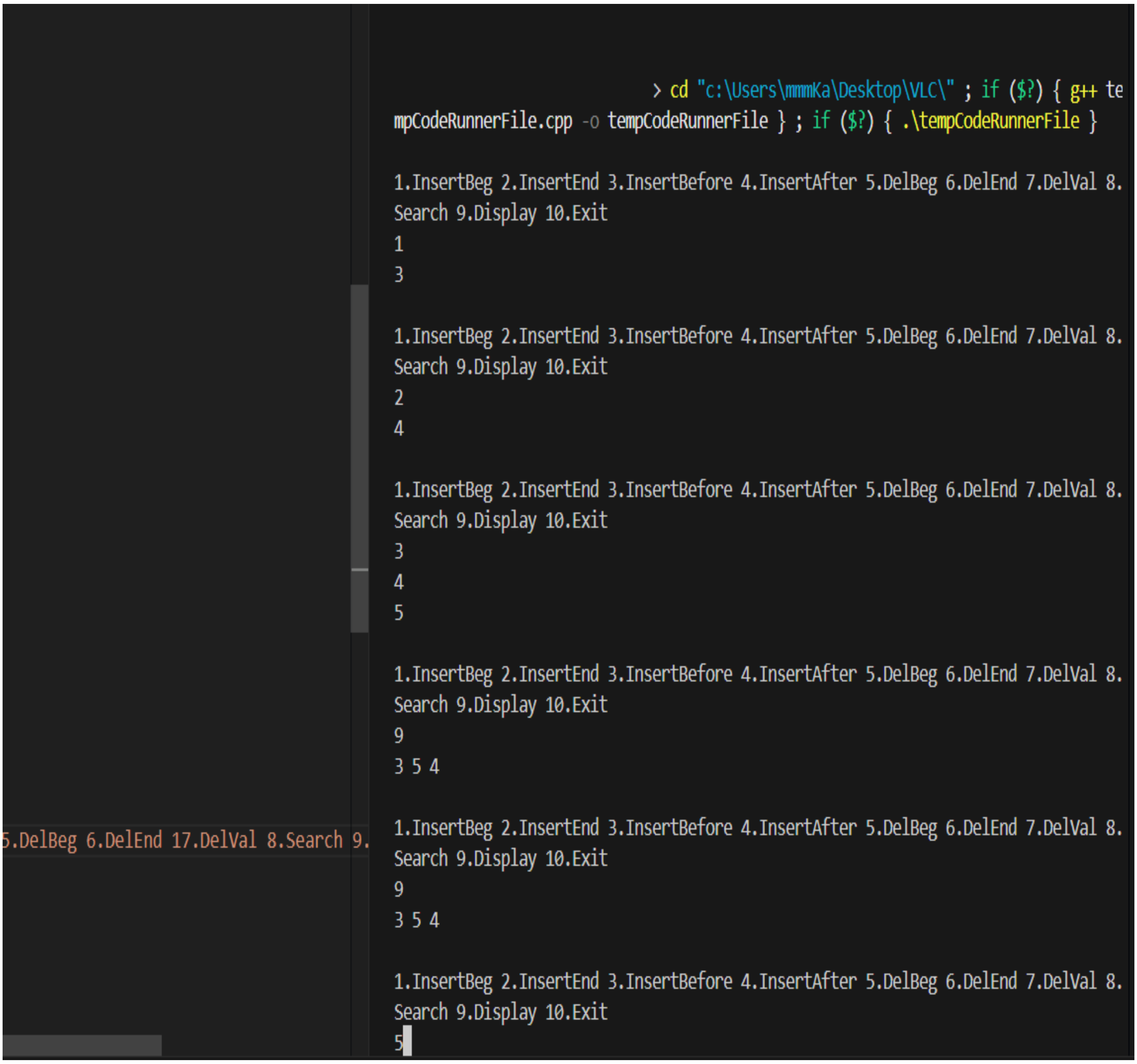
ASSIGMENT 5

Question 1

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

struct Node {
    int data;
    Node* next;
};

