FAIZAN CHOUDHARY

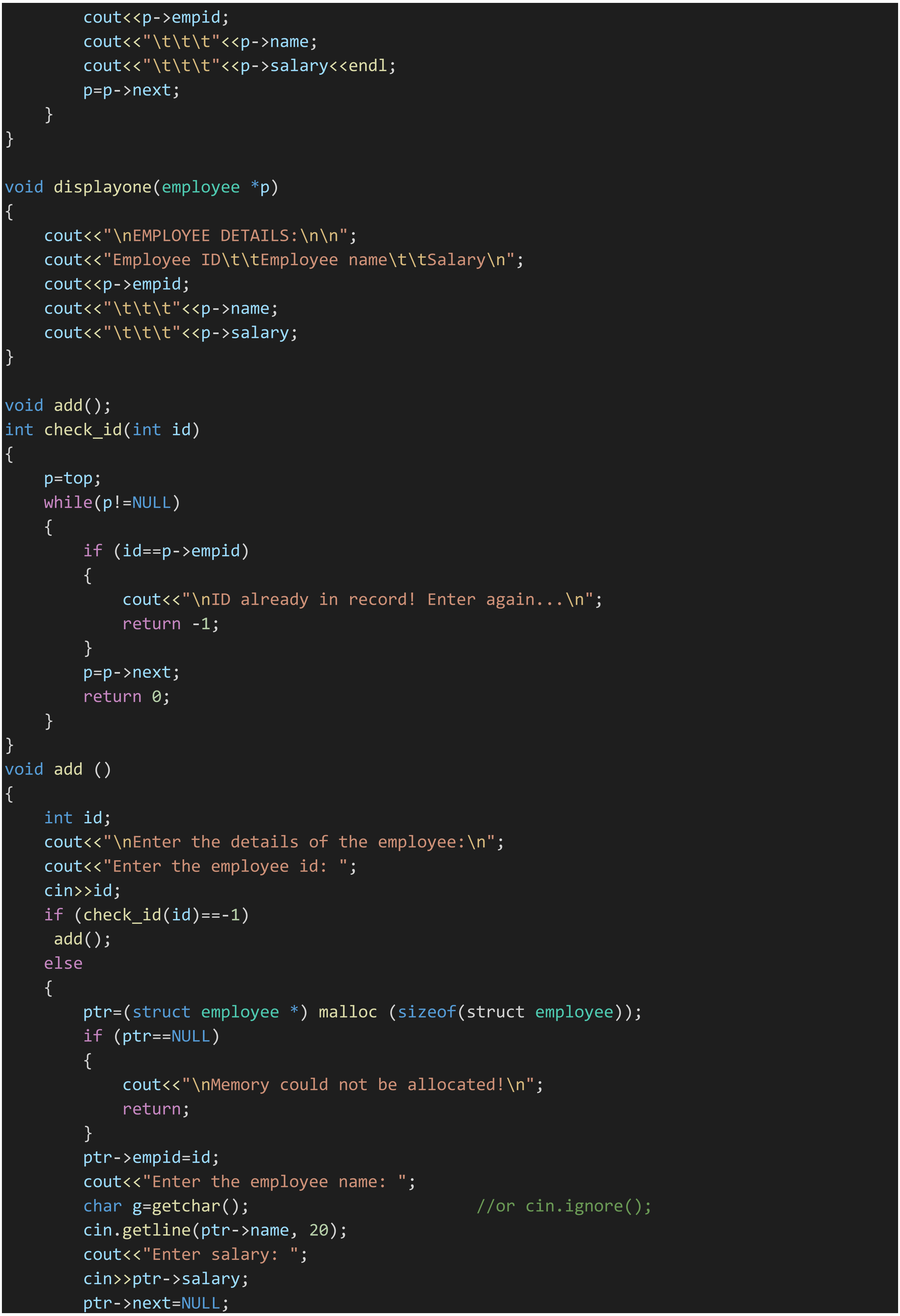
20BCS021

DSA LAB

