“UNIVERSIDAD NACIONAL DE SAN AGUSTÍN”

FACULTAD DE INGENIERÍA,PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

CURSO:

Ciencias de la Computación - Grupo “B”

DOCENTE:

Enzo Edir Velásquez Lobatón

ALUMNO:

Fabricio Huaquisto Quispe

REPOSITORIO:

https://github.com/fhuaquisto21/EPCC-CCII

Arequipa - Perú

2022

COLAS

1. **node.h**

```
class Node {
  private:
    int value;
    Node* next;
  public:
    Node(int);
    ~Node();
    int getValue();
    Node* getNext();
    void setValue(int);
    void setNext(Node*);
};
```

2. **node.cpp**

```
#include "node.h"

Node::Node(int _value) {
  this->value = _value;
  this->next = nullptr;
}

Node::~Node() {}

int Node::getValue() {
  return this->value;
}

Node* Node::getNext() {
  return this->next;
}

void Node::setNext(Node* _next) {
  this->next = _next;
}

void Node::setValue(int _value) {
  this->value = _value;
}
```

3. **cola.h**

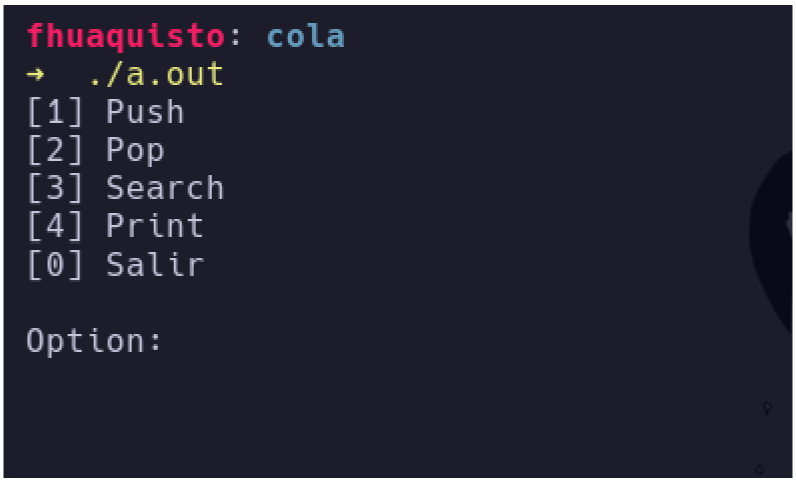
```
#include "node.cpp"

class Cola {
  private:
    Node* head;
  public:
    Cola();
    Cola(int);
    ~Cola();
    Node* push(int);
    Node* pop();
    Node* search(int);
    void printCola();
};
```

4. **pila.cpp**

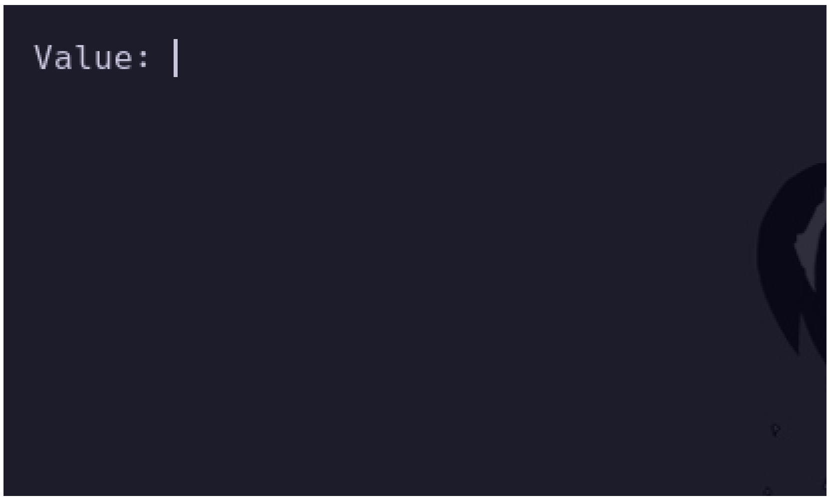
```
#include <iostream>
#include "cola.h"

Cola::Cola() {
  this->head = nullptr;
}

Cola::Cola(int _value) {
  Node* newNode = new Node(_value);
  this->head = newNode;
}

Cola::~Cola() {}

Node* Cola::push(int _value) {
  Node* newNode = new Node(_value);
  Node* currentNode = this->head;
  if (this->head == nullptr) {
    this->head = newNode;
    return this->head;
  }
  while (currentNode->getNext() != nullptr) {
    currentNode = currentNode->getNext();
  }
  currentNode->setNext(newNode);
  return currentNode;
}

Node* Cola::pop() {
  Node* auxNode = this->head;
  this->head = auxNode->getNext();
  delete auxNode;
  return this->head;
}

Node* Cola::search(int _index) {
  Node* currentNode = this->head;
  if (_index == 0) return currentNode;
  for (int i = 0; i < _index; ++i) {
    currentNode = currentNode->getNext();
    if (currentNode == nullptr) {
        return nullptr;
    }
  }
  return currentNode;
}

void Cola::printCola() {
  Node* currentNode = this->head;
  while (currentNode->getNext() != nullptr) {
    std::cout << currentNode->getValue() << " -> ";
    currentNode = currentNode->getNext();
  }
  std::cout << currentNode->getValue() << std::endl;
}
```

