**A logo of a globe with yellow rings around it

Description automatically generated**

**GROUP ASSIGNMENT**

**TECHNOLGY PARK MALAYSIA**

**CT111-3-2-COMT**

**COMPUTING THEORY**

**APD2F2309CS(CYB)**

**HANDOUT DATE:  16 OCTOBER 2023**

**HAND-IN DATE:  23 December 2023**

**WEIGHTAGE: 40%**

|  |  |
| --- | --- |
| **NAME:** | **TP NUMBER:** |
| **YAZEN ABOBAKR AHMED AL-MEHDHAR** | **TP069210** |
| **ABDULRAHMAN GAMIL MOHAMMED AHMED** | **TP071012** |
| **ABDULELAH HUSSEIN ABDULRAHMAN AL-KAF** | **TP069319** |
| **HAZIQ IRFAN BIN RADZALI** | **TP072306** |

PART B

[**1.** **Program Code Complexities:** 3](#_Toc154016065)

[**2.** **Computability:** 5](#_Toc154016066)

[**3.** **REFERNCES:** 6](#_Toc154016067)

**PART B:**

# **Program Code Complexities:**

The provided code snippet, as demonstrated in part A, repeatedly prompts the user for input and store acceptable numbers in a list between 1 and 5, ensuring that every number entered is greater than or equal to the one before it. When there are five numbers in the list, the loop ends, and the list of numbers is printed in ascending order.

**Best Case Complexity:**

In the best-case scenario, the user enters five valid numbers in ascending order and there are no invalid inputs. In this case, the loop iterates exactly five times, performing constant-time operations on each iteration: input, comparison, and list appending. Consequently, the best-case time complexity is O(1) since it is not affected by the size of the input:

A screenshot of a computer program

Description automatically generated

**Worst Case Complexity:**

In the worst-case scenario, the loop keeps going until five valid numbers are entered because the user keeps entering invalid numbers (indefinitely). The loop may run more than five times in this scenario. In the worst scenario, the user keeps entering invalid numbers and never manages to enter a valid number within the range on the first try.

For example, the user might enter:

A screenshot of a computer program

Description automatically generated

In this worst-case scenario, the loop iterates ten times, twice for each valid input. Consequently, the worst-case time complexity is O(n), where n is the number of iterations required to collect five valid numbers.

# **Computability:**

Computability studies whether a problem can be solved by a computer or not A code snippet is considered computable if it can be executed successfully, will eventually terminate for all possible inputs, and will yield the desired results. The given code snippet can be computed because it terminates for all inputs and follows a predictable logic.

1. **Predictable Logic**: The logic in the code is obvious and dependable. It continuously requests input from the user, verifies the input, saves legitimate numbers, and applies conditions to the input to make sure it satisfies predetermined standards. This guarantees that the code is able to handle a wide range of inputs without crashing.
2. **Input Validation**: The code makes sure the user's input is valid by confirming that it is between 1 and 5. The code displays an error message and keeps asking for input if an invalid input is entered
3. **Loop Termination**: When the list of numbers contains exactly five valid numbers, the code terminates. This is a clear termination condition. This guarantees that, regardless of the user's input, the loop will eventually come to an end.
4. **Error Handling**: When the user enters invalid input, the code gracefully handles errors by displaying informative error messages. By handling errors, we make sure that invalid inputs won't cause the code to crash or yield unexpected results.
5. **Finite State Machine**: With clearly defined states (waiting for input, validating input, appending to the list, and verifying ascending order) and transitions between them, the code can be conceptualized as a finite state machine. Because of this methodical approach to input handling, the code can be computed.

# **REFERNCES:**

Jones, N. (1995, December 31). Computability and Complexity from a Programming Perspective (MFPS Draft preview). Retrieved December 20, 2023, from doc lagout website: https://doc.lagout.org/science/0\_Computer%20Science/4\_Theory%20of%20Computation/Computability%20and%20Complexity.pdf