

C++

Assignment – 8

Name – Kishan R Vaghamashi

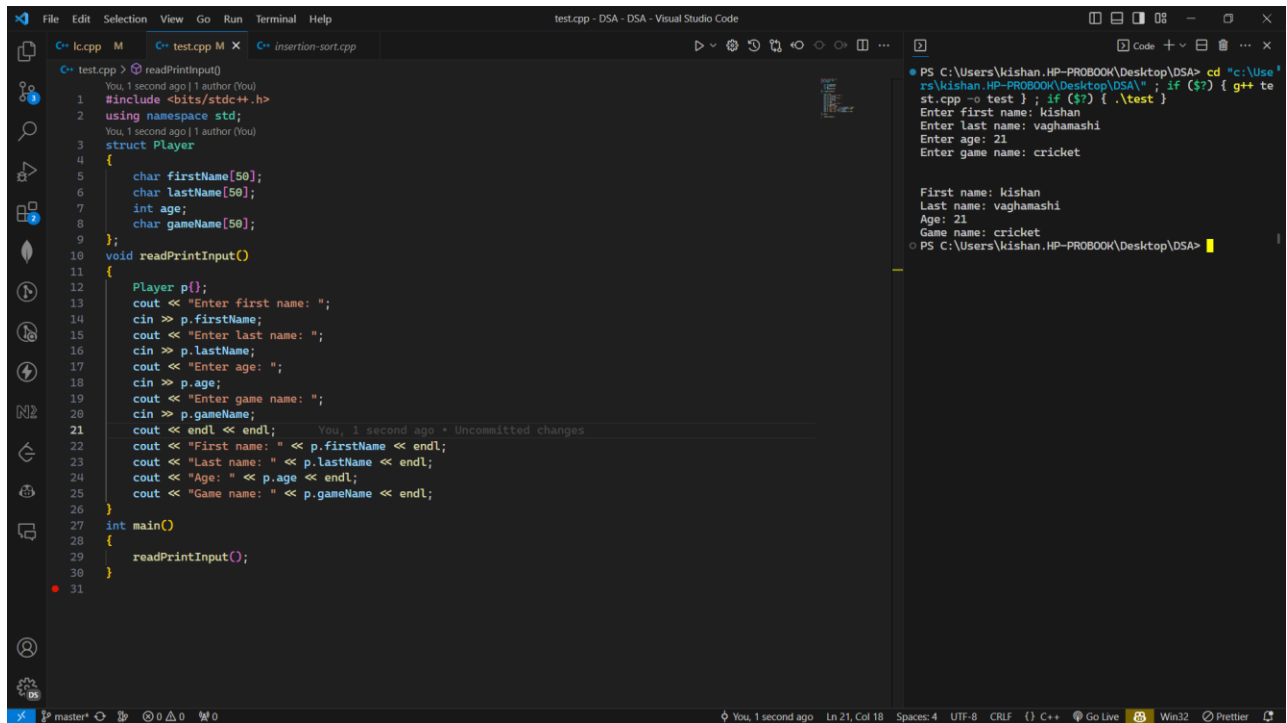
Student ID – 202312014

1)

A)

```
#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:



The screenshot shows the Visual Studio Code editor with a C++ file named `test.cpp`. The code defines a `Player` struct and a `readPrintInput` function. The terminal on the right shows the execution of the program, where the user enters 'kishan', 'vaghanashi', '21', and 'cricket'. The program then prints these values back.

```
#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();
}
```

Terminal Output:

```
PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA> cd "c:\Use
rs\kishan.HP-PROBOOK\Desktop\DSA" ; if ($?) { g++ te
st.cpp -o test } ; if ($?) { .\test }
Enter first name: kishan
Enter last name: vaghamashi
Enter age: 21
Enter game name: cricket

First name: kishan
Last name: vaghamashi
Age: 21
Game name: cricket
PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA>
```

B)

```
#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:

