C++

Assignment – 8

Name – Kishan R Vaghamashi

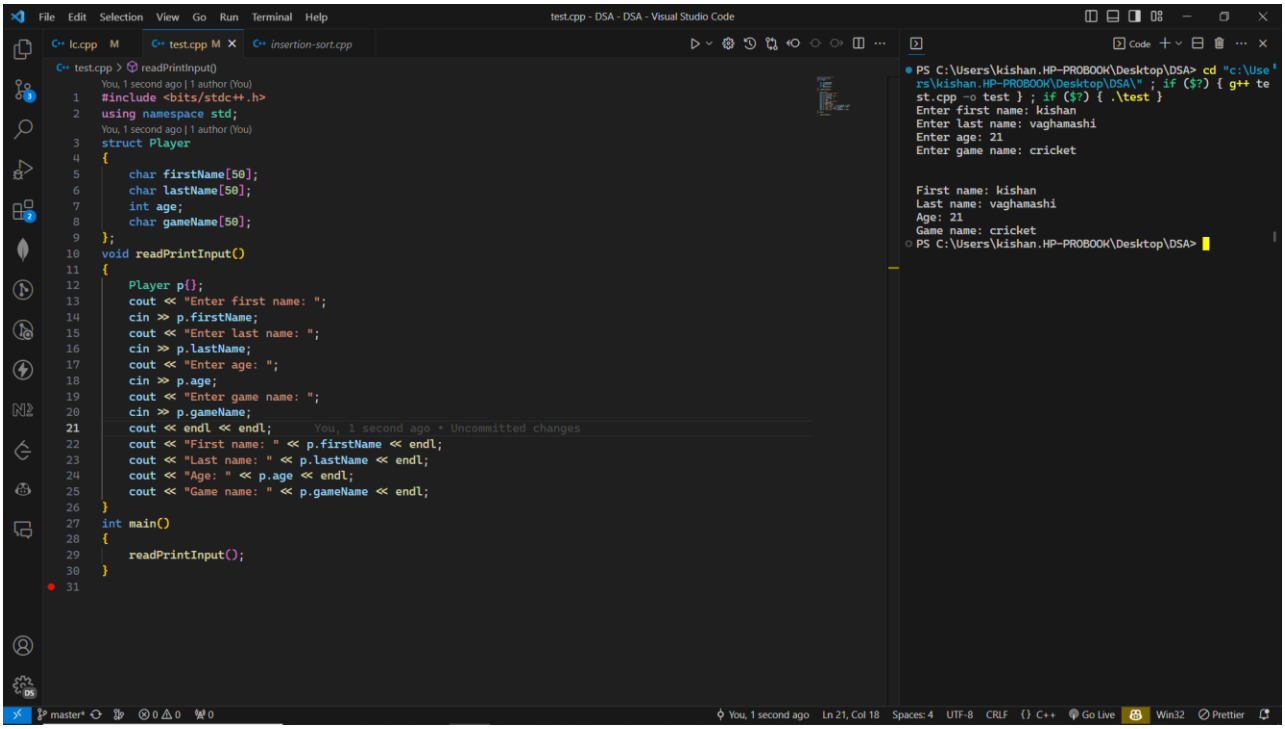
Student ID – 202312014

1)

A)

```cpp
#include <bits/stdc++.h>
using namespace std;
struct Player
{
    char firstName[50];
    char lastName[50];
    int age;
    char gameName[50];
};
void readPrintInput()
{
    Player p{};
    cout << "Enter first name: ";
    cin >> p.firstName;
    cout << "Enter last name: ";
    cin >> p.lastName;
    cout << "Enter age: ";
    cin >> p.age;
    cout << "Enter game name: ";
    cin >> p.gameName;
    cout << endl << endl;
    cout << "First name: " << p.firstName << endl;
    cout << "Last name: " << p.lastName << endl;
    cout << "Age: " << p.age << endl;
    cout << "Game name: " << p.gameName << endl;
}
int main()
{
    readPrintInput();
}
```

Output:

