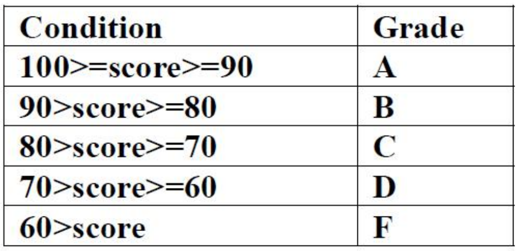
Lab 3: Conditionals, Loops and Arrays

Setup

1. In command line, change directory to lab directory which was created by cloning your bitbucket lab repository. If you haven't clonned your repository, you should clone it as described in previous labs.
2. Create a new directory named "lab3" in lab folder and change directory to lab3
   a) mkdir lab3
   b) cd lab3

Exercise 1: Write a program that prints the grade

Description: Your program will accept accept a score as a number and print the corresponding grade based on the below table



Condition	Grade
100>=score>=90	A
90>score>=80	B
80>score>=70	C
70>score>=60	D
60>score	F

Your code will executed as below (Second line is the expected ooutput):

```
> java PrintGrade 77
```

> C

1. Open a text editor and create a class named `PrintGrade` in the "lab3" directory.
2. Implement the main method of the `PrintGrade`.
3. Compile and run `PrintGrade` class.

Exercise 2 : Write a program that prints the prime numbers less than the given number

Description: Your program will accept a number as an argument such as shown below and print the Fibonacci Sequence up to that number.

Your code will executed as below (Second line is the expected ooutput):

```
> java FindPrimes 50
> 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47
```

1. Open a text editor and create a class called `FindPrimes` in the "lab3" directory..

2. Implement the main method of the `FindPrimes` to print the prime numbers described above..
3. Compile and run `FindPrimes` class

Exercise 3 : Write a matrix addition function

1. Open a text editor and create a class called MatrixCalculator in the “lab3” directory.
2. Declare and initialize the following 2 dimensional matrix in the main method of MatrixCalculator class

```
int[][] martrixA = {{6, 8, 2}, {9, 5, 11}, {7, 2, 5}};
int[][] martrixB = {{4, 6, 3}, {5, 8, 1}, {6, 6, 7}};
```

3. Write a function which accepts two parameters having type int[][]. Your function will perform the matrix addition operation for the given parameters and return the resulting matrix. Call the function from the main method with the parameters martrixA and martrixB
4. Print the result returned by the function to the console
5. Compile and run MatrixCalculator class

NOTE: In order to get a grade for this lab

- Your Bitbucket account name should have the format described in lab1.pdf
- Your repository name should be lab
- Your files should be in a folder named “lab3” in the repository
- Your files should be compiled successfuly
- You should complete all the steps in the exercises
- Your files should be submitted to Bitbucket.
    - You have to add commit and push files as described in lab1.pdf and lab2.pdf