5. **main.cpp**

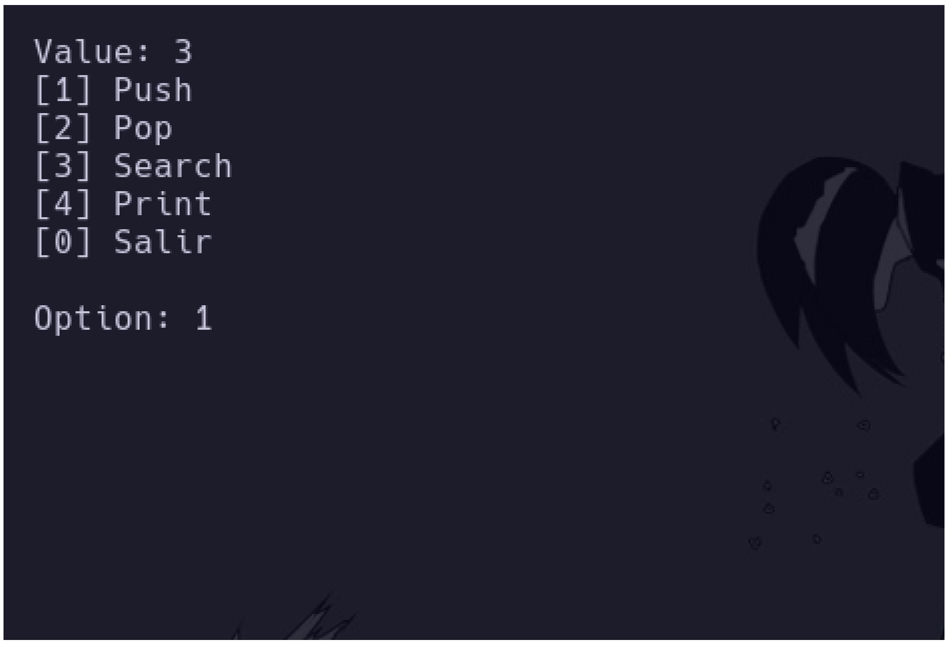
```
#include <iostream>
#include "cola.cpp"

void printMenu() {
  std::cout << "[1] Push" << std::endl;
  std::cout << "[2] Pop" << std::endl;
  std::cout << "[3] Search" << std::endl;
  std::cout << "[4] Print" << std::endl;
  std::cout << "[0] Salir" << std::endl;
  std::cout << std::endl << "Option: ";
}

int main() {
  Cola* cola = new Cola();
  int opt, index, value;
  do {
    printMenu();
    std::cin >> opt;
    printf("\e[1;1H\e[2J");
    switch (opt) {
      case 0:
        break;
      case 1:
        std::cout << "Value: ";
        std::cin >> value;
        cola->push(value);
        break;
      case 2:
        cola->pop();
        break;
      case 3:
        std::cout << "Index: ";
        std::cin >> index;
        std::cout << "El valor del nodo es: " <<
cola->search(index)->getValue() << std::endl;
        break;
      case 4:
        cola->printCola();
        break;
    }
  } while (opt != 0);

  return 0;
}
```



