Group 7
CPE200-CPE21S1
Esteron, Jenel F.
Fabreag, Patrick Kyel M.
Banania, Ariel Jr. T
Falco, Arvin Paul D.

```cpp
#include <iostream>
#include <vector>

using namespace std;
int input1, input2, input3;
class Node {
public:
    int dataval;
    Node* nextval;
    Node(int dataval = 0, Node* nextval = nullptr) : dataval(dataval), nextval(nextval) {}
};

class SLinkedList {
public:
    Node* headval;
    SLinkedList() : headval(nullptr) {}

    void listprint() {
        Node* printval = headval;
        while (printval != nullptr) {
            cout << printval->dataval << endl;
            printval = printval->nextval;
        }
    }
};

class Queue {
public:
    vector<int> queue;

    void enqueue(int item) {
        queue.push_back(item);
    }

```

```
    int dequeue() {
        if (queue.size() < 1) {
            return -1; // You can use any suitable value to indicate an empty queue
        }
        int front = queue.front();
        queue.erase(queue.begin());
        return front;
    }

    void display() {
        for (int item : queue) {
            cout << item << " ";
        }
        cout << endl;
    }

    int size() {
        return queue.size();
    }
};
void start(){
    int Choice;
    cout << "Enter Number of desired data structure: ";
    cin >> Choice;

    switch (Choice) {
    case 1:
    {
```

```
        {
            cout << "Array" << endl;
            vector<int> a;
            a.push_back(input1);
            a.push_back(input2);
            a.push_back(input3);
            cout << "Items in the array are: ";
            for (int item : a) {
                cout << item << " ";
            }
            cout << endl;
            break;
        }

        case 2:
        {
            cout << "Linkedlist" << endl;
            SLinkedList list;
            list.headval = new Node(input1);
            Node* e2 = new Node(input2);
            list.headval->nextval = e2;
            Node* e3 = new Node(input3);
            e2->nextval = e3;
            list.listprint();
            break;
        }

```

```
89      case 3:
90      {
91          cout << "Stack" << endl;
92          vector<int> stack;
93          stack.push_back(input1);
94          stack.push_back(input2);
95          stack.push_back(input3);
96          cout << "Initial stack" << endl;
97          for (int item : stack) {
98              cout << item << " ";
99          }
100         cout << endl;
101         cout << "Elements popped from stack:" << endl;
102         while (!stack.empty()) {
103             cout << stack.back() << endl;
104             stack.pop_back();
105         }
106         cout << "Stack after elements are popped is empty." << endl;
107         break;
108     }
109
110     case 4:
111     {
112         cout << "Queue" << endl;
113         Queue q;
114         q.enqueue(input1);
115         q.enqueue(input2);
116         q.enqueue(input3);
117         cout << "Items Inside the queue: ";
118         q.display();
119         cout << endl;
120         q.dequeue();
121         cout << "After removing an element: ";
```

```
122         q.display();
123         q.dequeue();
124         cout << "After removing an element: ";
125         q.display();
126         q.dequeue();
127         cout << "After removing an element: ";
128         q.display();
129         break;
130     }
131 
132     default:
133         cout << "Invalid choice" << endl;
134     }
135 }
136 int main() {
137     cout << "Enter 1st number : ";
138     cin >> input1;
139     cout << "Enter 2nd number : ";
140     cin >> input2;
141     cout << "Enter 3rd number : ";
142     cin >> input3;
143 
144     cout << input1 << endl;
145     cout << input2 << endl;
146     cout << input3 << endl;
147 
148     cout << "--Menu--" << endl;
149     cout << "1. Array" << endl;
150     cout << "2. Linkedlist" << endl;
151     cout << "3. Stack" << endl;
152     cout << "4. Queue" << endl;
153 
```

```
154     label1:
155     start();
156     char choice;
157     cout<<"Do you want to pick a different data structure?(Y/N): ";
158     cin>>choice;
159     if (choice == 'Y'){
160         goto label1;
161     }else if (choice == 'y'){
162         goto label1;
163     }else if (choice == 'N'){
164         return 0;
165     }else if (choice == 'n'){
166         return 0;
167     }
168     return 0;
169 }
```