The screenshot shows the Visual Studio Code editor with a C++ file named `test.cpp`. The code defines a `Player` struct and a `readPrintInput` function. The `main` function calls `readPrintInput`. The output window on the right shows the program's execution, displaying prompts for first name, last name, age, and game name, followed by the entered values: kishan, vaghamashi, 21, and cricket.

```

C++ test.cpp M C++ test.cpp M X C++ insertion-sort.cpp
C++ test.cpp > readPrintInput()
You, 14 seconds ago | 1 author (You)
1 #include <bits/stdc++.h>
2 using namespace std;
3 struct Player
4 {
5     char *firstName;
6     char *lastName;
7     int age;
8     char *gameName;
9 };
10 void readPrintInput()
11 {
12     Player *p = new Player();
13     cout << "Enter first name: ";
14     p->firstName = new char[50];
15     cin >> p->firstName;
16     cout << "Enter last name: ";
17     p->lastName = new char[50];
18     cin >> p->lastName;
19     cout << "Enter age: ";
20     cin >> p->age;
21     cout << "Enter game name: ";
22     p->gameName = new char[50];
23     cin >> p->gameName;
24     cout << endl
25         << endl;
26     cout << "First name: " << p->firstName << endl;
27     cout << "Last name: " << p->lastName << endl;
28     cout << "Age: " << p->age << endl;
29     cout << "Game name: " << p->gameName << endl;
30 }
31 int main()
32 {
33     readPrintInput();
34 }

```

PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA> cd "c:\Use
rs\kishan.HP-PROBOOK\Desktop\DSA" ; if (\$?) { g++ te
st.cpp -o test } ; if (\$?) { .\test }
Enter first name: kishan
Enter last name: vaghamashi
Enter age: 21
Enter game name: cricket

First name: kishan
Last name: vaghamashi
Age: 21
Game name: cricket
PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA>

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:

The screenshot shows the Visual Studio Code interface with the C++ code from the previous block in the editor. The terminal window on the right displays the output of the program, which prompts the user to enter details for four players and then prints them out.

```

PS C:\Users\Akshay\HP-PROBOOK\Desktop\DSA> cd "C:\Users\Akshay\HP-PROBOOK\Desktop\DSA" & g++ test.cpp -o test & .\test
Enter first name: kishan
Enter last name: vaghamashi
Enter age: 21
Enter game name: cricket
Enter first name: jay
Enter last name: vaghamashi
Enter age: 24
Enter game name: chess
Enter first name: parth
Enter last name: ahir
Enter age: 23
Enter game name: carrom
Enter first name: sheshank
Enter last name: vaghela
Enter age: 21
Enter game name: football

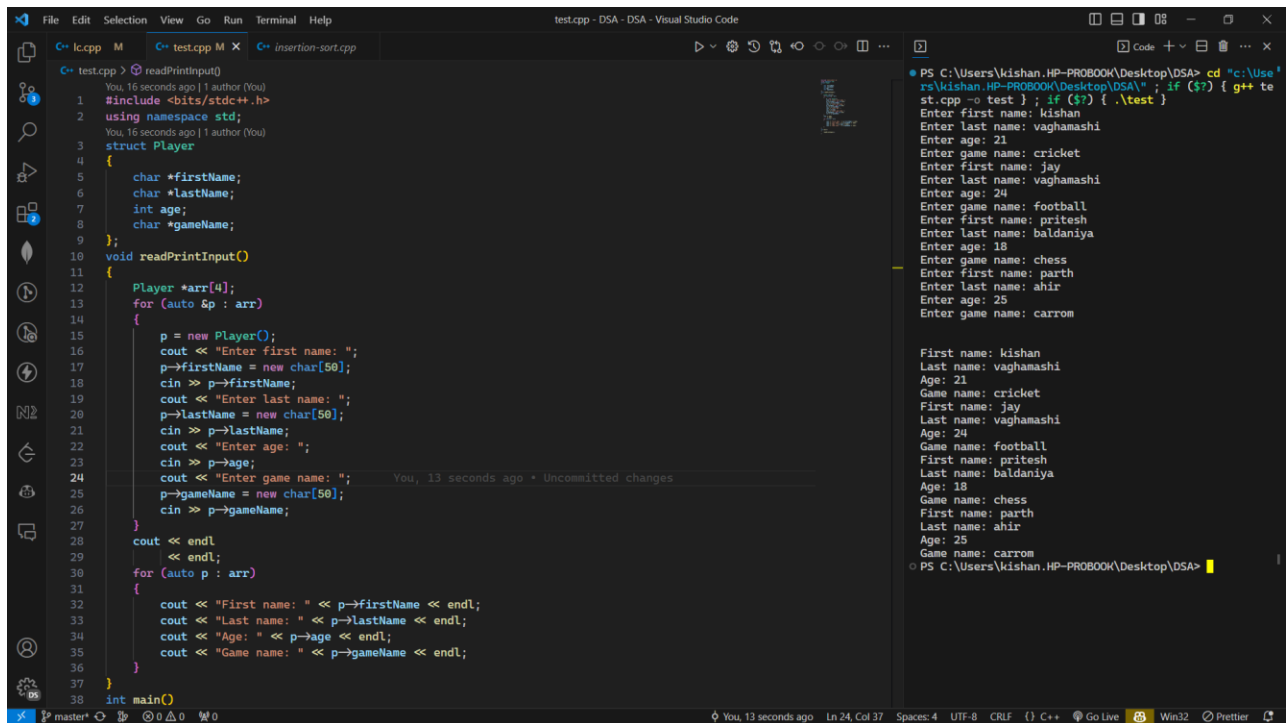
First name: kishan
Last name: vaghamashi
Age: 21
Game name: cricket
First name: jay
Last name: vaghamashi
Age: 24
Game name: chess
First name: parth
Last name: ahir
Age: 23
Game name: carrom
First name: sheshank
Last name: vaghela
Age: 21
Game name: football