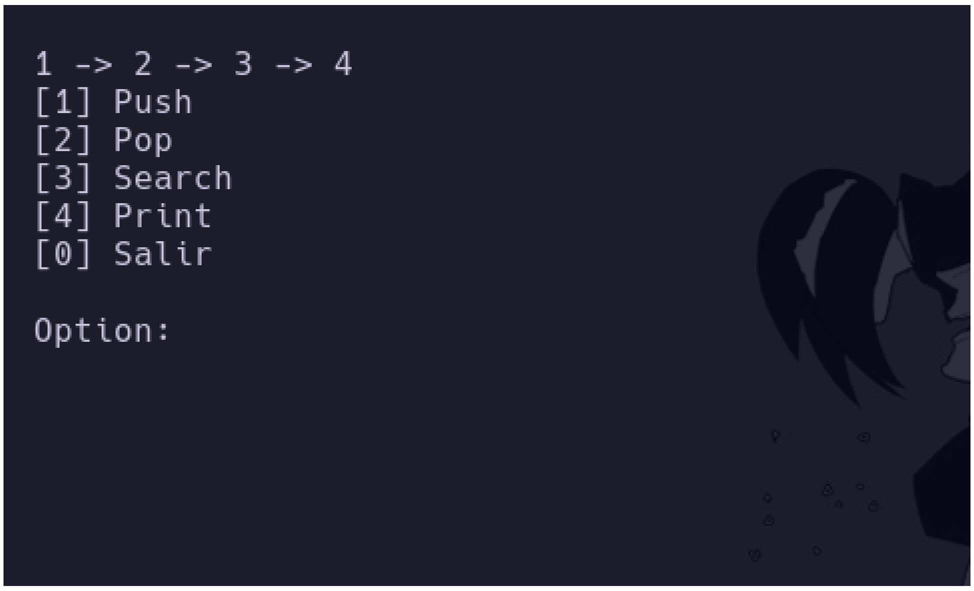
```
fhuaquisto: cola
➜  ./a.out
[1] Push
[2] Pop
[3] Search
[4] Print
[0] Salir

Option:
```



