

Assignment : 8

Programming

Name:-Jeevan Rajpurohit

Student id :- 202312090

1). Create a structure representing a player. It should have player's first name, last name, age and name of the game. Do the following activities on this structure-

a). Write a function that creates object of this structure, reads inputs from user for the properties of the object and prints the read values.

```
#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();
}

```

The screenshot shows a web browser window with the URL `onlinegdb.com/online_c++_compiler`. The code editor displays the following C++ code:

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 struct Player
4 {
5     char firstName[50];
6     char lastName[50];
7     int age;
8     char gameName[50];
9 };
10 void readPrintInput()
11 {
12     Player p{};
13     cout << "Enter first name: ";
14     cin >> p.firstName;
15     cout << "Enter last name: ";
16     cin >> p.lastName;
17     cout << "Enter age: ";
18     cin >> p.age;
19     cout << "Enter game name: ";
20     cin >> p.gameName;
21 }

```

The output window shows the following text:

```

Enter game name: ludo

First name: jeevan
Last name: rajpurohit
Age: 22
Game name: ludo

...Program finished with exit code 0
Press ENTER to exit console.

```

b). Do a. above using pointer to the structure object.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
struct Player
```

```
{
```

```
    char *firstName;
```

```
    char *lastName;
```

```
    int age;
```

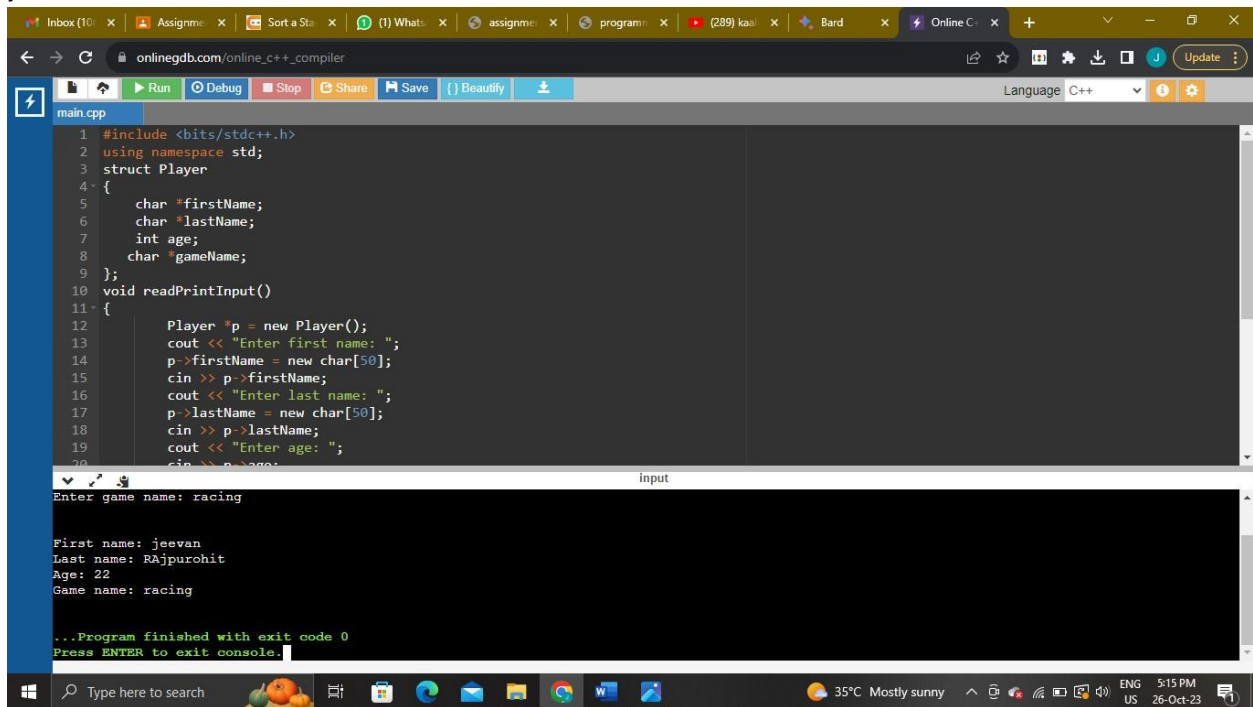
```
    char *gameName;
};

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();
```