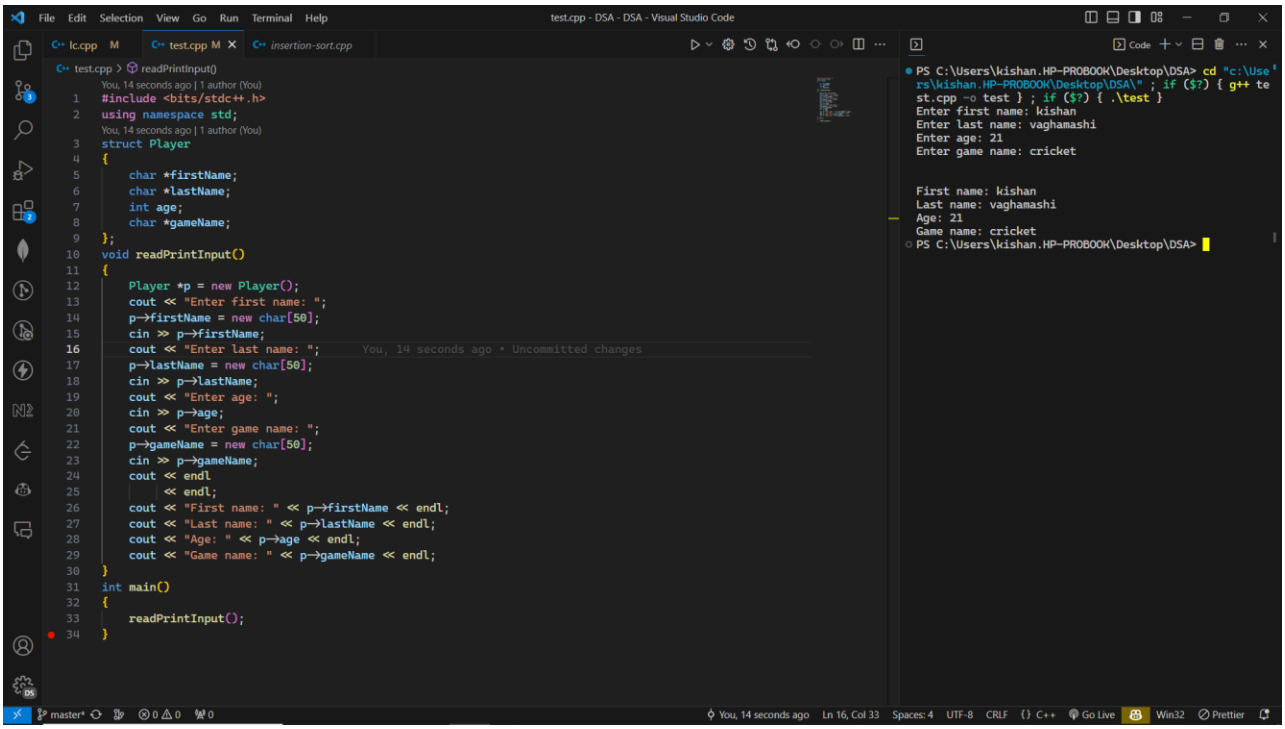


B)

```cpp
#include <bits/stdc++.h>
using namespace std;
struct Player
{
    char *firstName;
    char *lastName;
    int age;
    char *gameName;
};
void readPrintInput()
{
    Player *p = new Player();
    cout << "Enter first name: ";
    p->firstName = new char[50];
    cin >> p->firstName;
    cout << "Enter last name: ";
    p->lastName = new char[50];
    cin >> p->lastName;
    cout << "Enter age: ";
    cin >> p->age;
```

```
    cout << "Enter game name: ";
    p->gameName = new char[50];
    cin >> p->gameName;
    cout << endl
         << endl;
    cout << "First name: " << p->firstName << endl;
    cout << "Last name: " << p->lastName << endl;
    cout << "Age: " << p->age << endl;
    cout << "Game name: " << p->gameName << endl;
}
int main()
{
    readPrintInput();
}
```

Output:

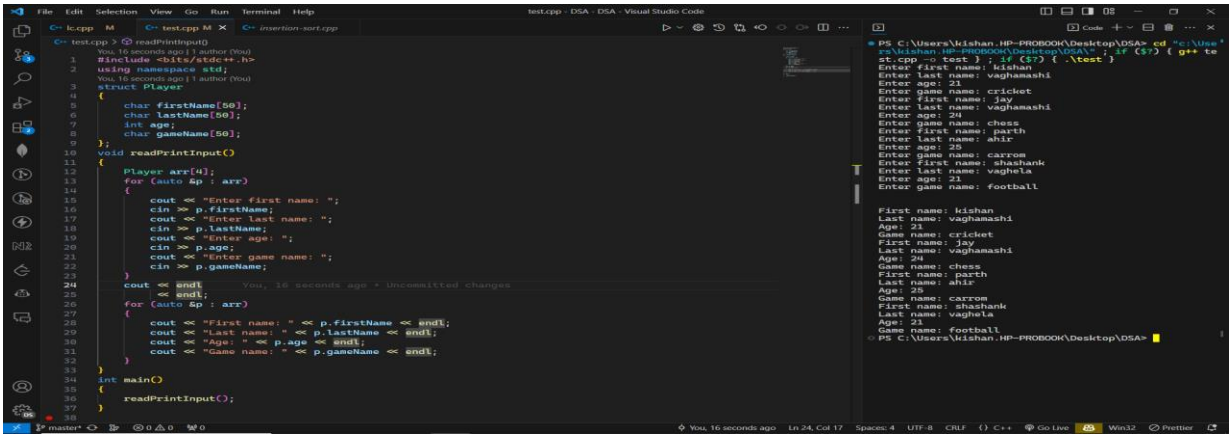


C)

```
#include <bits/stdc++.h>
using namespace std;
struct Player
{
    char firstName[50];
    char lastName[50];
```

```
    int age;
    char gameName[50];
};
void readPrintInput()
{
    Player arr[4];
    for (auto &p : arr)
    {
        cout << "Enter first name: ";
        cin >> p.firstName;
        cout << "Enter last name: ";
        cin >> p.lastName;
        cout << "Enter age: ";
        cin >> p.age;
        cout << "Enter game name: ";
        cin >> p.gameName;
    }
    cout << endl
         << endl;
    for (auto &p : arr)
    {
        cout << "First name: " << p.firstName << endl;
        cout << "Last name: " << p.lastName << endl;
        cout << "Age: " << p.age << endl;
        cout << "Game name: " << p.gameName << endl;
    }
}
int main()
{
    readPrintInput();
}
```