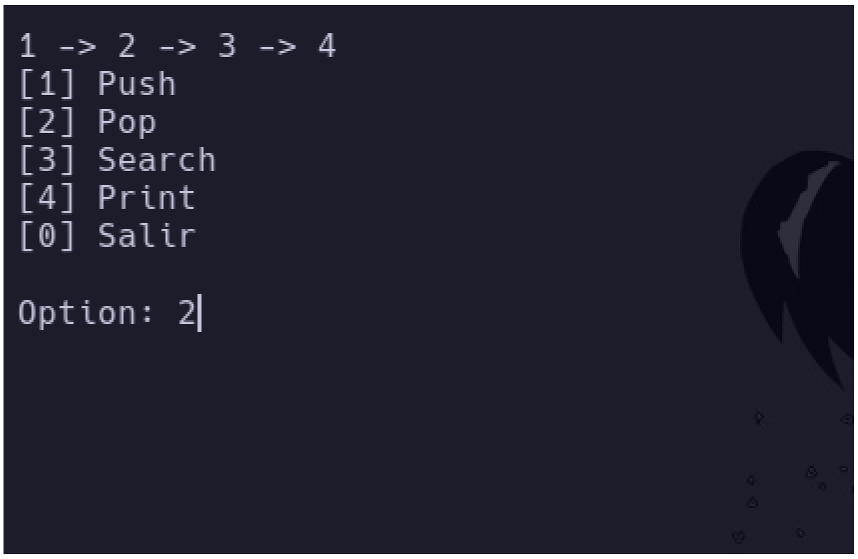
```
Value:
```



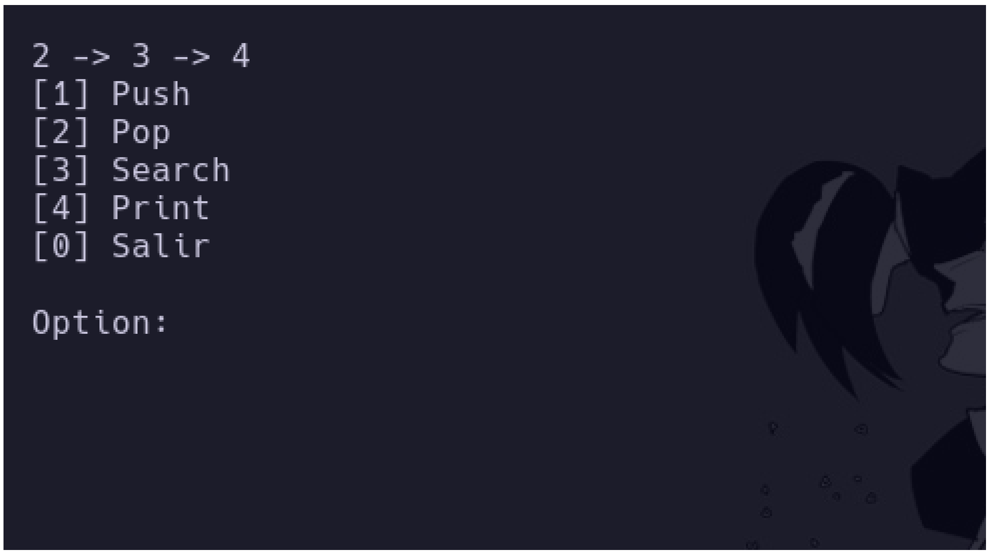
```
Value: 3
[1] Push
[2] Pop
[3] Search
[4] Print
[0] Salir

Option: 1
```



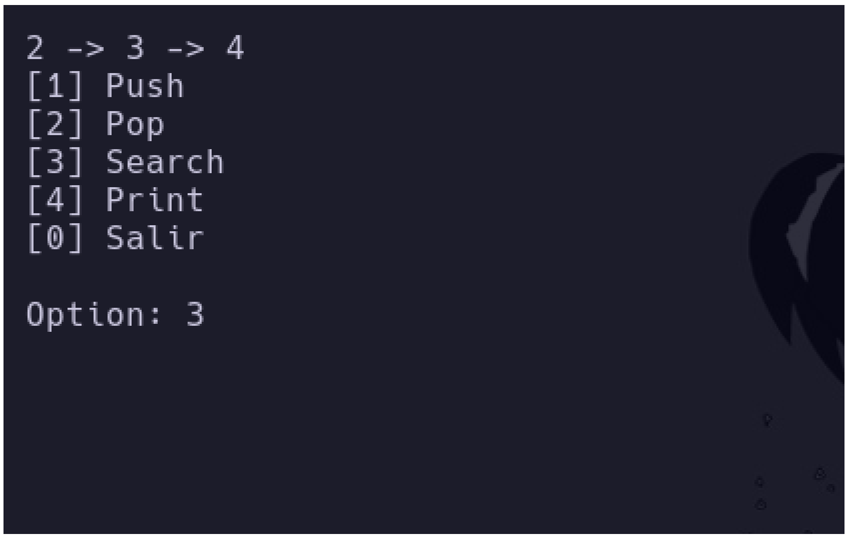
```
1 -> 2 -> 3 -> 4
[1] Push
[2] Pop
[3] Search
[4] Print
[0] Salir

Option:
```



