and `2` to characterize empty squares, we can represent the configuration of a tic-tac-toe board with an integer. For example:

```
X O O                   O O X                   O O O
O X X  =  100011110       X X  =  1211221       X X      =  112210
X X O                       X                     X O
```

Note that not all configurations have 9 digits, due to some having leading 0's. Your task is to write a function `ValidateBoard` that takes a configuration as input, and returns the string `X wins!`, `O wins!`, `It's a draw!`, or `Invalid configuration!`, depending on the configuration. An invalid configuration is a configuration that is impossible to achieve if the rules are followed. For example, `111111111` is an invalid configuration, as it is impossible for a tic-tac-toe board to have 9 `X`s.

```
#include <iostream>
#include <string>

std::string ValidateConfig(int config) {
    //Your code here
}

int main() {
    int config{0};
    std::cout << "Enter configuration: ";
    std::cin >> config;
    std::cout << "Result: " << ValidateConfig(config) << std::endl;
    return 0;
}
```

Below is an example of how your program should behave:

```
Enter configuration: 1211221
Result: X wins!
```

General Instructions:

- When writing your program, you may only use concepts you have learned in the course thus far.
- Your program should compile and run. You can assume the input will always be an integer that represents a configuration. For example, you do not need to consider inputs like `1010101010`, `9`, and so on.
- The formatting should match exactly as shown in the example above, including white-spaces.

4 Assignment 1 Submission Process

- Create a folder, name it `your_msu_id2`. For example, if your MSU email is `johndoe@msu.edu`, then you should name the folder `johndoe2`.
- For each programming task, create a sub-folder inside your `your_msu_id2` folder, and name it as the number that corresponds to the programming task number. For this assignment, there should be two sub-folders named '2' and '3'.
- Inside each sub-folder, put the `main.cpp` for the appropriate solution.
- Compress/Zip `your_msu_id2` folder and name it `your_msu_id2.zip`. For example, if the name of your folder is `johndoe2`, then you need to create a zip file named `johndoe2.zip`. Zip file guide: `https://copyrightservice.co.uk/reg/creating-zip-files`.
- Submit the zip file through D2L.