```

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:



The screenshot shows the Visual Studio Code editor with a C++ file named `test.cpp`. The code defines a `Player` struct with fields `firstName`, `lastName`, `age`, and `gameName`. It includes a `readPrintInput()` function that creates an array of 4 `Player` objects, prompts the user for input, and prints the details. The output window on the right shows the program's execution, displaying the input for each player and their details.

```
#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;
    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:

```
Enter first name: kishan
Enter last name: vaghamashi
Enter age: 21
Enter game name: cricket
Enter first name: jay
Enter last name: vaghamashi
Enter age: 24
Enter game name: football
Enter first name: pritesh
Enter last name: baldaniya
Enter age: 18
Enter game name: chess
Enter first name: parth
Enter last name: ahir
Enter age: 25
Enter game name: carrom

First name: kishan
Last name: vaghamashi
Age: 21
Game name: cricket
First name: jay
Last name: vaghamashi
Age: 24
Game name: football
First name: pritesh
Last name: baldaniya
Age: 18
Game name: chess
First name: parth
Last name: ahir
Age: 25
Game name: carrom
```

2)

```
#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;
    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:

The screenshot shows the Visual Studio Code editor with a C++ file named `test.cpp`. The code in the editor is as follows:

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int main()
4  {
5      char *arr[5];
6      for (auto &p : arr)
7      {
8          p = new char[200];
9      }
10     for (auto &p : arr)
11     {
12         cin >> p;
13     }
14     cout << endl;
15     << endl;
16     for (auto &p : arr)
17     {
18         cout << p << endl;
19     }
20     cout << endl;
21     << endl;
22     for (auto &p : arr)
23     {
24         char *temp_arr = new char[strlen(p) + 1];
25         memcpy(temp_arr, p, strlen(p) + 1);
26         delete[] p;
27         p = temp_arr;
28         cout << temp_arr << endl;
29     }
30 }
31

```

The output window on the right shows the following output:

```

kishan
jay
parth
pritesh
trushali

kishan
jay
parth
pritesh
trushali

```

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:

The screenshot shows the Visual Studio Code editor with a C++ file named `test.cpp`. The code defines a `Player` struct with `name`, `age`, and `game` members, and a `main` function that uses `unique_ptr` to create a `Player` object and prompts the user for input. The terminal on the right shows the output of the program, which matches the code's logic: it prompts for name, age, and game, and then displays the entered values separated by a line of equals signs.

```

PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA> cd "C:\Users\kishan.HP-PROBOOK\Desktop\DSA" & if ($?) { g++ test.cpp -o test }; if ($?) { .\test }
Enter player's name: kishan
Enter player's age: 21
Enter player's game: call of duty
=====
kishan
21
call of duty
PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA>

```

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:

The screenshot shows a Visual Studio Code editor with a C++ file named `test.cpp`. The code defines a 2D array of doubles and prompts the user to enter the number of rows and columns. It then prompts the user to enter elements for each row and column. The output window shows the program's execution, including the user's input and the resulting array.