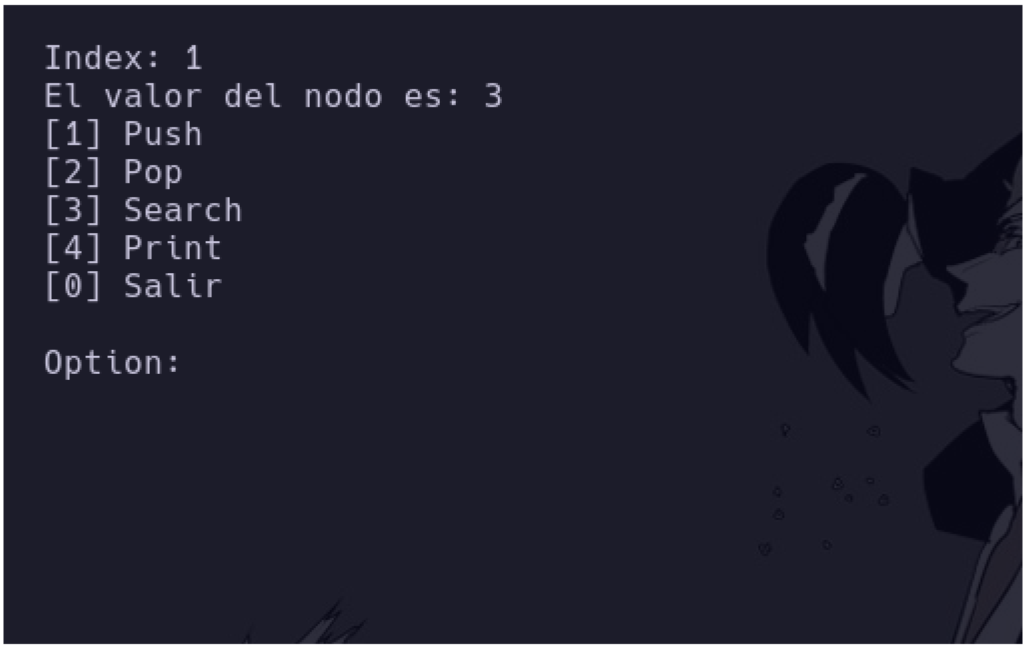
```
1 -> 2 -> 3 -> 4
[1] Push
[2] Pop
[3] Search
[4] Print
[0] Salir

Option: 2
```



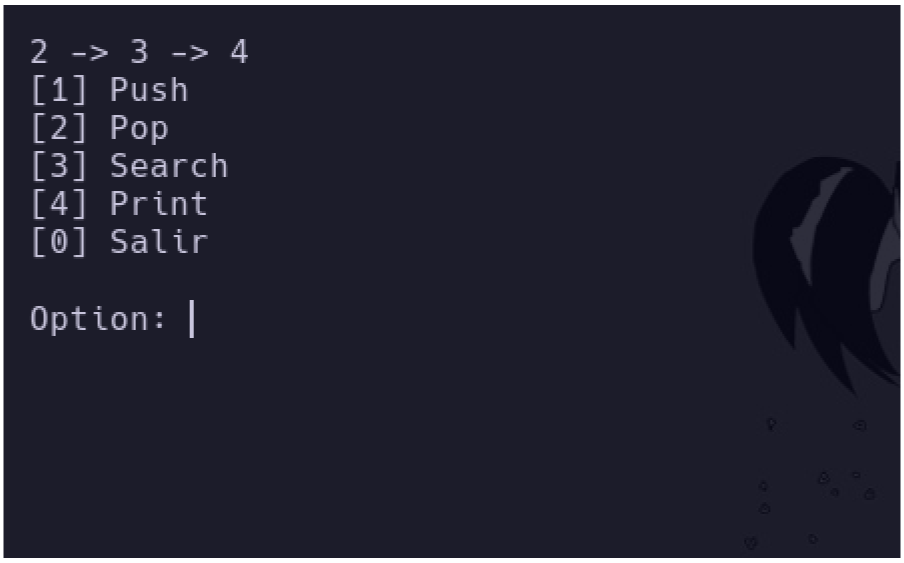
```
2 -> 3 -> 4
[1] Push
[2] Pop
[3] Search
[4] Print
[0] Salir

Option:
```



```
2 -> 3 -> 4
[1] Push
[2] Pop
[3] Search
[4] Print
[0] Salir

Option: 3
```



```
Index: 1
El valor del nodo es: 3
[1] Push
[2] Pop
[3] Search
[4] Print
[0] Salir

Option:
```



```
2 -> 3 -> 4
[1] Push
[2] Pop
[3] Search
[4] Print
[0] Salir

Option: |
```