

Activity No. 5	
QUEUES	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: 10 / 07 / 2024
Section: CPE21S4	Date Submitted: 10 / 07 / 2024
Name(s): ROALLOS, Jean Gabriel Vincent G.	Instructor: Prof. Maria Rizette Sayo

6. Output

```
Current elements in the queue:
Adia
Bona
Bonifacio
Cabilan
Carag

classRec.empty() : 0
classRec.size() : 5
classRec.front() : Adia
classRec.back() : Carag
classRec.pop() :

Bona
Bonifacio
Cabilan
Carag

...Program finished with exit code 0
Press ENTER to exit console.
```

Table 5-1. Queues using C++ STL

```
deque(head, 50);

10 20 30 40 50
40 50

...Program finished with exit code 0
Press ENTER to exit console.
```

Table 5-2. Queues using Linked List Implementation

```

10 added to the queue.
20 added to the queue.
30 added to the queue.
40 added to the queue.
50 added to the queue.
Front element: 10
Current size: 5
Queue elements: 10 20 30 40 50
10 removed from the queue.
20 removed from the queue.
Front element after two dequeues: 30
Current size after dequeues: 3
Queue elements: 30 40 50
60 added to the queue.
Front element after adding 60: 30
Queue elements: 30 40 50 60

...Program finished with exit code 0
Press ENTER to exit console.

```

Table 5-3. Queues using Array Implementation

## 7. Supplementary Activity

```

C/C++
#include <iostream>
#include <string>
using namespace std;

class Job {
private:
    int id;
    string username;
    int pages;

public:
    Job(int id, string username, int pages) {
        this->id = id;
        this->username = username;
        this->pages = pages;
    }

    int getId() {
        return id;
    }

    string getUserName() {
        return username;
    }
}

```

```

        int getPages() {
            return pages;
        }
};

class Printer {
private:
    Job** queue;
    int capacity;
    int size;
    int front;
    int rear;

public:
    Printer(int capacity) {
        this->capacity = capacity;
        this->size = 0;
        this->front = 0;
        this->rear = 0;
        queue = new Job*[capacity];
    }

    ~Printer() {
        for (int i = 0; i < size; i++) {
            delete queue[(front + i) % capacity];
        }
        delete[] queue;
    }

    void addJob(int id, string username, int pages) {
        if (size == capacity) {
            cout << "Printer queue is full.\n";
            return;
        }
        queue[rear] = new Job(id, username, pages);
        rear = (rear + 1) % capacity;
        size++;
    }

    void processJobs() {
        if (size == 0) {
            cout << "No jobs to process.\n";
            return;
        }
        for (int i = 0; i < size; i++) {
            Job* job = queue[(front + i) % capacity];
            cout << "Processing job " << job->getId()
                 << " for " << job->getPages()
                 << " pages by " << job->getUserName() << ".\n";
        }
        for (int i = 0; i < size; i++) {
            delete queue[(front + i) % capacity];
        }
        size = 0;
        front = 0;
    }
};

```

```

        rear = 0;
    }
};

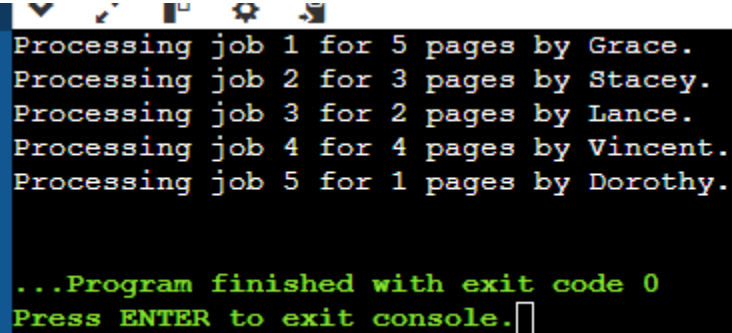
int main() {
    Printer printer(5);

    printer.addJob(1, "Grace", 5);
    printer.addJob(2, "Stacey", 3);
    printer.addJob(3, "Lance", 2);
    printer.addJob(4, "Vincent", 4);
    printer.addJob(5, "Dorothy", 1);

    printer.processJobs();

    return 0;
}

```



```

Processing job 1 for 5 pages by Grace.
Processing job 2 for 3 pages by Stacey.
Processing job 3 for 2 pages by Lance.
Processing job 4 for 4 pages by Vincent.
Processing job 5 for 1 pages by Dorothy.

...Program finished with exit code 0
Press ENTER to exit console.

```

## 8. Conclusion

I have deepened my understanding with queues and how to give its traits to linked lists and arrays. I was also reviewed about the different operations that occur with queues, as it was also included in the previous laboratory work. As for the supplementary activity, I think I have done well, but I took a while to figure out that I have to create a syntax for myself to create the necessary operations in the experiment.

## 9. Assessment Rubric