Node* head = nullptr;

void insertBeginning(int val) {
    Node* newNode = new Node{val, head};
    head = newNode;
}

void insertEnd(int val) {
    Node* newNode = new Node{val, nullptr};
    if (!head) head = newNode;
    else {
        Node* temp = head;
        while (temp->next) temp = temp->next;
        temp->next = newNode;
    }
}


void insertBefore(int key, int val) {
    if (!head) return;
    if (head->data == key) { insertBeginning(val); return; }
    Node* temp = head;
    while (temp->next && temp->next->data != key) temp = temp->next;
    if (temp->next) {
        Node* newNode = new Node{val, temp->next};
        temp->next = newNode;
    }
}


void insertAfter(int key, int val) {
    Node* temp = head;
    while (temp && temp->data != key) temp = temp->next;
    if (temp) {
        Node* newNode = new Node{val, temp->next};
        temp->next = newNode;
    }
}


void deleteBeginning() {
    if (!head) return;
    Node* temp = head;
    head = head->next;
    delete temp;
}
```

```
void deleteEnd() {
    if (!head) return;
    if (!head->next) { delete head; head = nullptr; return; }
    Node* temp = head;
    while (temp->next->next) temp = temp->next;
    delete temp->next;
    temp->next = nullptr;
}

void deleteValue(int key) {
    if (!head) return;
    if (head->data == key) { deleteBeginning(); return; }
    Node* temp = head;
    while (temp->next && temp->next->data != key) temp = temp->next;
    if (temp->next) {
        Node* del = temp->next;
        temp->next = del->next;
        delete del;
    }
}

void search(int key) {
    Node* temp = head;
    int pos = 1;
    while (temp) {
        if (temp->data == key) {
            cout << "Found at position " << pos << "\n";
            return;
        }
        temp = temp->next;
        pos++;
    }
    cout << "Not Found\n";
}

void display() {
    Node* temp = head;
    while (temp) { cout << temp->data << " "; temp = temp->next; }
    cout << "\n";
}

int main() {
    int choice, val, key;
    do {
        cout << "\n1.InsertBeg 2.InsertEnd 3.InsertBefore 4.InsertAfter 5.DelBeg 6.DelEnd 7.DelVal 8.Search 9.Display 10.Exit\n";
        cin >> choice;
        switch (choice) {
            case 1: cin >> val; insertBeginning(val); break;
            case 2: cin >> val; insertEnd(val); break;
            case 3: cin >> key >> val; insertBefore(key, val); break;
            case 4: cin >> key >> val; insertAfter(key, val); break;
            case 5: deleteBeginning(); break;
            case 6: deleteEnd(); break;
            case 7: cin >> key; deleteValue(key); break;
            case 8: cin >> key; search(key); break;

            case 9: display(); break;

        }

    } while (choice != 10);

    return 0;

