1. Write a menu-driven program to perform different operations with queue such a Enqueue ( ), Dequeue( ) and Display Queue( ).

#include <iostream>

using namespace std;

define SIZE 5

class Queue {

int arr[SIZE];

int front, rear;

public:

Queue() {

front = -1;

rear = -1;

}

void Enqueue(int value) {

if (rear == SIZE - 1) {

cout << "Queue is Full!\n";

} else {

if (front == -1) front = 0;

arr[++rear] = value;

cout << value << " added to queue.\n";

}

}

void Dequeue() {

if (front == -1 || front > rear) {

cout << "Queue is Empty!\n";

} else {

cout << arr[front] << " removed from queue.\n";

front++;

}

}

void Display() {

if (front == -1 || front > rear) {

cout << "Queue is Empty!\n";

} else {

cout << "Queue elements: ";

for (int i = front; i <= rear; i++) {

cout << arr[i] << " ";

}

cout << endl;

}

}

};

int main() {

Queue q;

int choice, value;

do {

cout << "\n=== Queue Menu ===\n";

cout << "1. Enqueue\n";

cout << "2. Dequeue\n";

cout << "3. Display\n";

cout << "4. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter value: ";

cin >> value;

q.Enqueue(value);

break;

case 2:

q.Dequeue();

break;

case 3:

q.Display();

break;

case 4:

cout << "Exiting program...\n";

break;

default:

cout << "Invalid choice!\n";

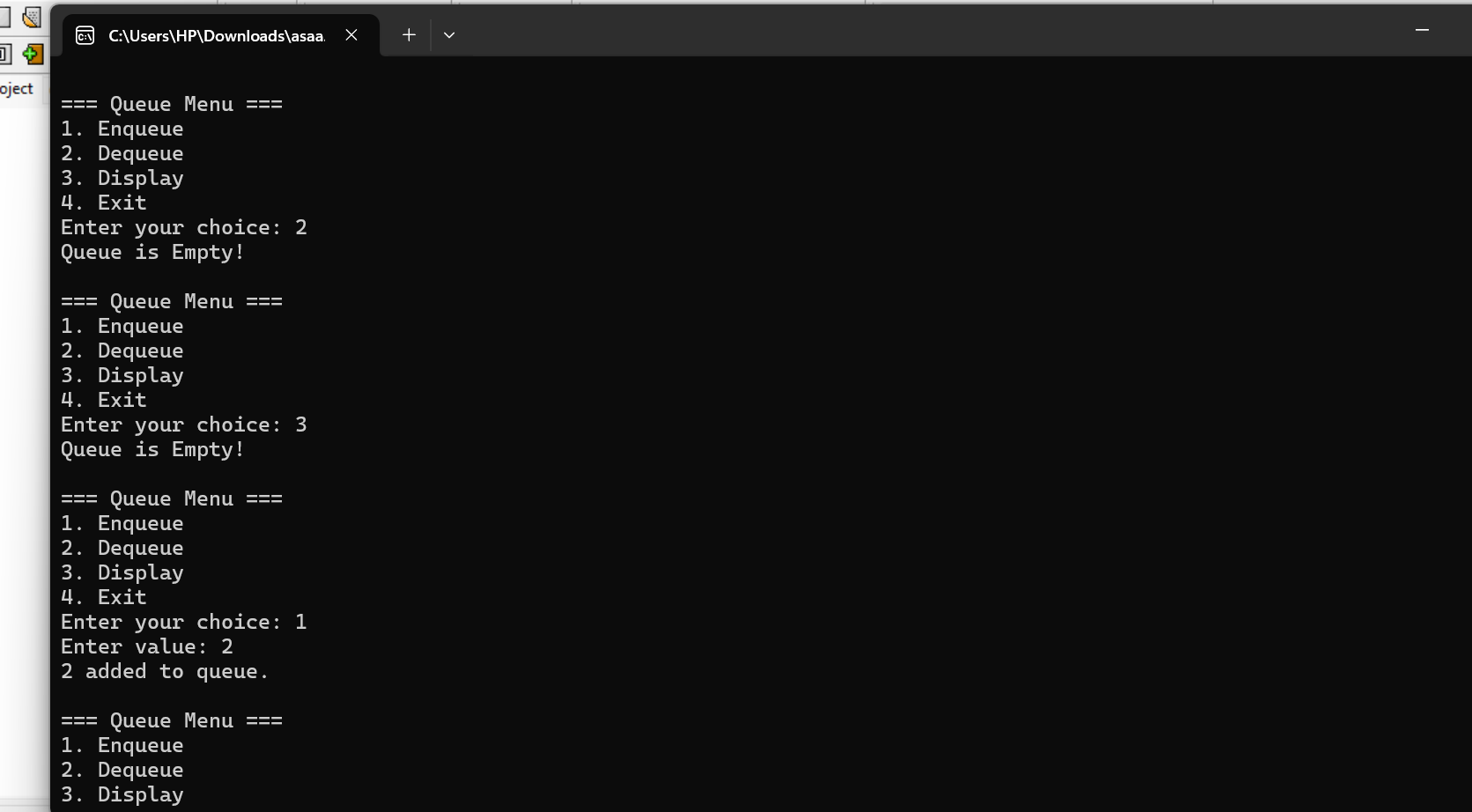
}

} while (choice != 4);

return 0;

}

OUTPUT



1. In this Exercise, you have to take a single string as input. Using this input string, you have to create multiple queues in which each queue will comprise of separate word appearing in the input string. Example: String = “Data Structure and Algo” Q1 = D → a → t → a Q2 = S → t → r → u → c → t → u → r → e Q3 = a → n → d

Q4 = A → l → g → o

#include <iostream>

#include <queue>

#include <sstream>

#include <string>

using namespace std;

int main() {

string input;

cout << "Enter a string: ";

getline(cin, input);

istringstream ss(input);

string word;

int queueNumber = 1;

while (ss >> word) {

queue<char> charQueue;

for (int i = 0; i < word.size(); i++) {

charQueue.push(word[i]);

}

cout << "Q" << queueNumber << " = ";

while (!charQueue.empty()) {

cout << charQueue.front();

charQueue.pop();

if (!charQueue.empty()) {

cout << " -> ";

}

}

cout << endl;

queueNumber++;

}

return 0;

}

OUTPUT