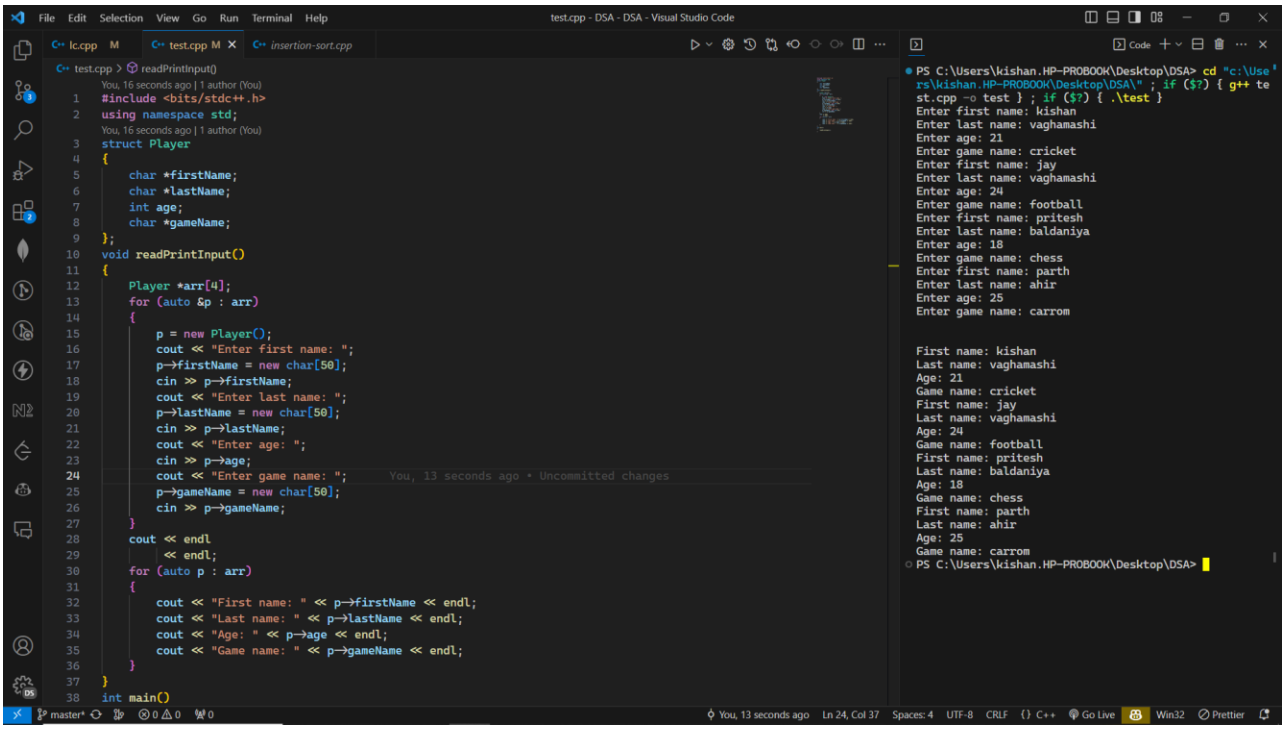
Output:



D)

```
#include <bits/stdc++.h>
using namespace std;
struct Player
{
    char *firstName;
    char *lastName;
    int age;
    char *gameName;
};
void readPrintInput()
{
    Player *arr[4];
    for (auto &p : arr)
    {
        p = new Player();
        cout << "Enter first name: ";
        p->firstName = new char[50];
        cin >> p->firstName;
        cout << "Enter last name: ";
        p->lastName = new char[50];
        cin >> p->lastName;
        cout << "Enter age: ";
        cin >> p->age;
        cout << "Enter game name: ";
        p->gameName = new char[50];
        cin >> p->gameName;
    }
    cout << endl
         << endl;
    for (auto p : arr)
    {
        cout << "First name: " << p->firstName << endl;
        cout << "Last name: " << p->lastName << endl;
        cout << "Age: " << p->age << endl;
        cout << "Game name: " << p->gameName << endl;
    }
}
int main()
{
    readPrintInput();
}
```

Output:

