



**School of Pre-University Studies**

**Foundation in Computing /Foundation**

**in Engineering**

**Introduction to Algorithm**

**(ITS30705)**

**Practical Assignment**

<b>Semester</b>	: August 2022
<b>Hand Out Date</b>	: 6 <sup>th</sup> September 2022 (Tuesday)
<b>Hand In Date</b>	: 14 <sup>th</sup> October 2022 (Friday 11:59 pm)
<b>Weightage</b>	: 10%
<b>Assignment Type</b>	: Individual

\*Academic impropriety:

Submitting the course work means you have agreed that your work is original and comply with the rules and regulations of Academic Impropriety.

Note: Copying, cheating, attempts to cheat, plagiarism, collusion, and any other attempts to gain an unfair advantage in assessment result in award 0 marks to all parties concerned.

## LEARNING OUTCOME

This assignment has been designed for students to:

- Demonstrate problem-solving skills using different sorting algorithms, randomized algorithms and bridge problem.
- Demonstrate practical skills in basic Python programming design by developing various control structures and algorithms such as selection structure, repetition structure, array and basic data structures.

## ASSIGNMENT DESCRIPTION

This assignment require students to:

- **Develop a console program using Python.**
- **Draw program's flowchart.**

Write a Python program to create a simple queue system to manage customer flow in a restaurant. There are four tables in the restaurant. The program must have all functions listed below:

1. Function to issue new ticket with new number auto increase by 1, start from 2001.
2. Function to assign first ticket in queue to selected table in the restaurant, from 1 to 4.

Upon starting the program, the program should display message as shown in the picture below:

*Output example*

```
Enter 0 to 5 for following options:
0 -> Issue new ticket number
1 -> Assign first ticket in queue to Table 1
2 -> Assign first ticket in queue to Table 2
3 -> Assign first ticket in queue to Table 3
4 -> Assign first ticket in queue to Table 4
5 -> Quit program

Tickets in queue: []
Table assignment: {'Table 1': 'Not assigned', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: |
```

When **0** is entered, a new ticket with increment number, start from 2001, is added to the queue, as shown in the picture below:

*Output example*

```
Tickets in queue: []
Table assignment: {'Table 1': 'Not assigned', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 0

Tickets in queue: [2001]
Table assignment: {'Table 1': 'Not assigned', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: |
```

Keep entering 0 to add more new tickets with increment number to the queue, as shown in the picture below:

### *Output example*

```
Tickets in queue: [2001]
Table assignment: {'Table 1': 'Not assigned', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 0

Tickets in queue: [2001, 2002]
Table assignment: {'Table 1': 'Not assigned', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 0

Tickets in queue: [2001, 2002, 2003]
Table assignment: {'Table 1': 'Not assigned', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 0

Tickets in queue: [2001, 2002, 2003, 2004]
Table assignment: {'Table 1': 'Not assigned', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: |
```

Enter 1 to 4 to assign first ticket in queue to respective table, then remove the ticket from queue, as shown in picture below:

### *Output example*

```
Tickets in queue: [2001, 2002, 2003]
Table assignment: {'Table 1': 'Not assigned', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 0

Tickets in queue: [2001, 2002, 2003, 2004]
Table assignment: {'Table 1': 'Not assigned', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 1

Tickets in queue: [2002, 2003, 2004]
Table assignment: {'Table 1': '2001', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 2

Tickets in queue: [2003, 2004]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: |
```

Keep entering number from 0 to 4 to issue more ticket and assign ticket to respective table, as shown in the picture below:

### *Output example*

```
Tickets in queue: [2002, 2003, 2004]
Table assignment: {'Table 1': '2001', 'Table 2': 'Not assigned', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 2

Tickets in queue: [2003, 2004]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 0

Tickets in queue: [2003, 2004, 2005]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 0

Tickets in queue: [2003, 2004, 2005, 2006]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': 'Not assigned', 'Table 4': 'Not assigned'}
Enter your option: 3

Tickets in queue: [2004, 2005, 2006]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': 'Not assigned'}
Enter your option: 4

Tickets in queue: [2005, 2006]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': '2004'}
Enter your option: 0
```

New ticket will replace the existing ticket number if assigned to same table, as shown in the picture below:

#### *Output example*

```
Tickets in queue: [2004, 2005, 2006]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': 'Not assigned'}
Enter your option: 4

Tickets in queue: [2005, 2006]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': '2004'}
Enter your option: 0

Tickets in queue: [2005, 2006, 2007]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': '2004'}
Enter your option: 1

Tickets in queue: [2006, 2007]
Table assignment: {'Table 1': '2005', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': '2004'}
Enter your option: |
```

If number other than 0 to 5 is enter, prompt message “Invalid option, try again...”, as shown in the picture below:

#### *Output example*

```
Tickets in queue: [2005, 2006, 2007]
Table assignment: {'Table 1': '2001', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': '2004'}
Enter your option: 1

Tickets in queue: [2006, 2007]
Table assignment: {'Table 1': '2005', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': '2004'}
Enter your option: 8

Invalid option, try again...
Tickets in queue: [2006, 2007]
Table assignment: {'Table 1': '2005', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': '2004'}
Enter your option: seven

Invalid option, try again...
Tickets in queue: [2006, 2007]
Table assignment: {'Table 1': '2005', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': '2004'}
Enter your option: |
```

Enter 5 to quit the program, as shown in picture below:

#### *Output example*

```
Invalid option, try again...
Tickets in queue: [2006, 2007]
Table assignment: {'Table 1': '2005', 'Table 2': '2002', 'Table 3': '2003', 'Table 4': '2004'}
Enter your option: 5

Quitting program...

○ (base) chempakaseri@Chempakas-MacBook-Pro ~ %
```

When you save the program, name the Python program file as “YourName\_RestaurantQueueSystem.py”. Draw a flowchart to describe the complete program flow.

## SUBMISSION INSTRUCTION

1. Compress the Python program file (.py) and the flowchart file (.doc, .pdf, or standard image file) into one zip file and name as "ITS30705\_YourName\_PracticalAssignment.zip".
2. Submit the zip file into MyTiMES portal BEFORE THE DUE DATE specified in this document's cover page.
3. After the submission, double check again the file upload in submission page to make sure that latest version of the zip file is uploaded successfully in TiMES portal.

### Important Notes:

- a) Emailed assignment will not be accepted.
- b) Late submissions will be penalized as per school policy.
- c) Each student is expected to contribute significantly to all deliverables as the assignment is a joint effort. In the event where a student's contribution is grossly unequal, marks shall be deducted and awarded to a group member who has done the work of his teammate (if applicable)

## MARKING RUBRIC

Marking Criteria	Outstanding (10 – 9)	Good (8 – 6)	Average (5 – 3)	Poor (2 – 0)
Program Function (1)				
Program Function (2)				
Program Control Structure				
Program Logic				
Flowchart				
Free of Program's Bugs				
<b>TOTAL MARKS (60 * 10%)</b>				