Exercise 1: Number [50 marks]

Write a program **number.c** to read in a positive integer *num* and complete the following 3 tasks:

1. Compute the product of all digits in *num*.

   For example, if *num* is 428, its product is 4*2*8 = 64.

   You are to write a function **compute_product()** for this task. You are to determine its return type and parameters.

2. Compute the difference between the largest digit and the smallest digit in *num*.

   For example, if *num* is 428, its smallest digit is 2 and biggest digit is 8 and difference is 8-2 = 6. Take note that the difference shall be a non-negative integer.

   You are to write a function **compute_difference()** for this task. You are to determine its return type and parameters.

3. Check if *num* is a hamming number. Hamming numbers are integers that can be expressed as $2^i*3^m*5^n$ where *i*, *m*, and *n* are integers bigger or equal to 0.

   2 and 10 are examples of hamming numbers while 11 and 14 are counter examples.

   You are to write a function **is_hamming_number()** for this task. You are to determine its return type and parameters.

You are not allowed to use array, string or any other structures that is yet covered in CS1010.

Three sample runs are shown below with user input highlighted in **bold**.

```
Enter a number: 15
Product of all digits: 5
Difference between largest and smallest digits: 4
15 is a hamming number
```

```
Enter a number: 321
Product of all digits: 6
Difference between largest and smallest digits: 2
321 is not a hamming number
```

```
Enter a number: 1024
Product of all digits: 0
Difference between largest and smallest digits: 4
1024 is a hamming number
```

Exercise 2: Day of A Week [50 marks]

January 1st of 1900 is a Monday. Write a program **dayofWeek.c** to read in a year and print out what day January 1st of that year is.

For example, if the given year is 1901, your program should print out "Tuesday" since January 1st of 1901 is a Tuesday.

You may assume that the given year is between 1900 and 2099 (both inclusive). No input validation is needed.

As a hint, a leap year lasts 366 days instead of usual 365.

You are not allowed to use array, string or any other structures that is yet covered in CS1010.

Three sample runs are shown below with user input highlighted in **bold**.

```
Enter year: 1901
Tuesday
```

```
Enter year: 1903
Thursday
```

```
Enter year: 1905
Sunday
```

=== END OF PAPER ===