LAB TASK 4:

Name: Saif Majid Khan

SAP-ID: 57114

CS3-1

Data Structures.

```
#include <iostream>
#include <string>
using namespace std;

class Queue {
private:
    char* arr;
    int front;
    int rear;
    int capacity;
    int count;
```

```
public:

    Queue(int size = 100) {
        arr = new char[size];
        capacity = size;
        front = 0;
        rear = -1;
        count = 0;
    }

    Queue() {
        delete[] arr;
    }
```

```
void enqueue(char element) {
        if (count == capacity) {
            cout << "Queue is full!" << endl;
            return;
        }
        rear = (rear + 1) % capacity;
        arr[rear] = element;
        count++;
    }

    char dequeue() {
        if (isEmpty()) {
            cout << "Queue is empty!" << endl;
            return '\0';
        }
```

```
char element = arr[front];
        front = (front + 1) % capacity;
        count--;
        return element;
    }

    bool isEmpty() {
        return count == 0;
    }

    void display() {
        if (isEmpty()) {
            cout << "Queue is empty!" << endl;
            return;
        }
```

```
 int f = front;
        for (int i = 0; i < count; i++) {
            cout << arr[f] << " ";
            f = (f + 1) % capacity;
        }
        cout << endl;
    }

    void concatenate(Queue& q) {
        while (!q.isEmpty()) {
            enqueue(q.dequeue());
        }
    }
};
```

```
void createQueuesAndConcatenate(string input) {
    Queue finalQueue(500);
    Queue wordQueue;

    string word = "";
    for (char ch : input) {
        if (ch != ' ') {
            wordQueue.enqueue(ch);
        } else {
            wordQueue.display();
            finalQueue.concatenate(wordQueue);
            wordQueue = Queue();
        }
    }
```

```
 if (!wordQueue.isEmpty()) {
        wordQueue.display();
        finalQueue.concatenate(wordQueue);
    }

    cout << "Concatenated Queue: ";
    finalQueue.display();
}

int main() {
    string input;
    cout << "Enter a string: ";
    getline(cin, input);

    createQueuesAndConcatenate(input);

    return 0;
}
```


input
Enter a string: 3
3
Concatenated Queue: 3
...Program finished with exit code 0
Press ENTER to exit console.