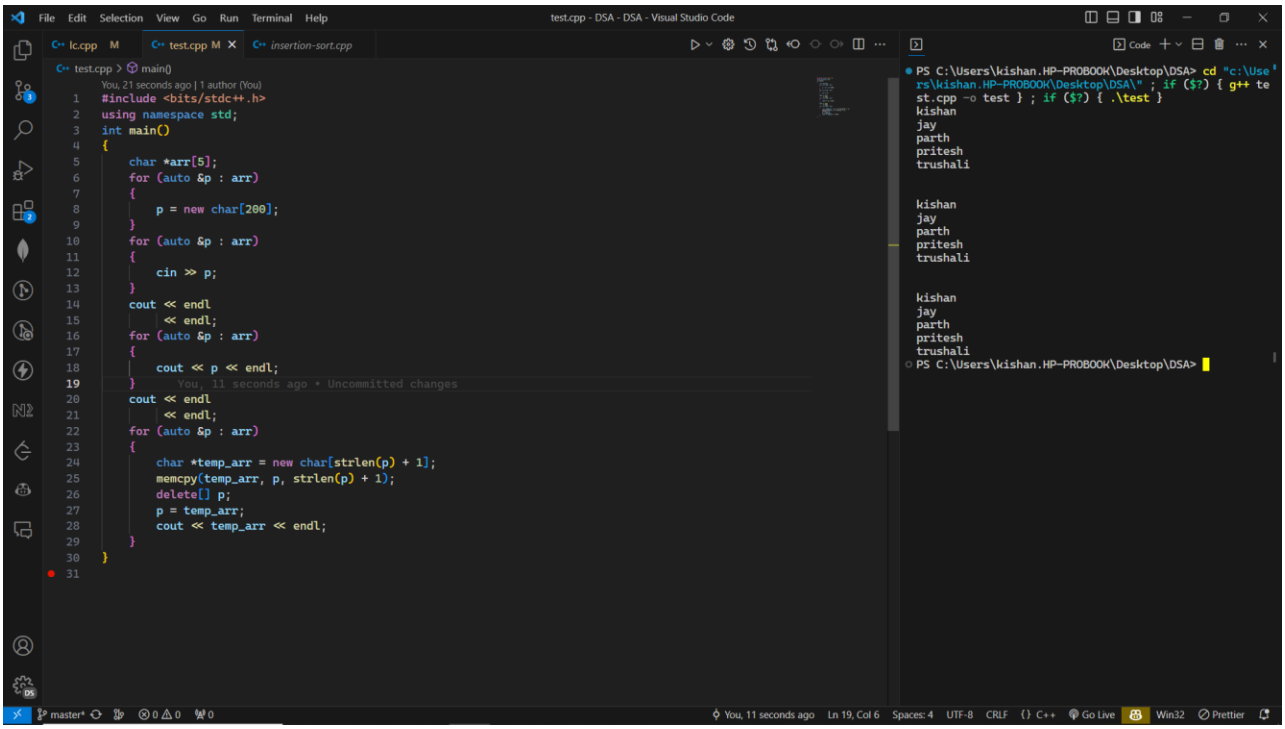


2)

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    char *arr[5];
    for (auto &p : arr)
    {
        p = new char[200];
    }
    for (auto &p : arr)
    {
        cin >> p;
    }
    cout << endl
         << endl;
    for (auto &p : arr)
    {
        cout << p << endl;
    }
    cout << endl
```

```
         << endl;
    for (auto &p : arr)
    {
        char *temp_arr = new char[strlen(p) + 1];
        memcpy(temp_arr, p, strlen(p) + 1);
        delete[] p;
        p = temp_arr;
        cout << temp_arr << endl;
    }
}
```

Output:

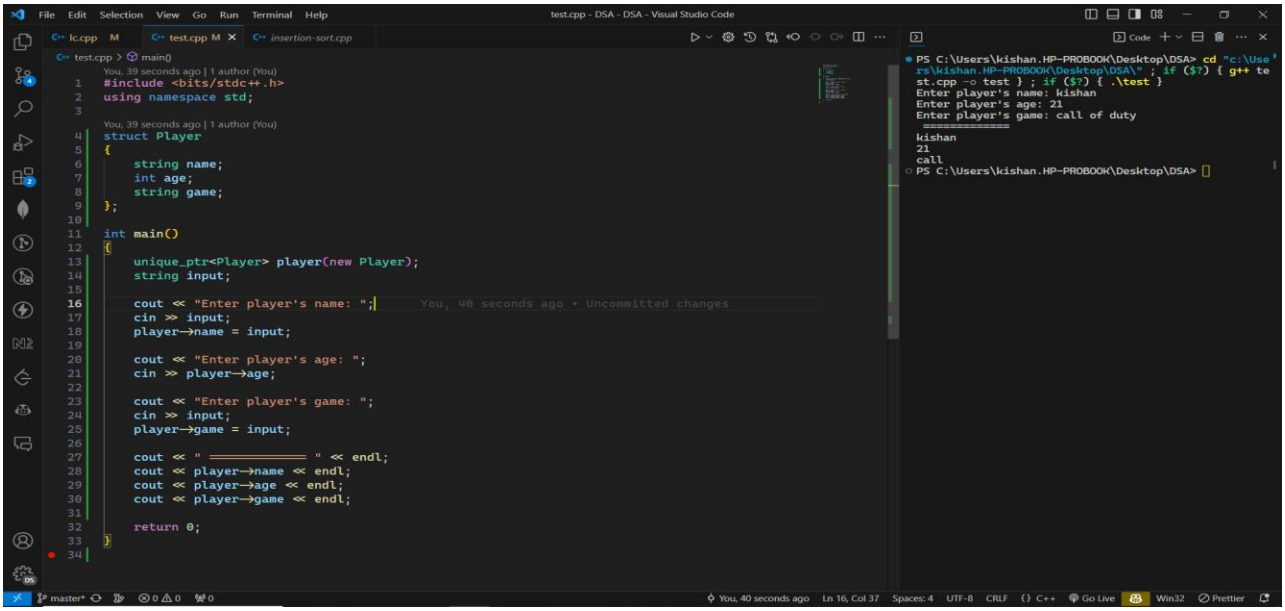


3)