}
```

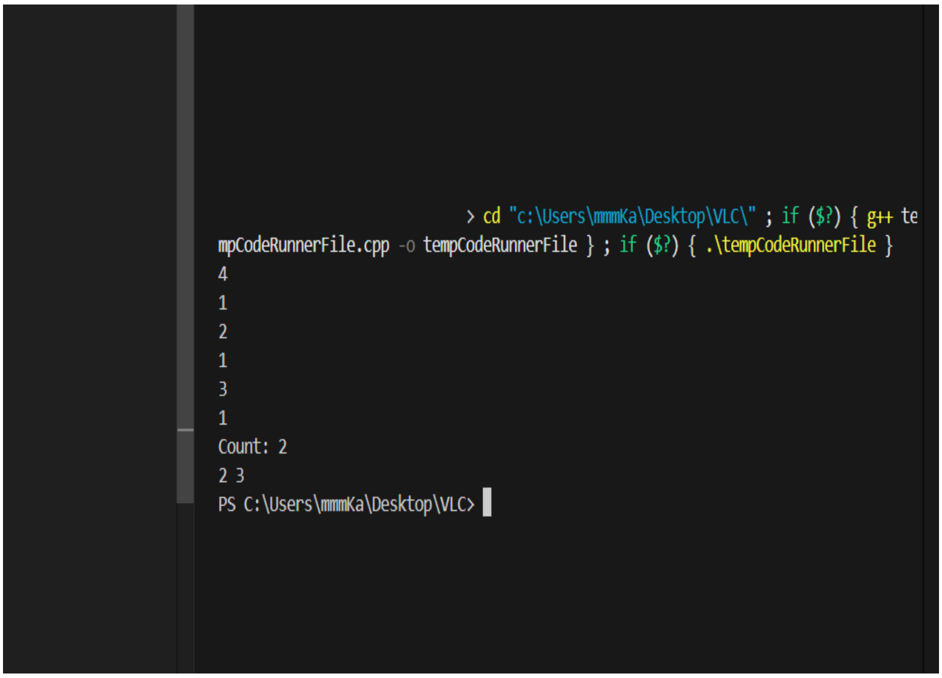


```
5.DelBeg 6.DelEnd 17.DelVal 8.Search 9.
```

```
                    > cd "c:\Users\mmmKa\Desktop\VLC\" ; if ($?) { g++ te
mpCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }

1.InsertBeg 2.InsertEnd 3.InsertBefore 4.InsertAfter 5.DelBeg 6.DelEnd 7.DelVal 8.
Search 9.Display 10.Exit
1
3

1.InsertBeg 2.InsertEnd 3.InsertBefore 4.InsertAfter 5.DelBeg 6.DelEnd 7.DelVal 8.
Search 9.Display 10.Exit
2
4

1.InsertBeg 2.InsertEnd 3.InsertBefore 4.InsertAfter 5.DelBeg 6.DelEnd 7.DelVal 8.
Search 9.Display 10.Exit
3
4
5

1.InsertBeg 2.InsertEnd 3.InsertBefore 4.InsertAfter 5.DelBeg 6.DelEnd 7.DelVal 8.
Search 9.Display 10.Exit
9
3 5 4

1.InsertBeg 2.InsertEnd 3.InsertBefore 4.InsertAfter 5.DelBeg 6.DelEnd 7.DelVal 8.
Search 9.Display 10.Exit
9
3 5 4

1.InsertBeg 2.InsertEnd 3.InsertBefore 4.InsertAfter 5.DelBeg 6.DelEnd 7.DelVal 8.
Search 9.Display 10.Exit
5
```

Question 2

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

struct Node {
    int data;
    Node* next;
};

Node* head = nullptr;

void insertEnd(int val) {
    Node* newNode = new Node{val, nullptr};
    if (!head) head = newNode;
    else {
        Node* temp = head;
        while (temp->next) temp = temp->next;
        temp->next = newNode;
    }
}

int deleteKey(int key) {
    int count = 0;
    while (head && head->data == key) {
        Node* temp = head;
        head = head->next;
        delete temp;
        count++;
    }
    Node* curr = head;
    while (curr && curr->next) {
        if (curr->next->data == key) {
            Node* del = curr->next;
            curr->next = del->next;
            delete del;
            count++;
        } else curr = curr->next;
    }
    return count;
}

void display() {
    Node* temp = head;
    while (temp) { cout << temp->data << " "; temp = temp->next; }
    cout << "\n";
}

int main() {
    int n, val, key;
    cin >> n;
    for (int i = 0; i < n; i++) { cin >> val; insertEnd(val); }
    cin >> key;
    int c = deleteKey(key);
    cout << "Count: " << c << "\n";
    display();
    return 0;
}
```