}



The screenshot shows a web browser with multiple tabs. The active tab is 'onlinegdb.com/online_c++_compiler'. The interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The main area displays a C++ program in 'main.cpp' with the following code:

```
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;
```

The output window shows the program's execution:

```
Enter game name: racing

First name: jeevan
Last name: RAjpurohit
Age: 22
Game name: racing

...Program finished with exit code 0
Press ENTER to exit console.
```

c). Create an array of size 4 and read from user the values and print them
`#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)`

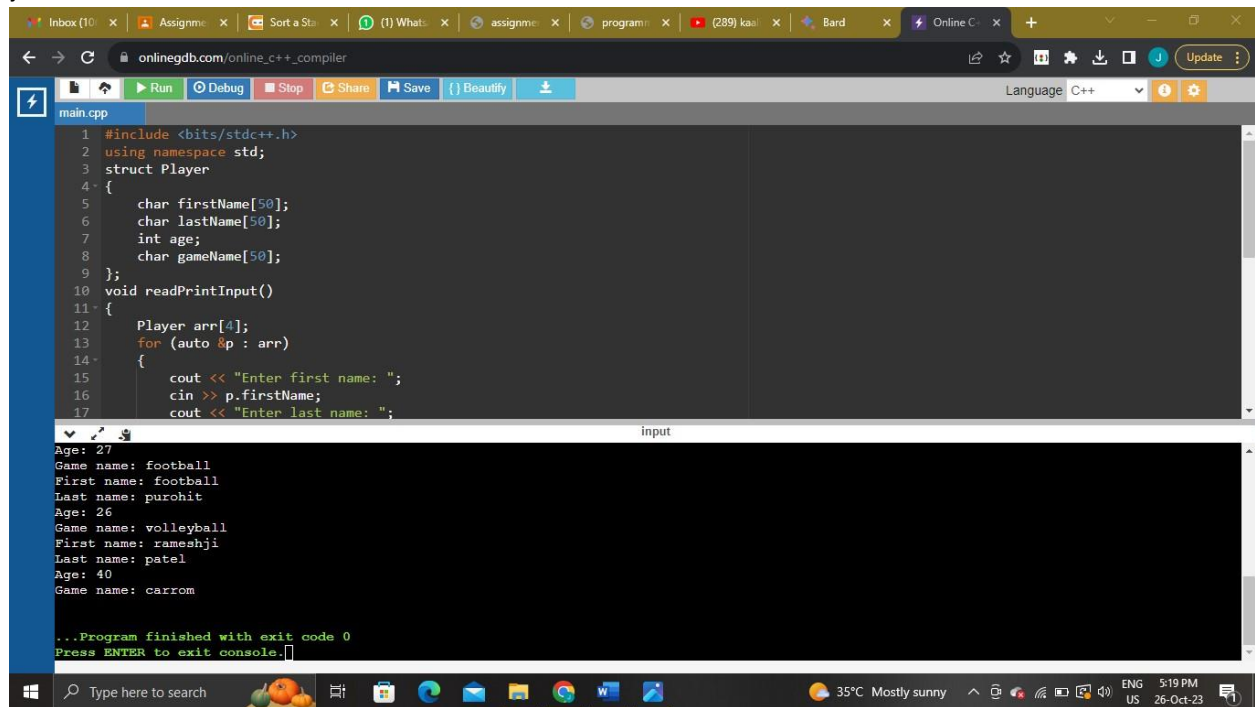
`{`

`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();
```

}



The screenshot shows a web browser with the onlinegdb.com online C++ compiler. The code editor displays a C++ program in a file named main.cpp. The program defines a struct Player with fields firstName, lastName, age, and gameName, each with a size of 50. It then defines a function readPrintInput that takes an array of Player objects and prints their details. The output window shows the program's execution, where it prompts for input and displays the details of three players: Purohit (Age 27, Game: football), Rameshji (Age 26, Game: volleyball), and Patel (Age 40, Game: carrom). The program finishes with exit code 0.