```
#include <bits/stdc++.h>
using namespace std;

struct Player
{
    string name;
    int age;
```

```
    string game;
};

int main()
{
    unique_ptr<Player> player(new Player);
    string input;

    cout << "Enter player's name: ";
    cin >> input;
    player->name = input;

    cout << "Enter player's age: ";
    cin >> player->age;

    cout << "Enter player's game: ";
    cin >> input;
    player->game = input;

    cout << " ============== " << endl;
    cout << player->name << endl;
    cout << player->age << endl;
    cout << player->game << endl;

    return 0;
}
```

Output:

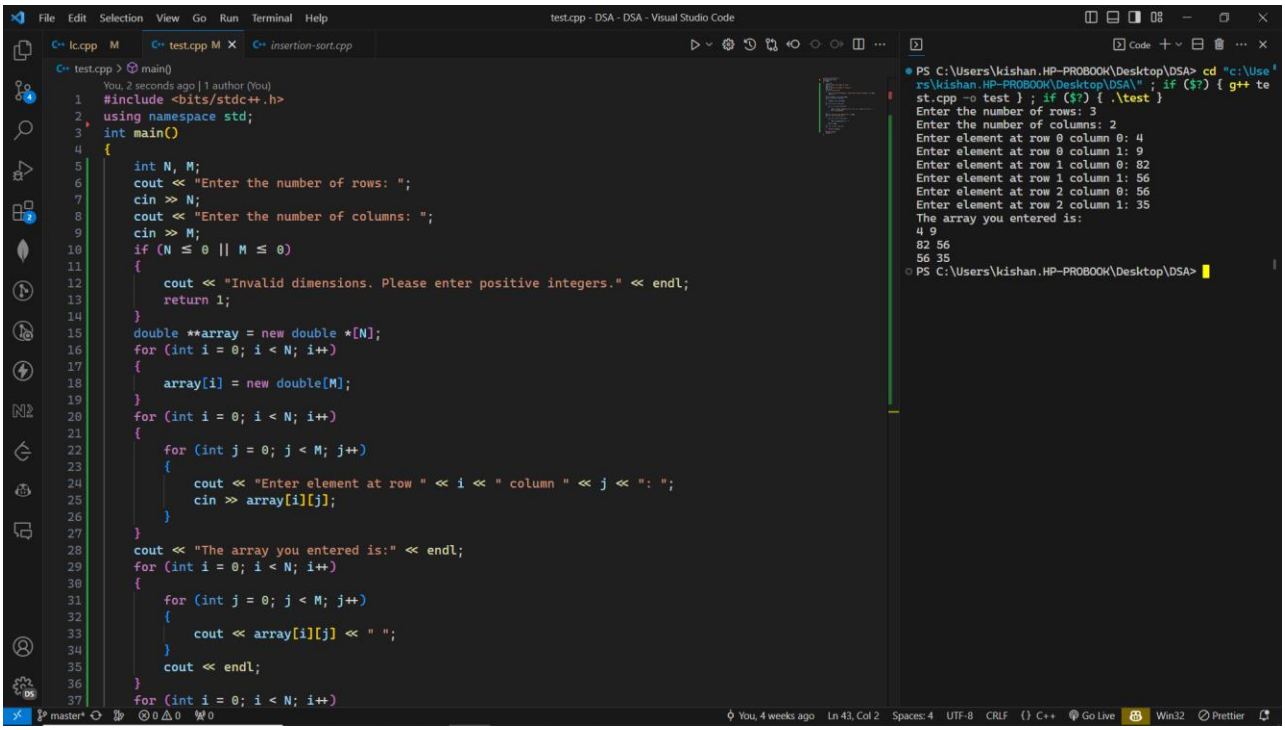


4)

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int N, M;
    cout << "Enter the number of rows: ";
    cin >> N;
    cout << "Enter the number of columns: ";
    cin >> M;
    if (N <= 0 || M <= 0)
    {
        cout << "Invalid dimensions. Please enter positive integers."
<< endl;
        return 1;
    }
    double **array = new double *[N];
    for (int i = 0; i < N; i++)
    {
        array[i] = new double[M];
    }
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < M; j++)
        {
            cout << "Enter element at row " << i << " column " << j <<
": ";
            cin >> array[i][j];
        }
    }
    cout << "The array you entered is:" << endl;
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < M; j++)
        {
            cout << array[i][j] << " ";
        }
        cout << endl;
    }
    for (int i = 0; i < N; i++)
    {
        delete[] array[i];
```

```
    }
    delete[] array;
    return 0;
}
```

Output:

