Swinburne University of Technology Sarawak

COS10009 Introduction to Programming

Control Flow, Arithmetic Expression & Operators – (Lab 02)

Pass Task 2.1 Name Tester

Task: Create a program that tests a user's name and echoes a custom message. Control flow enables you to easily add conditions and loops to your programs. In this task you will create a small program that uses conditions and loops to output custom messages to users.

To Do: Create a small program that will check the user's name and respond with different messages for different people.

Implement a main procedure with the following logic:

■ It reads a name from the user, and displays back a message.
■ Check if the name entered is your name.
■ If the name is your name, output the message 'Awesome name!'
■ Otherwise output the silly name message:

Steps:
1: Prompt 'Please enter your name: ', and assign the input to a variable
2: if name is equal to your name or your tutor's name then output ' is an awesome name' otherwise:
3. Print '– the name entered – is a silly name!'.

Saying that the name is 'silly' will have a much greater effect if we add lots of 'silly's to the out- put... Add a loop structure to repeat the word 'silly' 60 times. This will output the person's name and ' is a silly silly' ... with 60 'silly's, then ' name'.

Steps:
1: Make i equal 0
2: Print (staying on the same line) name, ' is a'
3: While i is less than 60
4: Print (on the same line) ' silly'
5: Increment i (make i equal i + 1)
6: Output ' name!' (moving to a new line)

Note: Indentation is important! It helps the reader to have a better view on instructions that sit inside the loop against instructions sit outside the while loop.

Hint: Append "\n" to the end of your string to print to a new line.

Pass Task 2.2 Calculating Leap Year

Task: Write a C program to determine whether a year entered by user is a leap year.

To Do:

In the Gregorian calendar, a normal year consists of 365 days. Because the actual length of a sidereal year (the time required for the Earth to revolve once about the Sun is actually 365.25635 days, a "leap year" of 366 days is used once every four years to eliminate the error caused by three normal (but short years. Any year that is evenly divisible by 4 is a leap year: for example, 1988, 1992, and 1996 are leap years.

However, there is still a small error that must be accounted for. To eliminate this error, the Gregorian calendar stipulates that a year that is not evenly divisible by 100 or divisible by 400 is a leap year.

For this reason, the following years are not leap years:
1700, 1800, 1900, 2100, 2200, 2300, 2500, 2600

Steps:
1: Prompt: 'Please enter year: ', then assign the input into a variable.
2: Use modulus operator (% to find remainder when the input is divided by 4, 100 and 400, store the results in separate variables.

3. Use a nested 'if else' structure to determine if the input is a leap year.
4. Prompt: 'Repeat for another input?', repeat step (1) if the user enters 'Yes', else end the program.

Once you have completed your program, perform the following desk check:

Test Data:

	First data set	Second data set	Third data set	Forth data set
Year	1996	2000	1900	2019

Expected Result:

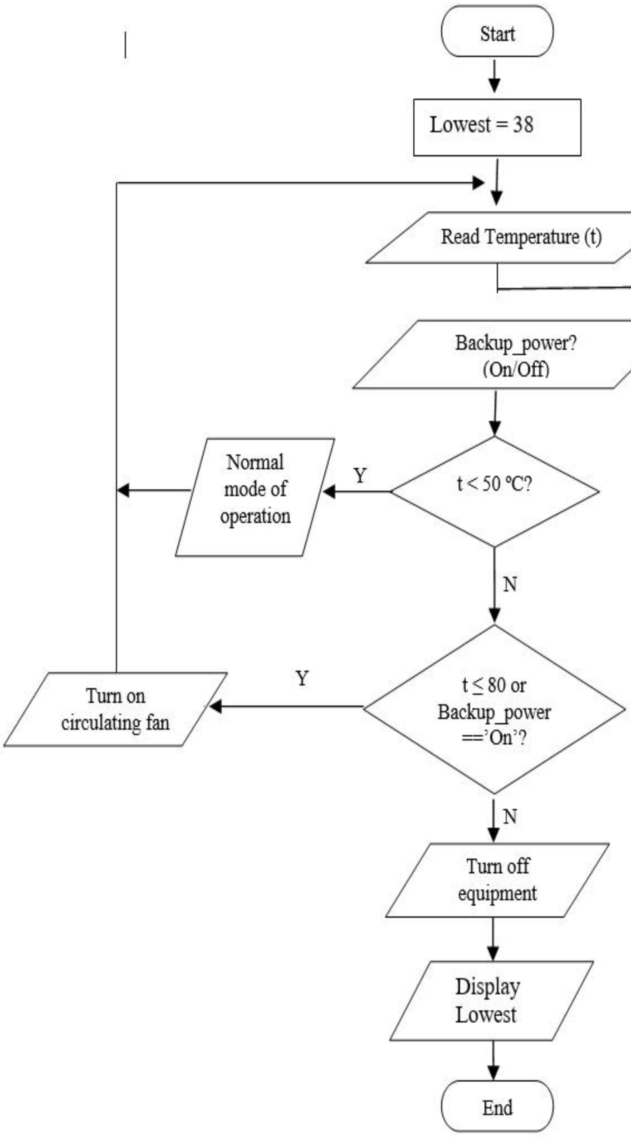
	First data set	Second data set	Third data set	Forth data set
Output	Leap Year	Leap Year	Not a Leap Year	Not a Leap Year

Pass Task 2.3 Machinery Temperature Management

Task: Write a program that manages the temperature reading from a sensor inside a large piece of machinery

To Do

The control flow of managing the temperature reading of a machine is shown in the flowchart below. Write a C program to implement process flow stated in the flow chart.



Note: Kindly ensure you have performed a proper desk check before submitting your program.

Credit Task 2.1 Fibonacci Series and Prime Number

Task: Develop a C program that prints the numbers of terms of Fibonacci series together with prime number as requested by the user.

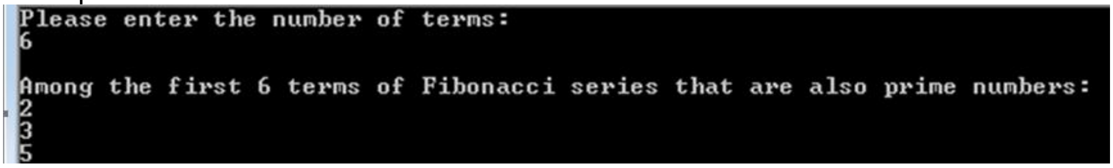
To Do

Prompt user to enter the number of terms, then use the input to construct a loop structure that generates Fibonacci series, but only those Fibonacci numbers which are also prime numbers will be printed to the screen as the final output.

Fibonacci numbers are numbers in Fibonacci series. First few numbers of series are 0, 1, 1, 2, 3, 5, 8 etc, except first two terms in sequence every other term is the sum of two previous terms, For example 8 = 3 + 5 (addition of 3, 5. This sequence has many applications in mathematics and Computer Science.

Prime numbers is a whole number greater than 1, whose only two whole-number factors are 1 and itself. The first few prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, 23, and 29.

Example of execution:



```
Please enter the number of terms:
6

Among the first 6 terms of Fibonacci series that are also prime numbers:
2
3
5
```