```

C++ test.cpp M test.cpp M X insertion-sort.cpp
C++ test.cpp > main()
You, 2 seconds ago | 1 author (You)
1 #include <bits/stdc++.h>
2 using namespace std;
3 int main()
4 {
5     int N, M;
6     cout << "Enter the number of rows: ";
7     cin >> N;
8     cout << "Enter the number of columns: ";
9     cin >> M;
10    if (N <= 0 || M <= 0)
11    {
12        cout << "Invalid dimensions. Please enter positive integers." << endl;
13        return 1;
14    }
15    double **array = new double *[N];
16    for (int i = 0; i < N; i++)
17    {
18        array[i] = new double[M];
19    }
20    for (int i = 0; i < N; i++)
21    {
22        for (int j = 0; j < M; j++)
23        {
24            cout << "Enter element at row " << i << " column " << j << ": ";
25            cin >> array[i][j];
26        }
27    }
28    cout << "The array you entered is:" << endl;
29    for (int i = 0; i < N; i++)
30    {
31        for (int j = 0; j < M; j++)
32        {
33            cout << array[i][j] << " ";
34        }
35        cout << endl;
36    }
37    for (int i = 0; i < N; i++)

```

```

PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA> cd "c:\Use
rs\kishan.HP-PROBOOK\Desktop\DSA" ; if ($?) { g++ te
st.cpp -o test } ; if ($?) { .\test }
Enter the number of rows: 3
Enter the number of columns: 2
Enter element at row 0 column 0: 4
Enter element at row 0 column 1: 9
Enter element at row 1 column 0: 82
Enter element at row 1 column 1: 56
Enter element at row 2 column 0: 56
Enter element at row 2 column 1: 35
The array you entered is:
4 9
82 56
56 35
PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA>

```

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;
}

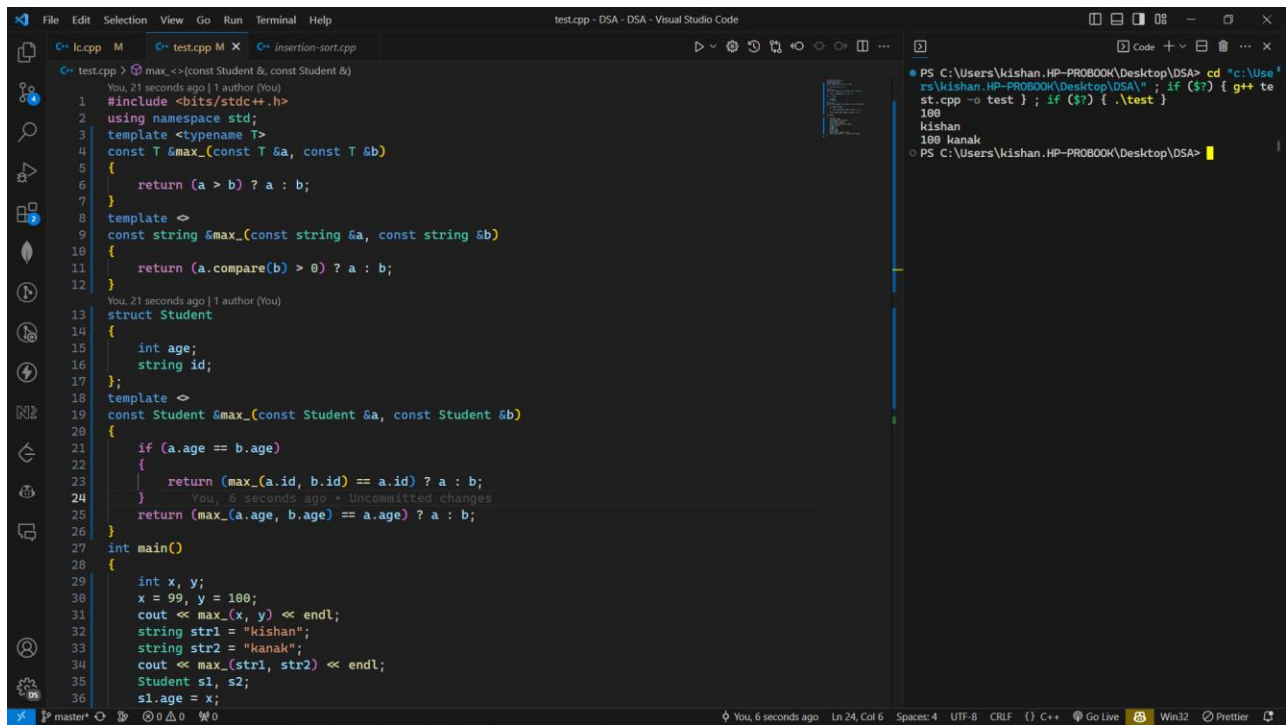
```

```

}
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:



```
test.cpp - DSA - Visual Studio Code
C++ test.cpp M X C++ insertion-sort.cpp
C++ test.cpp > max_<T>(const Student &, const Student &)
You, 21 seconds ago | 1 author (You)
1 #include <bits/stdc++.h>
2 using namespace std;
3 template <typename T>
4 const T &max_(const T &a, const T &b)
5 {
6     return (a > b) ? a : b;
7 }
8 template <
9 const string &max_(const string &a, const string &b)
10 {
11     return (a.compare(b) > 0) ? a : b;
12 }
13 You, 21 seconds ago | 1 author (You)
14 struct Student
15 {
16     int age;
17     string id;
18 };
19 template <
20 const Student &max_(const Student &a, const Student &b)
21 {
22     if (a.age == b.age)
23     {
24         return (max_(a.id, b.id) == a.id) ? a : b;
25     }
26     return (max_(a.age, b.age) == a.age) ? a : b;
27 }
28 int main()
29 {
30     int x, y;
31     x = 99, y = 100;
32     cout << max_(x, y) << endl;
33     string str1 = "kishan";
34     string str2 = "kanak";
35     cout << max_(str1, str2) << endl;
36     Student s1, s2;
37     s1.age = x;
38     s2.age = y;
39     cout << max_(s1, s2) << endl;
40 }
```

PS C:\Users\Kishan.HP-PROBOOK\Desktop\DSA> cd "C:\Users\Kishan.HP-PROBOOK\Desktop\DSA" & if (\$?) { g++ test.cpp -o test } ; if (\$?) { .\test }

kishan
100 kanak
PS C:\Users\Kishan.HP-PROBOOK\Desktop\DSA>

6)

```
#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;
    }
};
```

```

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:

The screenshot shows the Visual Studio Code editor with a C++ file named test.cpp. The code defines a Stack template, a Student struct, and a main function that demonstrates their usage. The output window on the right shows the execution results.

```

PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA> cd "C:\Users\kishan.HP-PROBOOK\Desktop\DSA" & if ($?) { g++ test.cpp -o test } & if ($?) { .\test }
2
i am ironman
20 captain america
PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA>

```

7)

```
#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:

The screenshot shows a Visual Studio Code editor with a C++ file named `test.cpp`. The code defines a `Stack` template and a `Student` struct. In the `main` function, it demonstrates stack operations: pushing 2 onto `s1`, popping it, pushing "i am ironman" onto `s2`, popping it, pushing a `Student` object onto `s3`, and popping it to print the age and ID. The output in the terminal is:

```

2
i am ironman
20 captain america

```

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:

The screenshot shows the Visual Studio Code editor with a C++ file named `test.cpp`. The code defines a `ListNode` struct and a `main` function that creates a linked list with nodes containing values 10, 20, and 30. The terminal output shows the execution of the program, resulting in the values 10, 20, and 30 being printed on a single line.

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  template <typename T>
4  struct ListNode
5  {
6      T data;
7      ListNode *next = NULL;
8      ListNode(T data)
9      {
10         this->data = data;
11     }
12 };
13 int main()
14 {
15     ListNode<int> *head = new ListNode<int>(10);
16     head->next = new ListNode<int>(20);
17     head->next->next = new ListNode<int>(30);
18     while (head != NULL)
19     {
20         cout << head->data << " ";
21         head = head->next;
22     }
23 }

```

```

PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA> cd "C:\Users\kishan.HP-PROBOOK\Desktop\DSA" & if ($?) { g++ test.cpp -o test } ; if ($?) { .\test }
10 20 30
PS C:\Users\kishan.HP-PROBOOK\Desktop\DSA>

```

```

#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:

The screenshot shows the Visual Studio Code interface with a C++ file named 'test.cpp'. The code is as follows:

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  template <typename T>
4  struct TreeNode
5  {
6      T data;
7      TreeNode *left;
8      TreeNode *right;
9      TreeNode(T data)
10     {
11         this->data = data;
12     }
13 };
14 int main()
15 {
16     TreeNode<int> *parent = new TreeNode<int>(10);
17     parent->left = new TreeNode<int>(20);
18     parent->right = new TreeNode<int>(30);
19     cout << parent->data << " ";
20     cout << parent->left->data << " ";
21     cout << parent->right->data << " ";
22 }

```

The status bar at the bottom indicates the file is on the 'master' branch, at line 22, column 2, with 4 spaces, using UTF-8 encoding and CRLF line endings. The editor is running C++ with the Go Live extension and the Prettier formatter.