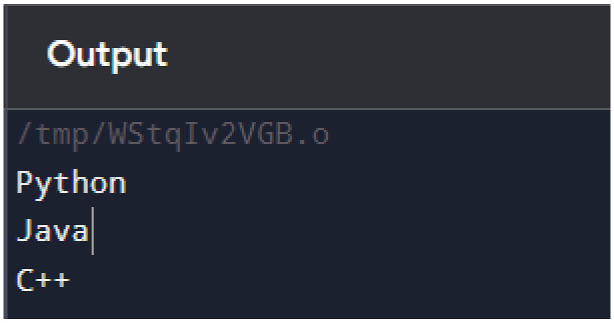
VIRTUCIO, DOMINIC JOSEPH P.
CPE 21S4

Stack and Queues Implementation

USING STACKS

C/C++

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

void display(stack<string> pl) {
    while (!pl.empty()) {
        cout << pl.top()<<endl;
        pl.pop();
    }
    cout << endl;
}

int main() {
    stack<string> pl;


    pl.push("C++");
    pl.push("Java");
    pl.push("Python");


    display(pl);

    return 0;
}
```

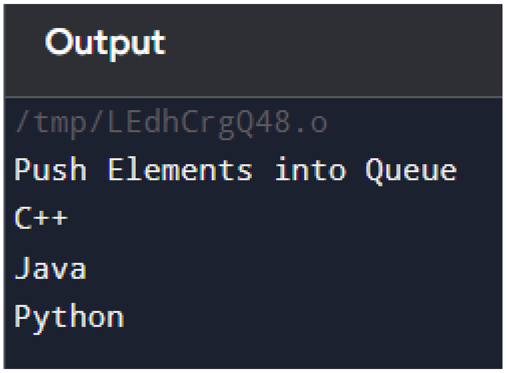


```
/tmp/WStqIv2VGB.o
Python
Java
C++
```

USING QUEUE

C/C++

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

void display(queue<string> pl) {
    while (!pl.empty()) {
        cout << pl.front()<<endl;
        pl.pop();
    }
    cout << endl;
}

int main() {
    queue<string> pl;

    cout << "Push Elements into Queue" << endl;
    pl.push("C++");
    pl.push("Java");
    pl.push("Python");

    display(pl);

    return 0;
}
```

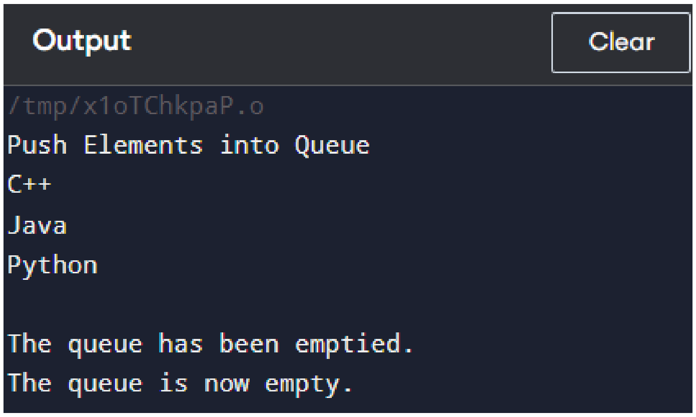


```
/tmp/LEdhCrgQ48.o
Push Elements into Queue
C++
Java
Python
```

EMPTYING THE QUEUE

C/C++

```cpp
#include <iostream>
#include <queue>
using namespace std;

void display(queue<string> pl) {
    if (pl.empty()) {
        cout << "The queue is empty." << endl;
    } else {
        while (!pl.empty()) {
            cout << pl.front() << endl;
            pl.pop();
        }
    }
    cout << endl;
}

void emptyQueue(queue<string>& pl) {
    while (!pl.empty()) {
        pl.pop();
    }
    cout << "The queue has been emptied." << endl;
}

int main() {
    queue<string> pl;

    cout << "Push Elements into Queue" << endl;
    pl.push("C++");
    pl.push("Java");
    pl.push("Python");

    display(pl);
    emptyQueue(pl);

    if (pl.empty()) {
        cout << "The queue is now empty." << endl;
    }

    return 0;
}
```

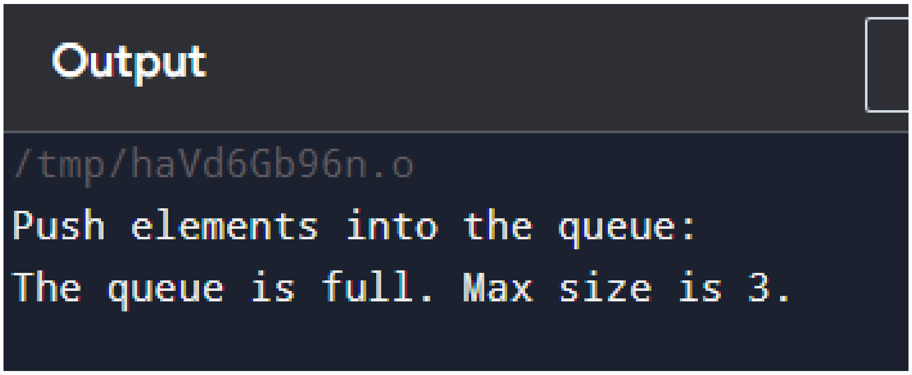


```
/tmp/x1oTChkpaP.o
Push Elements into Queue
C++
Java
Python

The queue has been emptied.
The queue is now empty.
```

PUSHING ELEMENTS INTO THE QUEUE BUT IT IS FULL:

C/C++

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

const size_t MAX_SIZE = 3;

void display(queue<string> pl) {
    size_t count = 0;
    while (!pl.empty() && count < MAX_SIZE) {
        cout << pl.front() << endl;
        pl.pop();
        count++;
    }
    cout << endl;
}

bool isFull(queue<string>& pl) {
    return pl.size() >= MAX_SIZE;
}

int main() {
    queue<string> pl;
    cout << "Push elements into the queue:" << endl;

    pl.push("C++");
    pl.push("Java");
    pl.push("Python");
    pl.push("C#");

    if (!isFull(pl)) {
        display(pl);
    } else {
        cout << "The queue is full. Max size is 3." << endl;
    }

    return 0;
}
```



```
/tmp/haVd6Gb96n.o
Push elements into the queue:
The queue is full. Max size is 3.
```