

Praktikum Algoritma Struktur Data

Praktikum 04: Overview Queue



Oleh:

Bintang Bimantara/ 5223600025

Program Studi Sarjana Terapan Teknologi Game

Departemen Teknologi Multimedia Kreatif

Politeknik Elektronika Negeri Surabaya

2024

1. Queue Menggunakan Linked List

Program ini mengimplementasikan struktur data Queue menggunakan Linked List.

Output

Clear

```
▲ /tmp/n7siP7F4yy.o
41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
   66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
   90 91 92 93 94 95 96 97 98 99 100

=== Code Execution Successful ===|
```

2. Enqueue pada Queue Linked List Program ini hanya memuat implementasi metode `enqueue` yaitu untuk menambahkan elemen baru pada akhir Queue Linked List.

```
File Edit Format View Help
void Queue::enqueue(int value) {
    Node* temp = new Node(value, nullptr);
    if (head == nullptr) {
        head = tail = temp;
    }
    else {
        tail->next = temp;
        tail = temp;
    }
    count++;
}
```

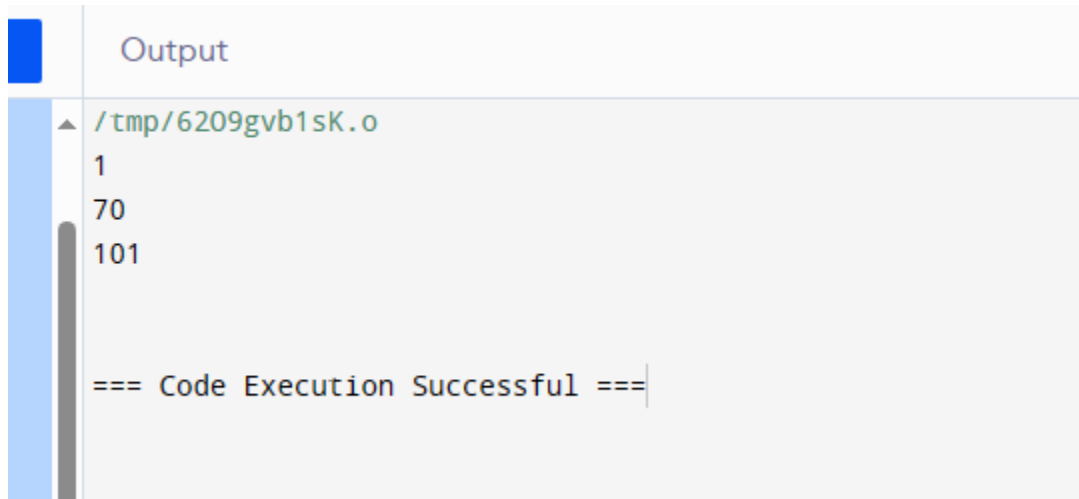
3. Dequeue pada Queue Linked List

Program ini hanya memuat implementasi metode `dequeue` yaitu menghapus elemen dari awal Queue Linked List.

```
File Edit Format View Help
int Queue::dequeue() {
    if (empty()) {
        throw invalid_argument("QueueEmptyException");
    }
    int value = head->value;
    Node* temp = head;
    head = head->next;
    delete temp;
    count--;
    return value;
}
```

4. Implementasi Queue menggunakan 2 Stack

Program ini mengimplementasikan struktur data Queue menggunakan dua buah Stack.



```
Output
/tmp/6209gvb1sK.o
1
70
101

=== Code Execution Successful ===
```

Tugas

1. Stack using a Queue

Pada permasalahan ini, kita harus mengimplementasikan struktur data Stack menggunakan struktur data Queue. Salah satu solusi yang diberikan adalah dengan menggunakan satu antrian (queue) saja.




```
Output Clear
/tmp/Wgp7dyEuyz.o
elemen atas: 3
element atas setelah pop: 2

=== Code Execution Successful ===
```

2. Reverse Stack

Pada permasalahan ini, kita harus mengimplementasikan struktur data Stack lalu hanya menambahkan rekursi yang dapat melakukan reverse isi data stack.

Output

Clear

▲

/tmp/9YVxPVS3T5.o

Stack sebelum pembalikan:
4 3 2 1
Stack setelah pembalikan:
1 2 3 4

=== Code Execution Successful ===