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 struct Player
4 {
5     char firstName[50];
6     char lastName[50];
7     int age;
8     char gameName[50];
9 };
10 void readPrintInput()
11 {
12     Player arr[4];
13     for (auto &p : arr)
14     {
15         cout << "Enter first name: ";
16         cin >> p.firstName;
17         cout << "Enter last name: ";
```

input

```
Age: 27
Game name: football
First name: football
Last name: purohit
Age: 26
Game name: volleyball
First name: rameshji
Last name: patel
Age: 40
Game name: carrom

...Program finished with exit code 0
Press ENTER to exit console.
```

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();
```

```

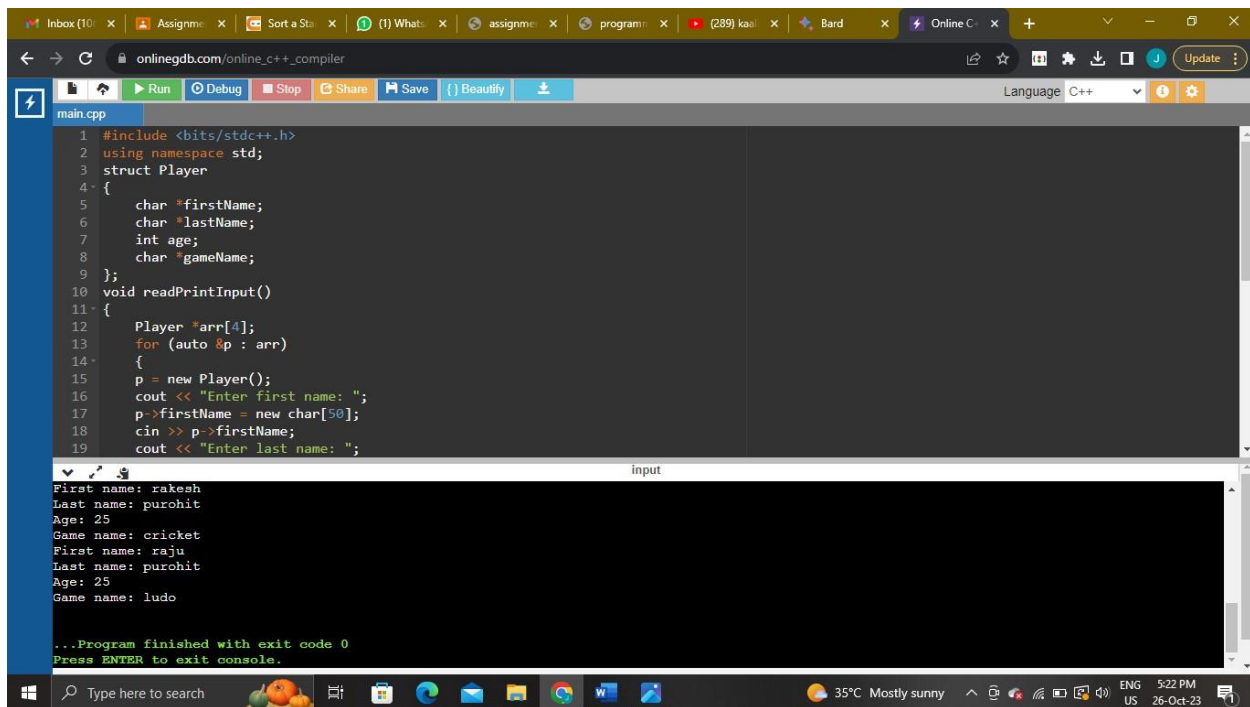
    cout << "Enter first name: ";
    p->firstName = new char[50];
    cin >> p->firstName;
    cout << "Enter last name: ";
    p->lastName = new char[50];
    cin >> p->lastName;
    cout << "Enter age: ";
    cin >> p->age;
    cout << "Enter game name: ";
    p->gameName = new char[50];
    cin >> p->gameName;
}

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();
}

```



The screenshot shows a web browser window with the URL `onlinegdb.com/online_c++_compiler`. The browser has several tabs open, including 'Inbox (10)', 'Assignme', 'Sort a St', '(1) Whats', 'assignme', 'program', '(289) kaal', 'Bard', and 'Online C'. The compiler interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C++. The code in `main.cpp` is as follows:

```
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 *arr[4];
13     for (auto &p : arr)
14     {
15         p = new Player();
16         cout << "Enter first name: ";
17         p->firstName = new char[50];
18         cin >> p->firstName;
19         cout << "Enter last name: ";
```

The output window shows the following input and output:

```
First name: rakesh
Last name: purohit
Age: 25
Game name: cricket
First name: raju
Last name: purohit
Age: 25
Game name: ludo

...Program finished with exit code 0
Press ENTER to exit console.
```

2).

Write a program that declares array of 5 pointers to strings. That is, each element of this 5 element array is a `char*`. Allocate memory to read a maximum of 200 characters for each element of the array. Read 5 names from the user using these pointers. Display all names. Now, re-allocate memory to these pointers equal to the size of the names. Display the names again to verify it is correct allocation. Don't forget to free the pointers at end of the program.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
char *arr[5];
```

```
for (auto &p : arr)
```

```
{
```

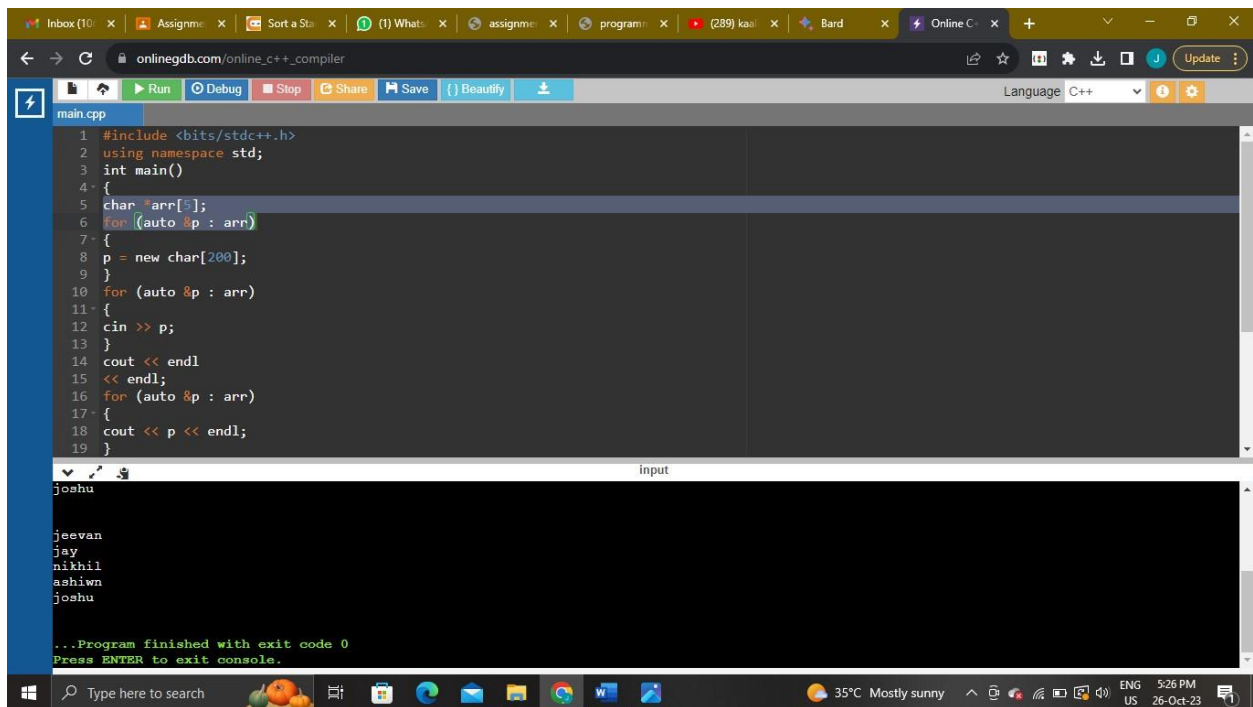
```
p = new char[200];
```

```
}
```

```
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;  
}  
}
```



3).

Declare a structure Player. It has following member variables (properties):

```
name: a pointer to string that is name of the player
```

```
age: an integer to store name
```

game: a pointer to string that stores game of the player

The program should read string inputs from user in single pointer variable named input. This input variable is pointer whose memory is allocated on heap with size of 100 characters. After user given an input (eg. for name) the program copies this string to the field of the structure (eg. the name field). Make sure only required characters are copied.

Display the name, age and game of the player after reading the data. Make sure you de-allocate all memories allocated before program exits.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
struct Player
```

 $\{$

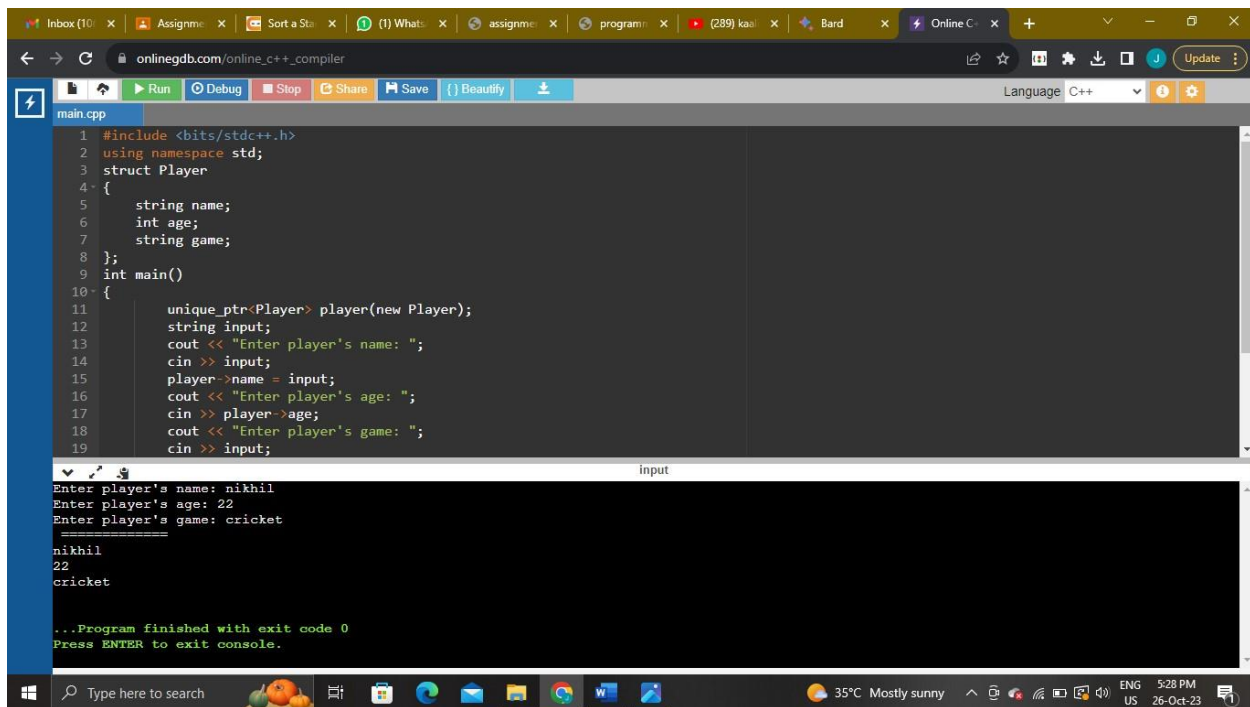
```
string name;
```

```
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;
}
```



The screenshot shows a web browser with multiple tabs. The active tab is 'onlinegdb.com/online_c++_compiler'. The code editor displays a C++ program in 'main.cpp' that defines a 'Player' struct and a 'main' function. The program prompts the user for a player's name, age, and game. The console output shows the program running successfully with the input: 'nikhil', '22', and 'cricket'. The program finishes with exit code 0.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 struct Player
4 {
5     string name;
6     int age;
7     string game;
8 };
9 int main()
10 {
11     unique_ptr<Player> player(new Player);
12     string input;
13     cout << "Enter player's name: ";
14     cin >> input;
15     player->name = input;
16     cout << "Enter player's age: ";
17     cin >> player->age;
18     cout << "Enter player's game: ";
19     cin >> input;
```

input

```
Enter player's name: nikhil
Enter player's age: 22
Enter player's game: cricket

nikhil
22
cricket

...Program finished with exit code 0
Press ENTER to exit console.
```

4).

Write a program that reads the $N \times M$ dimensions of a two-dimensional array of doubles. If user enters 0 or less for either N or M , the program exits. After reading the size of array from user, i.e. N for rows and M for columns, it allocates memory as required. Then, it reads the numbers from user and puts these in appropriate 'cells' for the matrix. It then displays the numbers read and free's up memory before exiting.

```
#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;
}

```

```

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."
13        << endl;
14        return 1;
15    }
16    double **array = new double *[N];
17    for (int i = 0; i < N; i++)

```

input

```

Enter element at row 1 column 2: 4
Enter element at row 1 column 3: 5
Enter element at row 2 column 0: 1
Enter element at row 2 column 1: 3
Enter element at row 2 column 2: 8
Enter element at row 2 column 3: 4
The array you entered is:
4 6 1 2
8 9 4 5
1 3 8 4
...Program finished with exit code 0
Press ENTER to exit console.

```

5).

Create a max function template and use it for int, std::string, and Student type with age and ID as members.

```

#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 = "JEEVAN";
    string str2 = "NIKHIL";
    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;

}

```

The screenshot shows a web browser window with the URL `onlinegdb.com/online_c++_compiler`. The code editor contains the following C++ code:

```

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 struct Student
14 {
15     int age;
16     string id;
17 };

```

The input field contains the text "100 NIKHIL". The output field shows the result of the program execution:

```

100
NIKHIL
100 NIKHIL

...Program finished with exit code 0
Press ENTER to exit console.

```

6). Build a stack class using templates and use it for int, std::string, and Student type with age and ID as members.

```

#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()

```



```

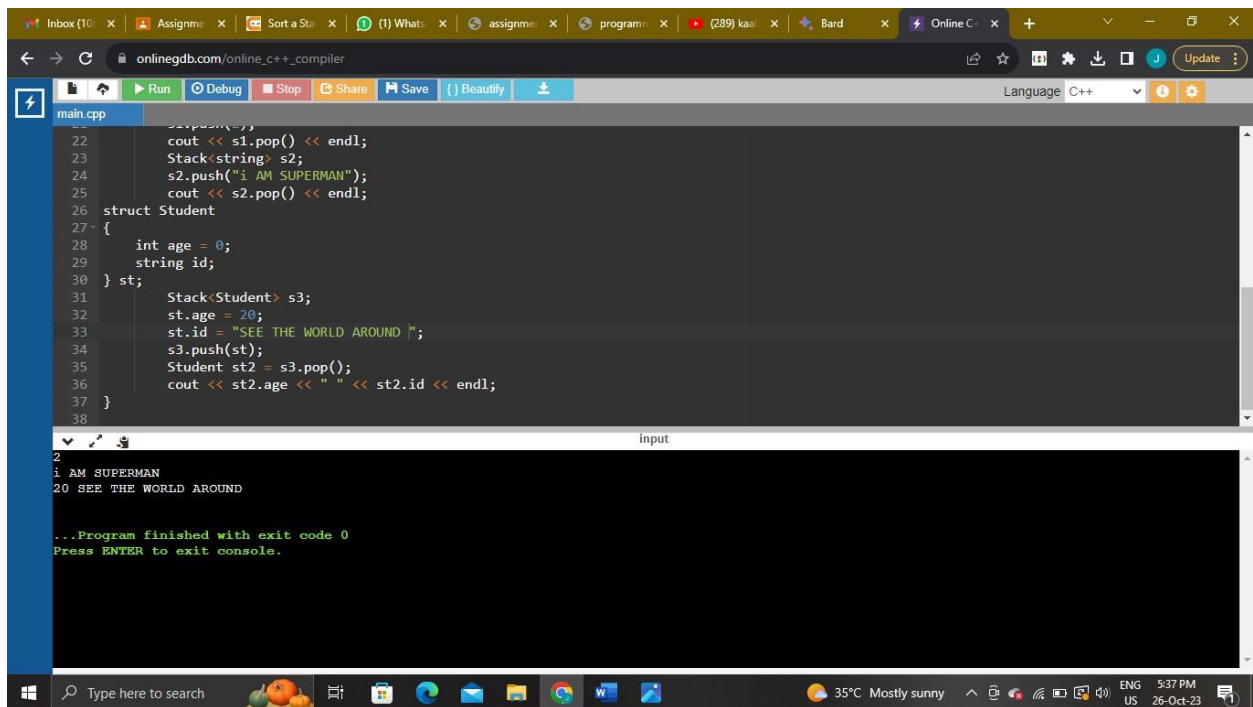
{
    T element = elements[--top];
return element;
}

};

int main()
{
    Stack<int> s1;
    s1.push(2);
    cout << s1.pop() << endl;
    Stack<string> s2;
    s2.push("i AM SUPERMAN");
    cout << s2.pop() << endl;
struct Student
{
    int age = 0;
    string id;
} st;

    Stack<Student> s3;
    st.age = 20;
    st.id = "SEE THE WORLD AROUND ";
    s3.push(st);
    Student st2 = s3.pop();
    cout << st2.age << " " << st2.id << endl;
}

```



The screenshot shows a web browser window with multiple tabs. The active tab is 'onlinegdb.com/online_c++_compiler'. The interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The main area displays a C++ program in 'main.cpp' with line numbers 22 to 38. The code uses a stack and a struct named 'Student'. Below the code editor is an 'input' section and an output console. The console shows the program's output, including 'i AM SUPERMAN' and '20 SEE THE WORLD AROUND', followed by a message indicating the program finished with exit code 0.

```
22     cout << s1.pop() << endl;
23     Stack<string> s2;
24     s2.push("i AM SUPERMAN");
25     cout << s2.pop() << endl;
26     struct Student
27     {
28         int age = 0;
29         string id;
30     } st;
31     Stack<Student> s3;
32     st.age = 20;
33     st.id = "SEE THE WORLD AROUND ";
34     s3.push(st);
35     Student st2 = s3.pop();
36     cout << st2.age << " " << st2.id << endl;
37 }
38
```

input

2
i AM SUPERMAN
20 SEE THE WORLD AROUND

...Program finished with exit code 0
Press ENTER to exit console.

Q7 and Q8 were removed by SIR.