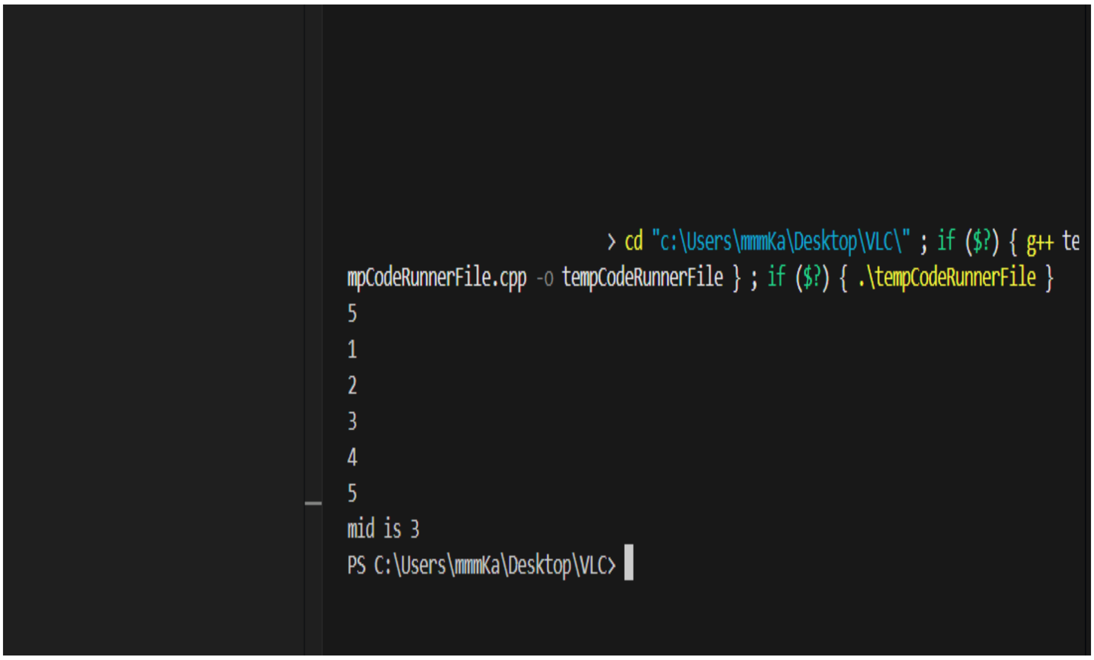


```
                    > cd "c:\Users\mmmKa\Desktop\VLC\" ; if ($?) { g++ te
mpCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
4
1
2
1
3
1
Count: 2
2 3
PS C:\Users\mmmKa\Desktop\VLC>
```

Question 3

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

struct Node {
    int data;
    Node* next;
};

Node* head = nullptr;

void insertEnd(int val) {
    Node* newNode = new Node{val, nullptr};
    if (!head) head = newNode;
    else {
        Node* temp = head;
        while (temp->next) temp = temp->next;
        temp->next = newNode;
    }
}

void findMiddle() {
    Node* slow = head;
    Node* fast = head;
    while (fast && fast->next) {
        slow = slow->next;
        fast = fast->next->next;
    }
    if (slow) cout << slow->data << "\n";
}
```

```
int main() {

   int n, val;

   cin >> n;

   for (int i = 0; i < n; i++) { cin >> val; insertEnd(val); }

   findMiddle();

   return 0;

