

Stack

```
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;  
    cout << "Push elements into stack:" << endl;  
    //add items to the stack  
    pl.push("C++");  
    pl.push("Java");  
    pl.push("Python");  
    display(pl);  
  
    return 0;  
}
```

main.cpp	Output
<pre>1 2 #include <iostream> 3 #include <stack> 4 using namespace std; 5 6 void display(stack<string> pl) { 7 while (!pl.empty()) { 8 cout << pl.top() << endl; 9 pl.pop(); 10 } 11 cout << endl; 12 } 13 14 int main() { 15 stack<string> pl; 16 cout << "Push elements into stack:" << endl; 17 //add items to the stack 18 pl.push("C++"); 19 pl.push("Java"); 20 pl.push("Python"); 21 display(pl); 22 23 return 0; 24 } 25 26</pre>	<pre>/tmp/0tLZSS76Xv.o Push elements into stack: Python Java C++ === Code Execution Successful ===</pre>

Queue

```
C/C++

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

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

int main() {
    queue<string> q;
    cout << "Push elements into queue:" << endl;
    //Add items to the queue
    q.push("C++");
    q.push("Java");
    q.push("Python");

    display(q);

    return 0;
}
```

main.cpp	Run	Output
<pre>1 #include <iostream> 2 #include <queue> 3 using namespace std; 4 5 void display(queue<string> q) { 6 while (!q.empty()) { 7 cout << q.front() << endl; 8 q.pop(); 9 } 10 cout << endl; 11 } 12 13 int main() { 14 queue<string> q; 15 cout << "Push elements into queue:" << endl; 16 //Add items to the queue 17 q.push("C++"); 18 q.push("Java"); 19 q.push("Python"); 20 21 display(q); 22 23 return 0; 24 } 25</pre>		<pre>/tmp/RWJ2NH4764.o Push elements into queue: C++ Java Python === Code Execution Successful ===</pre>