

**Laporan Praktikum**

**Algoritma dan Struktur Data**

Ganjil 2023/2024

Program Studi Teknik Informatika

Institut Teknologi Sumatera

|  |  |
| --- | --- |
| **Modul :** | **Stack Queue** |
| **Nama :** | **Rayhan Fadel Irwanto** |
| **NIM :** | **122140236** |
| **Kelas (Kelas Asal) :** | **ASD RA** |

Instruksi sederhana :

* Disarankan kepada **Praktikan Algoritma Struktur Data** untuk mengeditnya menggunakan Google Docs agar tidak berantakan dan rapi,
* Silahkan mengganti **Nama Modul** baik yang ada pada **Cover** dan **Header** sesuai dengan materi praktikum,
* Gunakan text styling seperti **Heading 1**, **Normal Text** yang telah terformat / Text Style lainnya yang digunakan untuk menjaga estetika laporan,
* Gunakan [**Syntax Highlighter**](https://highlight.hohli.com/index.php)untuk merapikan kode yang sudah Praktikan buat ke dalam Laporan Praktikum.

**Materi Praktikum**

Stack Queue

**Link repl.it Source Code**

<https://replit.com/@RayhanFadel/queue-TM4>

**Source Code**

Gunakan [**Syntax Highlighter**](https://highlight.hohli.com/index.php)untuk merapikan Source Code yang dipindahkan dari text editor anda ke dokumen ini.

1. #include <cctype> // for character type checking
2. #include <iostream>
3. using namespace std;
5. **const int MAX\_SIZE = 100; // Maximum queue size**
7. struct Queue {
8. int queueArray[MAX\_SIZE];
9. int front, rear;
11. Queue() {
12. front = -1;
13. rear = -1;
14. }
16. bool isEmpty() { return front == -1 && rear == -1; }
18. bool isFull() { return rear == MAX\_SIZE - 1; }
20. **void enqueue(int value) {**
21. if (isFull()) {
22. cout << "Queue is full. Cannot add more elements." << endl;
23. return;
24. }
26. if (isEmpty()) {
27. front = rear = 0;
28. } else {
29. rear = (rear + 1) % MAX\_SIZE;
30. **}**
32. queueArray[rear] = value;
33. }
35. **int dequeue() {**
36. if (isEmpty()) {
37. cout << "Queue is empty. Cannot remove elements." << endl;
38. return -1;
39. }
41. int dequeuedValue = queueArray[front];
43. if (front == rear) {
44. front = rear = -1;
45. **} else {**
46. front = (front + 1) % MAX\_SIZE;
47. }
49. return dequeuedValue;
50. **}**
52. void printQueue() {
53. if (isEmpty()) {
54. cout << "Queue is empty." << endl;
55. **return;**
56. }
58. cout << "Queue contents:" << endl;
60. **int i = front;**
61. while (i != rear) {
62. cout << queueArray[i] << " -> index " << i << " (" << char(queueArray[i])
63. << ")" << endl;
64. i = (i + 1) % MAX\_SIZE;
65. **}**
66. cout << queueArray[rear] << " -> index " << i << " ("
67. << char(queueArray[rear]) << ")" << endl;
68. }
69. };
71. int main() {
72. string input;
73. Queue queue;
75. **cout << "Enter a string: ";**
76. cin >> input;
78. for (int i = 0; i < input.length(); i++) {
79. char c = input[i];
81. if (islower(c)) {
82. queue.enqueue(int(c));
83. cout << "ENQUEUE: " << int(c) << " (" << c << ")" << endl;
84. } else if (c == '-') {
85. **int dequeuedValue = queue.dequeue();**
87. if (dequeuedValue != -1) {
88. cout << "DEQUEUE: " << dequeuedValue << " (" << char(dequeuedValue)
89. << ")" << endl;
90. **}**
91. }
92. }
94. queue.printQueue(); // Display remaining characters in the queue
95. **return 0;**
96. }

**Dokumentasi Hasil Running**

|  |
| --- |
|  |
| **Gambar 1.** Output Code Queue Tugas Minggu 4.  Code di atas adalah implementasi dari struktur data antrian (queue) dalam C++. Antrian direpresentasikan melalui struktur 'Queue' yang menggunakan array 'queueArray' sebagai wadah elemen-elemen antrian. Struktur ini dilengkapi dengan fungsi-fungsi seperti 'isEmpty' dan 'isFull' untuk memeriksa apakah antrian kosong atau penuh, 'enqueue' untuk menambahkan elemen ke antrian, 'dequeue' untuk menghapus elemen dari antrian, serta 'printQueue' untuk menampilkan isi antrian.  Pada bagian utama program, pengguna diminta untuk memasukkan sebuah string. Selanjutnya, program akan mengecek setiap karakter dalam string tersebut. Jika karakter berupa huruf kecil, maka karakter tersebut akan dimasukkan ke dalam antrian menggunakan fungsi 'enqueue'. Sedangkan jika karakter merupakan tanda '-', program akan menghapus elemen dari antrian menggunakan fungsi 'dequeue'.  Seluruh elemen yang tersisa dalam antrian akan ditampilkan setelah proses penghapusan selesai dengan menggunakan fungsi 'printQueue'.  Dengan struktur data antrian ini, program dapat menambah atau menghapus elemen sesuai dengan aturan yang telah ditetapkan dan menampilkan isi antrian dengan jelas. |

|  |
| --- |
|  |
|  |