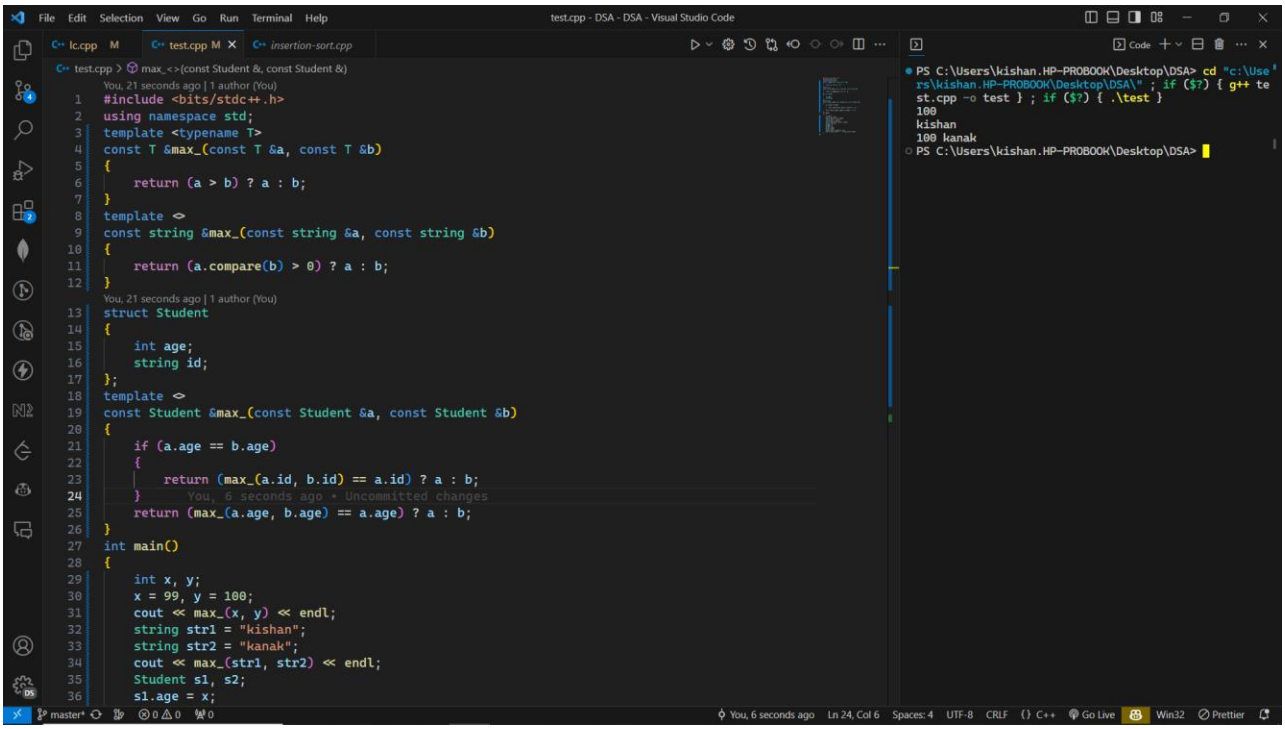


5)

```
#include <bits/stdc++.h>
using namespace std;
template <typename T>
const T &max_(const T &a, const T &b)
{
    return (a > b) ? a : b;
}
template <>
const string &max_(const string &a, const string &b)
{
    return (a.compare(b) > 0) ? a : b;
```

```
}
struct Student
{
    int age;
    string id;
};
template <>
const Student &max_(const Student &a, const Student &b)
{
    if (a.age == b.age)
    {
        return (max_(a.id, b.id) == a.id) ? a : b;
    }
    return (max_(a.age, b.age) == a.age) ? a : b;
}
int main()
{
    int x, y;
    x = 99, y = 100;
    cout << max_(x, y) << endl;
    string str1 = "kishan";
    string str2 = "kanak";
    cout << max_(str1, str2) << endl;
    Student s1, s2;
    s1.age = x;
    s2.age = y;
    s1.id = str1;
    s2.id = str2;
    Student max_s = max_(s1, s2);
    cout << max_s.age << " " << max_s.id << endl;
}
```

Output:

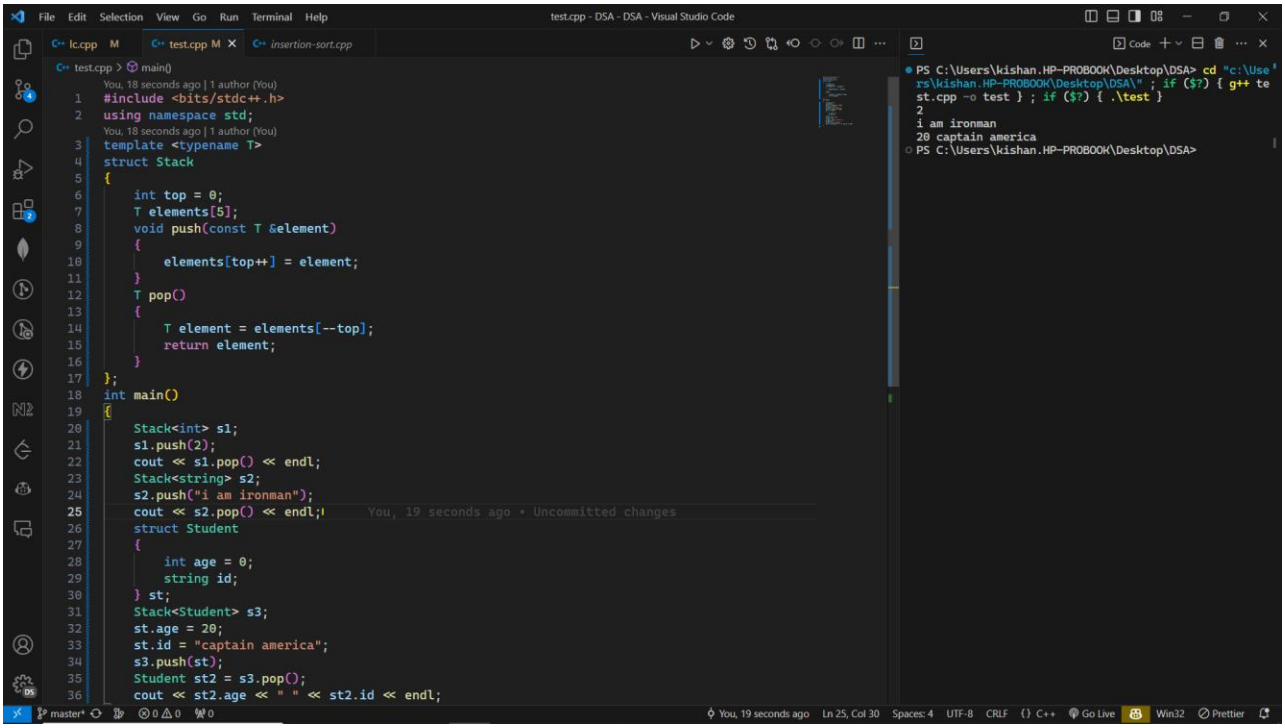


6)

```cpp
#include <bits/stdc++.h>
using namespace std;
template <typename T>
struct Stack
{
    int top = 0;
    T elements[5];
    void push(const T &element)
    {
        elements[top++] = element;
    }
    T pop()
    {
        T element = elements[--top];
        return element;
    }
};
```

```cpp
int main()
{
    Stack<int> s1;
    s1.push(2);
    cout << s1.pop() << endl;
    Stack<string> s2;
    s2.push("i am ironman");
    cout << s2.pop() << endl;
    struct Student
    {
        int age = 0;
        string id;
    } st;
    Stack<Student> s3;
    st.age = 20;
    st.id = "captain america";
    s3.push(st);
    Student st2 = s3.pop();
    cout << st2.age << " " << st2.id << endl;
}
```

Output:

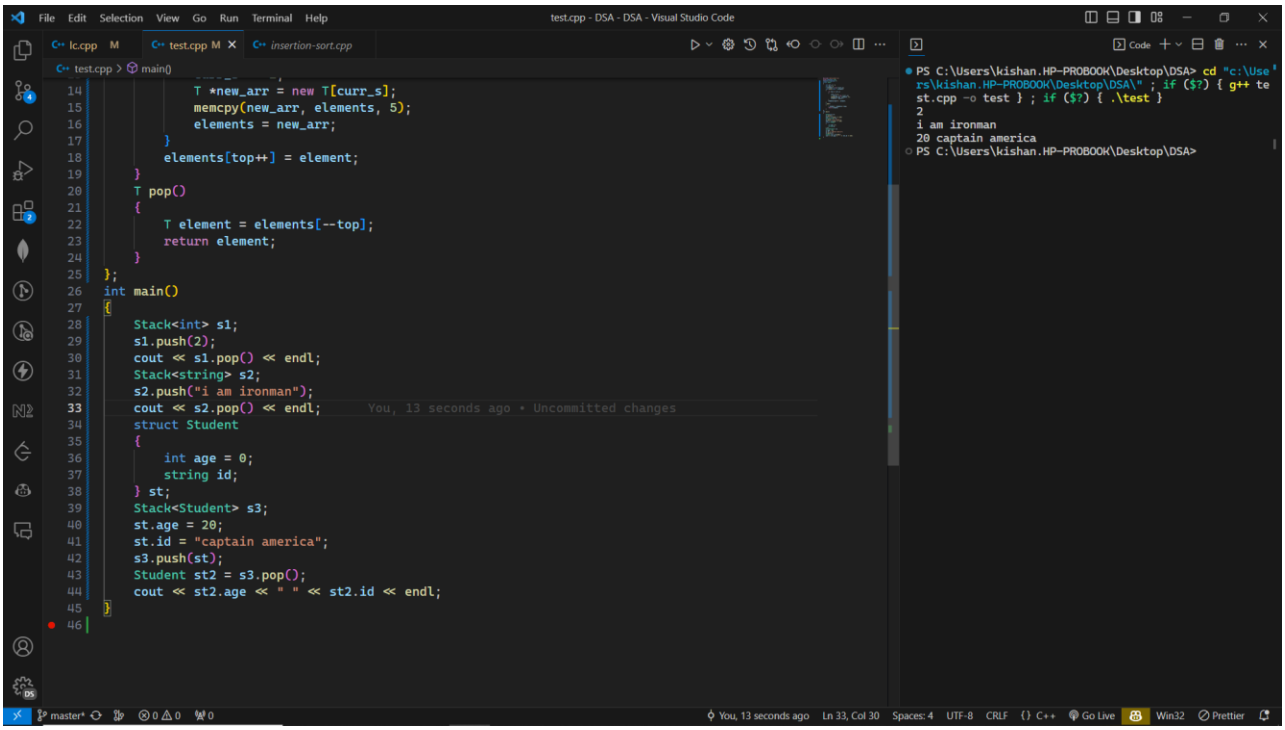


7)

```cpp
#include <bits/stdc++.h>
using namespace std;
template <typename T>
struct Stack
{
    int curr_s = 2;
    int top = 0;
    T *elements = new T[curr_s];
    void push(const T &element)
    {
        if (top == curr_s)
        {
            curr_s *= 2;
            T *new_arr = new T[curr_s];
            memcpy(new_arr, elements, 5);
            elements = new_arr;
        }
        elements[top++] = element;
    }
    T pop()
    {
        T element = elements[--top];
        return element;
    }
};
int main()
{
    Stack<int> s1;
    s1.push(2);
    cout << s1.pop() << endl;
    Stack<string> s2;
    s2.push("i am ironman");
    cout << s2.pop() << endl;
    struct Student
    {
        int age = 0;
        string id;
    } st;
    Stack<Student> s3;
    st.age = 20;
    st.id = "captain america";
```

```
    s3.push(st);
    Student st2 = s3.pop();
    cout << st2.age << " " << st2.id << endl;
}
```

Output:

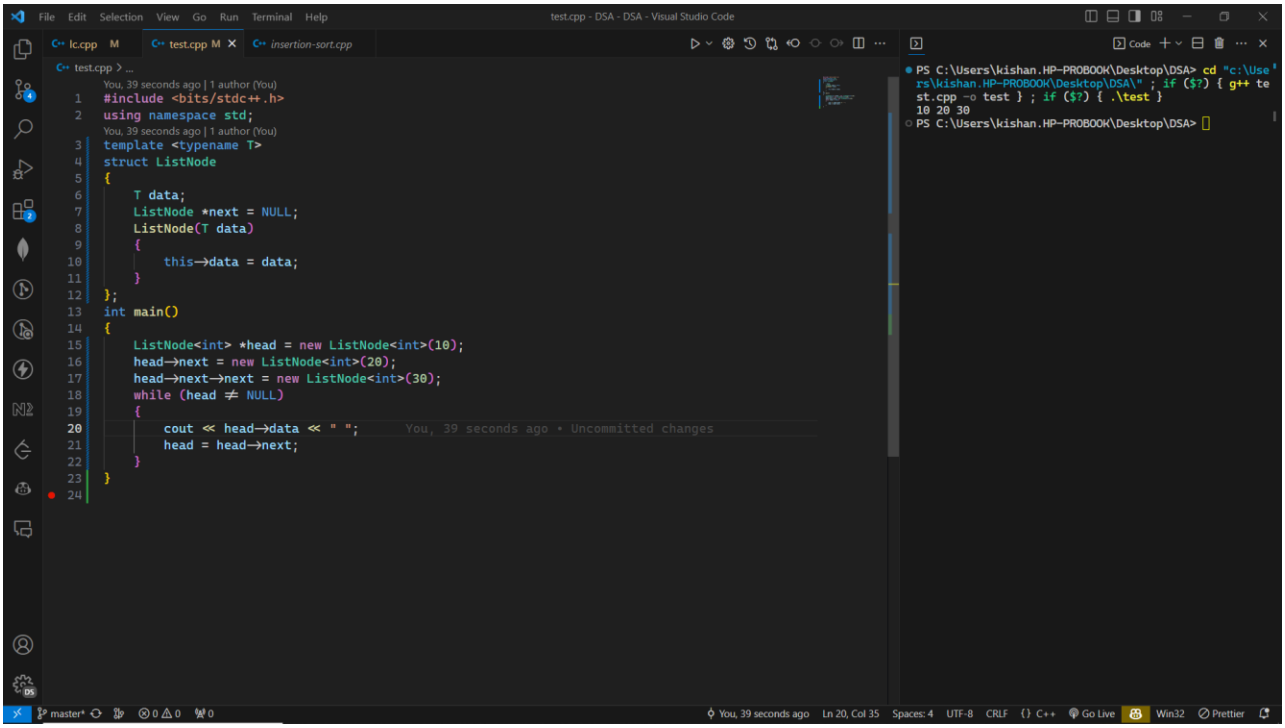


Que-4)

```
#include <bits/stdc++.h>
using namespace std;
template <typename T>
struct ListNode
{
    T data;
    ListNode *next = NULL;
    ListNode(T data)
    {
        this->data = data;
```

```
    }
};
int main()
{
    ListNode<int> *head = new ListNode<int>(10);
    head->next = new ListNode<int>(20);
    head->next->next = new ListNode<int>(30);
    while (head)
    {
        cout << head->data << " ";
        head = head->next;
    }
}
```

Output:




```
#include <bits/stdc++.h>
using namespace std;
template <typename T>
struct TreeNode
{
```

```
    T data;
    TreeNode *left;
    TreeNode *right;
    TreeNode(T data)
    {
        this->data = data;
    }
};
int main()
{
    TreeNode<int> *parent = new TreeNode<int>(10);
    parent->left = new TreeNode<int>(20);
    parent->right = new TreeNode<int>(30);
    cout << parent->data << " ";
    cout << parent->left->data << " ";
    cout << parent->right->data << " ";
}
```

Output: