PART-A: Warm-Up Tasks (45 Minutes)

Task 1: Create a Simple Struct

Write a C++ program that defines a struct named Student with the fields: name, age, and cgpa.
Create one Student object with sample data and print all the values.

Sample Code:

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

struct Student {
    string name;
    int age;
    float cgpa;
};

int main() {
    Student s1 = {"Ali", 18, 3.1};
    cout << s1.name << " " << s1.age << " " << s1.cgpa;
}
```

Task 2: Input Data into a Struct

Modify Task 1 so the program asks the user to input name, age, and CGPA instead of using fixed values.
Store this data in a Student struct and display it.

Task 3: Array of Structs

Create an array of 3 Student objects. Use a loop to:

- Input data for each student
- Display all students

SOLUTION:

```
#include<iostream>
using namespace std;
struct student {
                                string name;
                                int age;
                                float cgpa;
                                };
int main(){

            // Task 1
            student s1 = {"kamran",15,3.2};
            cout<<s1.name<<endl;
            cout<<s1.age<<endl;
            cout<<s1.cgpa<<endl;

            // Task 2
            student s2;
            cout<<"\nenter name: ";
            cin>>s2.name;
            cout<<"enter age: ";
            cin>>s2.age;
            cout<<"enter cgpa: ";
            cin>>s2.cgpa;

            cout<<s2.name<<endl;
            cout<<s2.age<<endl;
            cout<<s2.cgpa<<endl;

       // Task 3
       student std[3];
       for(int i=0;i<3;i++){
                                cout<<"Enter name: ";
                                cin>>std[i].name;
                                cout<<"Enter age: ";
                                cin>>std[i].age;
                                cout<<"Enter cgpa: ";
                                cin>>std[i].cgpa;
                                }
       for(int i=0;i<3;i++){
                                cout<<std[i].name<<" ";
                                cout<<std[i].age<<" ";
                                cout<<std[i].cgpa<<" ";
                                cout<<endl;
                                }
 cout<<endl;
 system("PAUSE");
 return 0;
}
```

OUTPUT:

```
kamran
15
3.2

.....Task 2.....

enter name: ali
enter age: 22
enter cgpa: 3
ali
22
3

.....Task 3.....
Enter name: muhammad
Enter age: 23
Enter cgpa: 3.4
Enter name: baqa
Enter age: 34
Enter cgpa: 4
Enter name: ahmad
Enter age: 25
Enter cgpa: 2.8
muhammad 23 3.4
baqa 34 4
ahmad 25 2.8

Press any key to continue . . .
```

PART-B: Core Tasks (1 Hour)

Task 4: Passing Structs to Functions

Write the following functions:

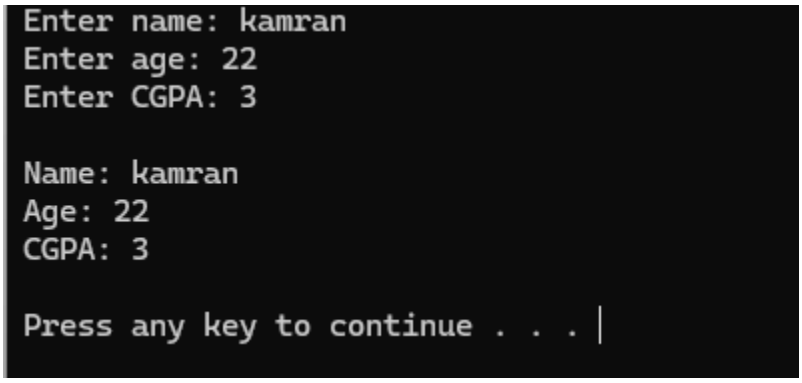
1. `Student inputStudent()` – Reads student details and returns a struct.
2. `void printStudent(Student s)` – Prints the struct.

In main(), call these functions to input a student and display the result.

SOLUTION:

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

struct student {
                                string name;
                                int age;
                                float cgpa;
                                };
student input_student(){
                        student std;
               cout<<"Enter name: ";
                                cin>>std.name;
                                cout<<"Enter age: ";
                                cin>>std.age;
                                cout<<"Enter CGPA: ";
                                cin>>std.cgpa;
       return std;
       }
void print_student(student std){
                    cout<<endl;
                                cout<<"Name: "<<std.name<<endl;
                                cout<<"Age: "<<std.age<<endl;
                                cout<<"CGPA: "<<std.cgpa<<endl;
                                }
int main(){
           student x=input_student();
           print_student(x);
 cout<<endl;
 system("PAUSE");
 return 0;}
```

OUTPUT:



Task 5: Searching in Struct Array

Create an array of 5 Student objects.

Write a function int searchByName(Student arr[], int n, string key) that:

- Searches the array by name
- Returns the index if found, or -1 if not found

Display appropriate messages in main().

SOLUTION:

```
#include<iostream>
using namespace std;
struct student{
                  string name;
                  int age;
                  float cgpa;
                  };

int search_by_name(student arr[2],string key){
                      bool found = false;
                      int ind = -3;
                      for(int i=0;i<2;i++){
                                              if(arr[i].name == key){

found=true;

ind = i;
                                              }
                      }
                      if(found){
                                                    return ind;
                                                    }

                      else
                      return -1;
                      }
int main(){
            student array[2];
            array[0].name = "kamran";
            array[0].age = 18;
            array[0].cgpa = 3.2;
```

```
            array[1].name = "ali";
            array[1].age = 22;
            array[1].cgpa = 3.8;

            string n;
            cout<<"Enter name to search: ";
            cin>>n;
            int x;
          x= search_by_name(array,n);

            if(x==-1){
                                        cout<<"Name not found";
                                        }
            else
            cout<<"Name found at "<<x<<" index";
 cout<<endl;
 cout<<endl;
 system("PAUSE");
 return 0;
}
```

OUTPUT:

```
Enter name to search: ali
Name found at 1 index

Press any key to continue . . .
```

Task 6: Find Student with Highest CGPA

Write a function Student findTopper(Student arr[], int n) that returns the student with the highest CGPA.
Display the topper's name and CGPA.

SOLUTION:

```
#include<iostream>
using namespace std;
struct student{
                  string name;
                  int age;
                  float cgpa;
                  };

int search_by_name(student arr[4]){
                      int ind = 0;
                      float high = 0.0;
                      for(int i=0;i<=3;i++){
                                              if(arr[i].cgpa > high){

ind = i;

high=arr[i].cgpa;
```

```
                                                }
                                                }
                        return ind;
                                                                }
void print_topper(student arr[4],int a){
            cout<<"Name of topper: "<<arr[a].name<<endl;
            cout<<"CGPA: "<<arr[a].cgpa;
            }

int main(){
           student array[4];
           array[0].name = "kamran";
           array[0].age = 18;
           array[0].cgpa = 3.9;

           array[1].name = "ali";
           array[1].age = 22;
           array[1].cgpa = 3.7;

           array[2].name = "khan";
           array[2].age = 22;
           array[2].cgpa = 4;

           array[3].name = "muhammad";
           array[3].age = 22;
           array[3].cgpa = 4.6;

           int x;
       x= search_by_name(array);

           print_topper(array,x);
 cout<<endl;
 system("PAUSE");
 return 0;
}
```

OUTPUT:

```
Name of topper: muhammad
CGPA: 4.6
Press any key to continue . . .
```

PART-C: Final Application (1 Hour 15 Minutes)

Task 7: Mini Student Database System (Menu-Driven Program)

Develop a complete menu-driven application using structs. The Student struct should contain:

- Roll number
- Name
- CGPA

The program should show the following menu:

1. Add Student
2. Display All Students
3. Search Student by Roll Number
4. Update CGPA
5. Exit

Details:

- Add Student → Append a new student to the array
- Display All → Print all stored students
- Search → Return index or show 'Not found'
- Update CGPA → Modify CGPA of an existing student

SOLUTION:

```
#include<iostream>
using namespace std;
struct student{
                  string name;
                  int age;
                  float cgpa;
                  };
int c = 4;
void add_student(student array[20]){
            cout<<"Enter name of student: ";
            cin>>array[c].name;
            cout<<"Enter age of student: ";
            cin>>array[c].age;
            cout<<"Enter cgpa of student: ";
            cin>>array[c].cgpa;
            cout<<"Student added..."<<endl;
            c++;
            }

void print_student(student arr[20], int size){
            for(int i=0; i<size; i++){
                                    cout<<"\nName: "<<arr[i].name<<endl;
                                    cout<<"Age: "<<arr[i].age<<endl;
                                    cout<<"CGPA: "<<arr[i].cgpa<<endl;
                                    }
            }
```

```
int search_student(student arr[20], string key){
            int ind = -1;
            for(int i=0; i<20; i++){
                                    if(arr[i].name == key){
                                                                                                ind =
i;
                                                                                                }
                                    }
       return ind;

            }

void update_cgpa(student arr[20], string key){
            int ind = -1;
            float new_cgpa;
            bool updated = false;
            for(int i=0; i<20; i++){
                                    if(arr[i].name == key){

      cout<<"Enter new cgpa: ";

      cin>>new_cgpa;

      arr[i].cgpa = new_cgpa;

      cout<<"gpa updated...";

      updated = true;

      break;
                                                                                                }
                                                if(!updated){

cout<<"student not found";
                                                                                                 }

                                    }
            }


                        }
int main(){
            student array[20]= {{"kamran", 18, 3.9} , {"ali", 35, 3.7} ,
                                                                {"khan", 12, 4},
{"muhammad", 22, 4.6}};
            while(true){



            cout<<"\n\n\tMENU"<<endl<<endl;
            cout<<"1. Add new student"<<endl;
            cout<<"2. Print all students"<<endl;
            cout<<"3. Search student by name"<<endl;
            cout<<"4. update CPGA"<<endl;
            cout<<"5. EXIT"<<endl;

            int choice;
```

```
            cout<<endl<<"Enter your choice: ";
            cin>>choice;

            if(choice == 1){
                                        add_student(array);
                                        cout<<endl;
                                        }
            else if(choice ==2){
                        print_student(array, c);
                        }
        else if(choice ==3){
                        string key1;
                        cout<<endl<<"Enter name of student to search: ";
                        cin>>key1;

                        int x = search_student(array,key1);
                        if(x==-1){
                             cout<<"Student not found";
                             }
                             else
                             cout<<"student found at "<< x<<" index";
                        }
        else if(choice ==4){
                       string key2;
                        cout<<endl<<"Enter name of student: ";
                        cin>>key2;
                        update_cgpa(array,key2);
                        }
        else if(choice ==5){
                       cout<<"T H A N K   Y O U !";
                       break;
                        }




            }

 cout<<endl;
 system("PAUSE");
 return 0;
}
```

OUTPUT:

```
        MENU

1. Add new student
2. Print all students
3. Search student by name
4. update CPGA
5. EXIT

Enter your choice: 1
Enter name of student: GUL
Enter age of student: 34
Enter cgpa of student: 3.7
Student added...



        MENU

1. Add new student
2. Print all students
3. Search student by name
4. update CPGA
5. EXIT

Enter your choice: 3

Enter name of student to search: khan
student found at 2 index
```

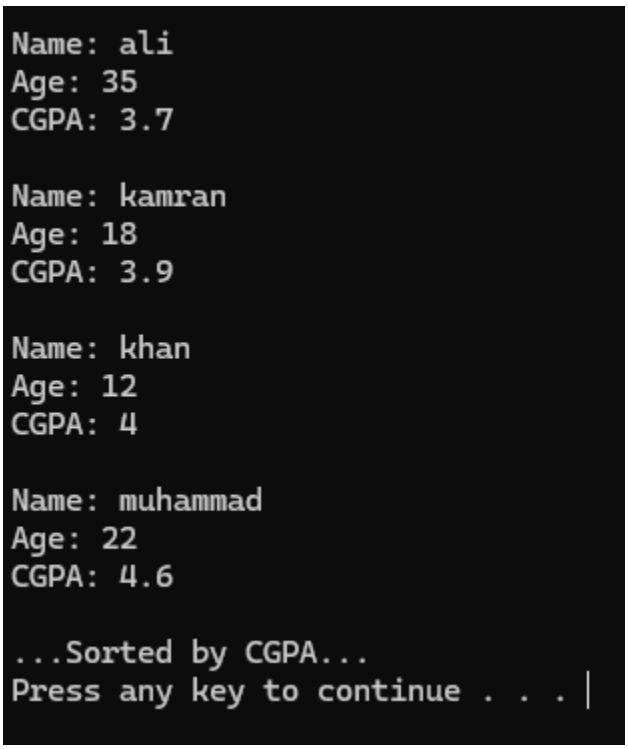
BONUS TASK (Optional)

Sort the students by CGPA using bubble sort and print the sorted list.

SOLUTION:

```
#include<iostream>
using namespace std;
struct student{
                  string name;
                  int age;
                  float cgpa;
                  };
int c = 4;
void print_student(student arr[20], int size){
            for(int i=0; i<size; i++){
                                    cout<<"\nName: "<<arr[i].name<<endl;
                                    cout<<"Age: "<<arr[i].age<<endl;
                                    cout<<"CGPA: "<<arr[i].cgpa<<endl;
                                    }
            }

void bubble_sort(student arr[20]){
                           for (int i=c-1;i>=0;i--){
                                      for (int j=0;j<i;j++){
                                                if (arr[j].cgpa >
arr[j+1].cgpa){
                                                student temp = arr[j];
                                                arr[j] = arr[j+1];
                                                arr[j+1] = temp;
                  }
            }
      }

                        }
int main(){
            student array[20]= {{"kamran", 18, 3.9} , {"ali", 35, 3.7} ,
                                                            {"khan", 12, 4},
{"muhammad", 22, 4.6}};

                        bubble_sort(array);
                        print_student(array,c);
                        cout<<endl;
                        cout<<"...Sorted by CGPA...";


 cout<<endl;
 system("PAUSE");
 return 0;
}
```

OUTPUT:


Name: ali
Age: 35
CGPA: 3.7
Name: kamran
Age: 18
CGPA: 3.9
Name: khan
Age: 12
CGPA: 4
Name: muhammad
Age: 22
CGPA: 4.6
...Sorted by CGPA...
Press any key to continue . . .