Graded Tasks

1. Write a flowchart that prints ‘Hello World’ to the screen.
2. Write a flowchart that asks the user for their name and greets them with their name.
3. Modify the previous flowchart such that only the users Alice and Bob are greeted with their names.
4. Write a flowchart that asks the user for a number n and prints the sum of the numbers 1 to n
5. Write a flowchart that asks the user for a number n and gives them the possibility to choose between computing the sum and computing the product of 1,…, n.
6. Write a guessing game where the user has to guess a secret number. After every guess the program tells the user whether their number was too large or too small. At the end the number of tries needed should be printed. It counts only as one try if they input the same number multiple times consecutively.

7. Credit card numbers follow certain patterns. A credit card number must have between 13 and 16 digits. It must start with

   4 for Visa cards
   5 for Master cards
   37 for American Express cards
   6 for Discover cards

   In 1954, Hans Luhn of IBM proposed an algorithm for validating credit card numbers. The algorithm is useful to determine whether a card number is entered correctly or whether a credit card is scanned correctly by a scanner. All credit card numbers are generated following this validity check, commonly known as the Luhn check or the Mod 10 check, which can be described as follows (for illustration, consider the card number 4388576018402626):

   Step-1. Double every second digit from right to left. If doubling of a digit results in a two-digit number, add up the two digits to get a single-digit number.

   2*2=4
   2*2=4
   4*2=8
   1*2=2
   6*2=12 (1+2= 3)
   5*2=10 (1+0=1)
   8*2=16 (1+6=7)
   4*2=8

   Step-2. Now add all single-digit numbers from Step 1.

   4+4+8+2+3+1+7+8 = 37

   Step-3. Add all digits in the odd places from right to left in the card number.
   6+6+0+8+7+8+3=38

Step-4. Sum the results from Step 2 and Step 3;
37+38=75

Step-5. If the result from Step 4 is divisible by 10, the card number valid; otherwise, it is invalid. For example, the number 4388576018402626 is invalid, but the number 4388576018410707 is valid.

Draw a flowchart that prompts the user to enter a credit card number as an integer. Display whether the number is valid or invalid.