}
```



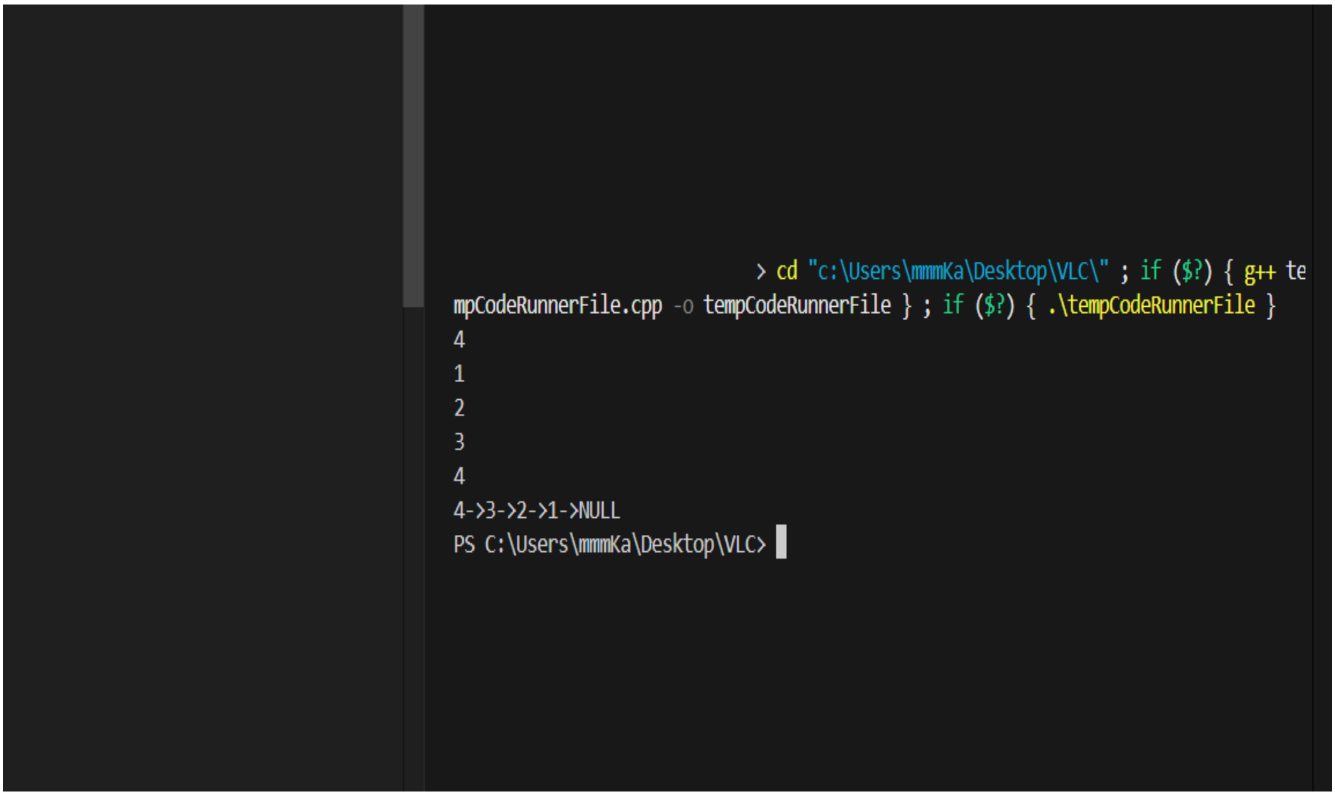
```
> cd "c:\Users\mmmKa\Desktop\VLC\" ; if ($?) { g++ te
mpCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
5
1
2
3
4
5
mid is 3
PS C:\Users\mmmKa\Desktop\VLC>
```

Question 4

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

struct Node {
  int data;
  Node* next;
};

Node* head = nullptr;

void insertEnd(int val) {
  Node* newNode = new Node{val, nullptr};
  if (!head) head = newNode;
  else {
    Node* temp = head;
    while (temp->next) temp = temp->next;
    temp->next = newNode;
  }
}

void reverseList() {
  Node* prev = nullptr;
  Node* curr = head;
  Node* next = nullptr;
  while (curr) {
    next = curr->next;
    curr->next = prev;
    prev = curr;
    curr = next;
  }
```

```
    head = prev;
}

void display() {
    Node* temp = head;
    while (temp) { cout << temp->data << "->"; temp = temp->next; }
    cout << "NULL\n";
}

int main() {
    int n, val;
    cin >> n;
    for (int i = 0; i < n; i++) { cin >> val; insertEnd(val); }
    reverseList();
    display();
    return 0;
}
```

-



```
                    > cd "c:\Users\mmmKa\Desktop\VLC\" ; if ($?) { g++ te
mpCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
4
1
2
3
4
4->3->2->1->NULL
PS C:\Users\mmmKa\Desktop\VLC>
```