C1-S2 PRACTICE

FIRST STEPS IN C++

At the end of this practice, you should be able to…

- ✓ **Execute** and debug C++ programs
- ✓ Transition from C-style printf and scanf to **cout** and **cint**
- ✓ Manipulate **if-else** and **switch**-**case** conditions
- ✓ Manipulate sdt::**string** and sdt::**vector**
- ✓ Iterate through array elements or vector with a **for** loop
- ✓ Iterate with a **do-while** loop to validate user inputs

How to compile your code?

Assuming your file is named: exercise.cpp:

- ✓ Open a **terminal** at your file location
- ✓ **Compile** your Program using the following command

```
g++ -o exercise exercise.cpp
```

- ✓ **Run** Your Program using the following command

```
./exercise
```

How to submit?

- ✓ Make a report PDF containing the screenshot of your program code and output for each exercise.
- ✓ **Submit your final source code report PDF to Microsoft team and turn it in.**

Are you lost?

You can read the following documentation to be ready for this practice:

- ✓ [WC3 School – Quick Start]
- ✓ [Geek For Geek – First program in C++]
- ✓ [You Tube Playlist on C++ course]

EXERCISE 1

- *Please read the start code* 😊
- *This exercise is based on you preassessment test*

Objectives

We want to check if a given number is inside a given range.

1- Input the number and the range defined by a minimum and maximum
2- Output whether the number is "inside" or "outside" the range

Input – Outputs

INPUT	int	The number
	int	The range min
	int	The range max
OUTPUT	string	The result

Examples

INPUT	**OUTPUT**	**EXPLANATION**
14 10 20	inside	*14 is in the range [10, 20]*
10 10 20	inside	*10 is in the range [10, 20]*
20 18 21	inside	*20 is in the range [18, 21]*
9 10 20	outside	*9 is out of the range [10, 20]*
-15 10 20	outside	*-15 is out of the range [10, 20]*

Q1 - Write a function that determines whether a number if in a specified range

```
bool isInside(int number, int min, int max)
```

- This function returns **true** if the number is within the range [**min, max**], and false otherwise

Q2 – Write the main() function to input the data and output the result

- Make sure your code work for all the above test cases

Q3 – (**BONUS**) If the user accidentally enters a minimum value that is larger than the maximum value, ask them to enter the values again.

EXERCISE 2

- 💡 *Please read the start code* 😊
- 💡 *This exercise is based on you preassessment test*

Objectives

We want to compare the ages of two people based on their dates of birth and determines which person is the youngest.

1- Input the date of birth
2- Output which person is the youngest or if they are of the same age.

Input – Outputs

INPUT	int	*person 1 year of birth*
	int	*person 1 month of birth*
	int	*person 1 day of birth*
	int	*person 1 year of birth*
	int	*person 1 month of birth*
	int	*person 1 day of birth*
OUTPUT	string	The result text

Examples

INPUT	**OUTPUT**	**EXPLANATION**
2002 12 31 2002 12 25	The first person is the youngest	The first person is born in dec 31th 2002 The second person is born in dec 25th 2002
2002 12 31 2002 12 31	Both persons have the same age	Both persons are born in dec 31th 2002
2001 12 31 2002 12 31	The second person is the youngest	The first person is born on dec 31th 2001 The second person is born on dec 31th 2002

Q1 – Write a function compareDates, describe as follows:

```
int compareDates(int year1, int month1, int day1, int year2, int month2, int day2)
```

- This function returns
  - -1 if date1 < date2
  - 0 if date1 == date2
  - 1 if date1 > date2

Q2 – Write the main() function to input the dates and output the result

- Make sure your code work for all the above test cases

Q3 – (**BONUS**) We want to ensure the input values for year, month, and day **are valid**:

The validation of dates is defined as follows:

Input type	Constraint
YEAR	A **positive** number
MONTH	A number **between 1 and 12**
DAY	Shall be **between 1 and the maximum number of days** in the given month (*)

(*) Some months have 28, 30 or 31 days - February has 28 days in a common year and 29 days in a leap year (every 4 years)

Write a function returning **whether a date data is valid or not**

```
bool isValidDate(int year, int month, int day)
```

- This function returns true if the date is valid, false otherwise
- Note: you can use the provided function isLeap(year) to check if a year is a leap year.

Examples

Year, Month, Day	Output	Explanation
-500, 04, 32	false	The year shall be a positive number
2001,13, 10	false	The month shall be a number between 1 and 12
2001, 02, 29	false	The number of days cannot exceed 28 for the month 2
2000, 04, 32	false	The number of days cannot exceed 30 for the month 4
2024, 08, 12	true	The date is valid
2000, 02, 29	true	The date is valid

Then, use this function to validate the inputs (the 2 dates)

- If ones of the 2 dates are invalid, output “invalid dates”
- Otherwise, write the result of the dates comparison

EXERCISE 3

- *Please read the start code* 😊
- *This exercise is based on you preassessment test*

Objectives

We want, for each student of our database:

- To calculate their **average score**
- To calculate their **GPA**
- To calculate their **grade letters**

How to compute the GPA?

The grade point average is computed as follow:

Average	GPA
90 to 100	4.0
80 to 89	3.0
70 to 79	2.0
60 to 69	1.0
Less than 60	0.0

How to compute the Grade Letter?

The grade point average is computed as follow:

GPA	GRADE LETTER
4.0	A
3.0	B
2.0	C
1.0	D
0.0	E

Q1 - Complete the function `calculateAverage`

```
string calculatAverage(Student student)
```

Note 1: the number of student scores can change! You can use the vector function size() to get the number of element in the vector

Note 2: Wich kind of loop will you choose to implement this function? Do-while? While? For? Justify your answer in your code

Q2 - Complete the function `calculateGPA`

```
float calculateGPA(Student student)
```

Q3 - Complete the function `calculateGrade`

```
string calculatGrade(Student student)
```

Q4 – In the main, call the function **printStudentStatistics** for each student

Note: the number of students can change!

You can use the vector function size() to get the number of element in the vector

- You should display the bellow output:

```
------------------------------------------------------------
| Student ID | Name      | Average Score |  GPA | Grade |
------------------------------------------------------------
|          1 |      Alice |          87.40 | 3.00 |     B |
|          2 |        Bob |          78.60 | 2.00 |     C |
|          3 |    Charlie |          90.40 | 4.00 |     A |
|          4 |      David |          70.00 | 2.00 |     C |
|          5 |        Eve |          67.20 | 1.00 |     D |
```

Q5 – Look at the following line of code:

```
std::fixed << std::setprecision(DECIMAL_DIGIT);
```

- What the **fixed** and **setprecision** function are used of?
- Change the precision to 4 digits and observe what happen