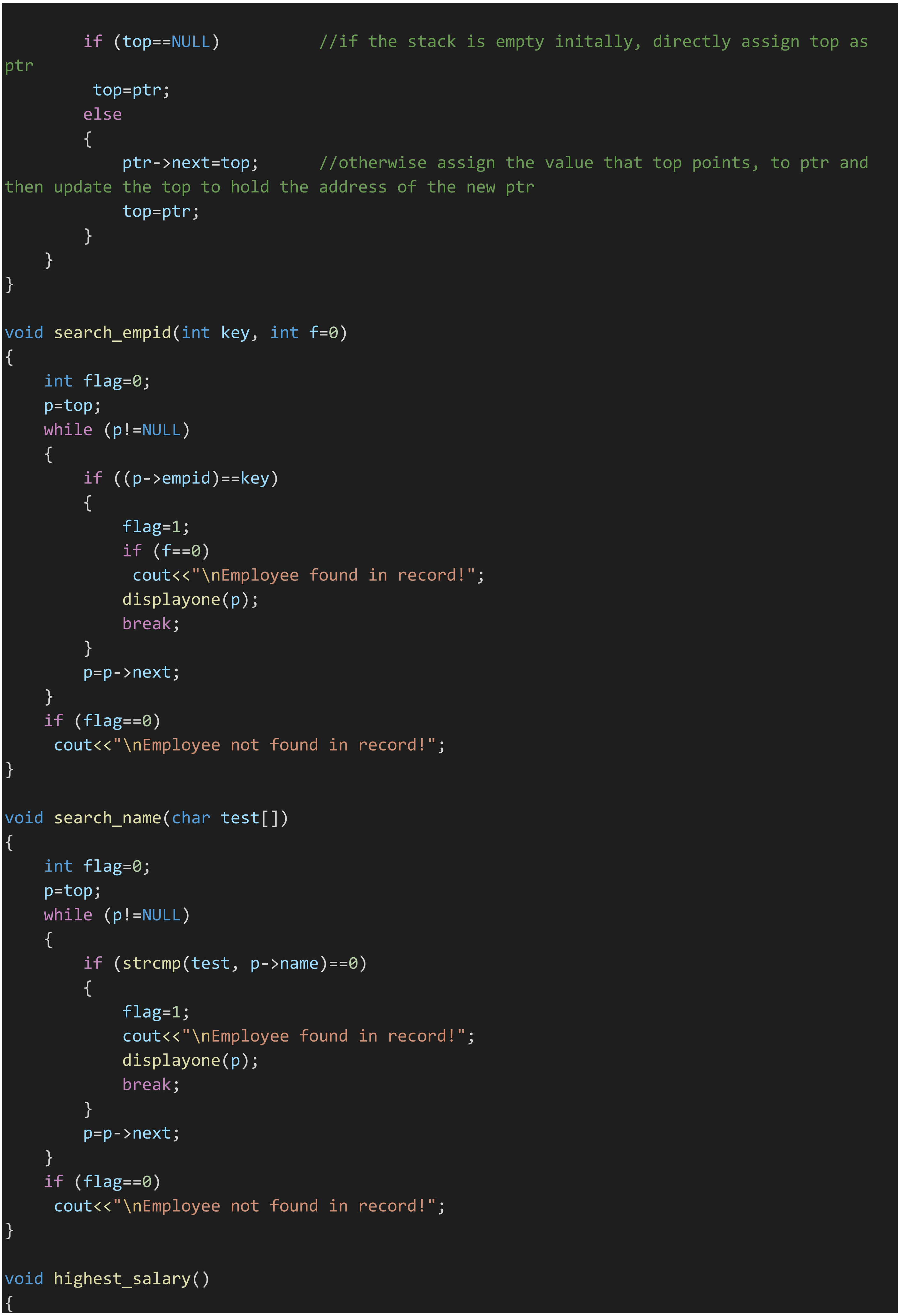
26th October 2021

CODE: (code pasted in this format for readability)

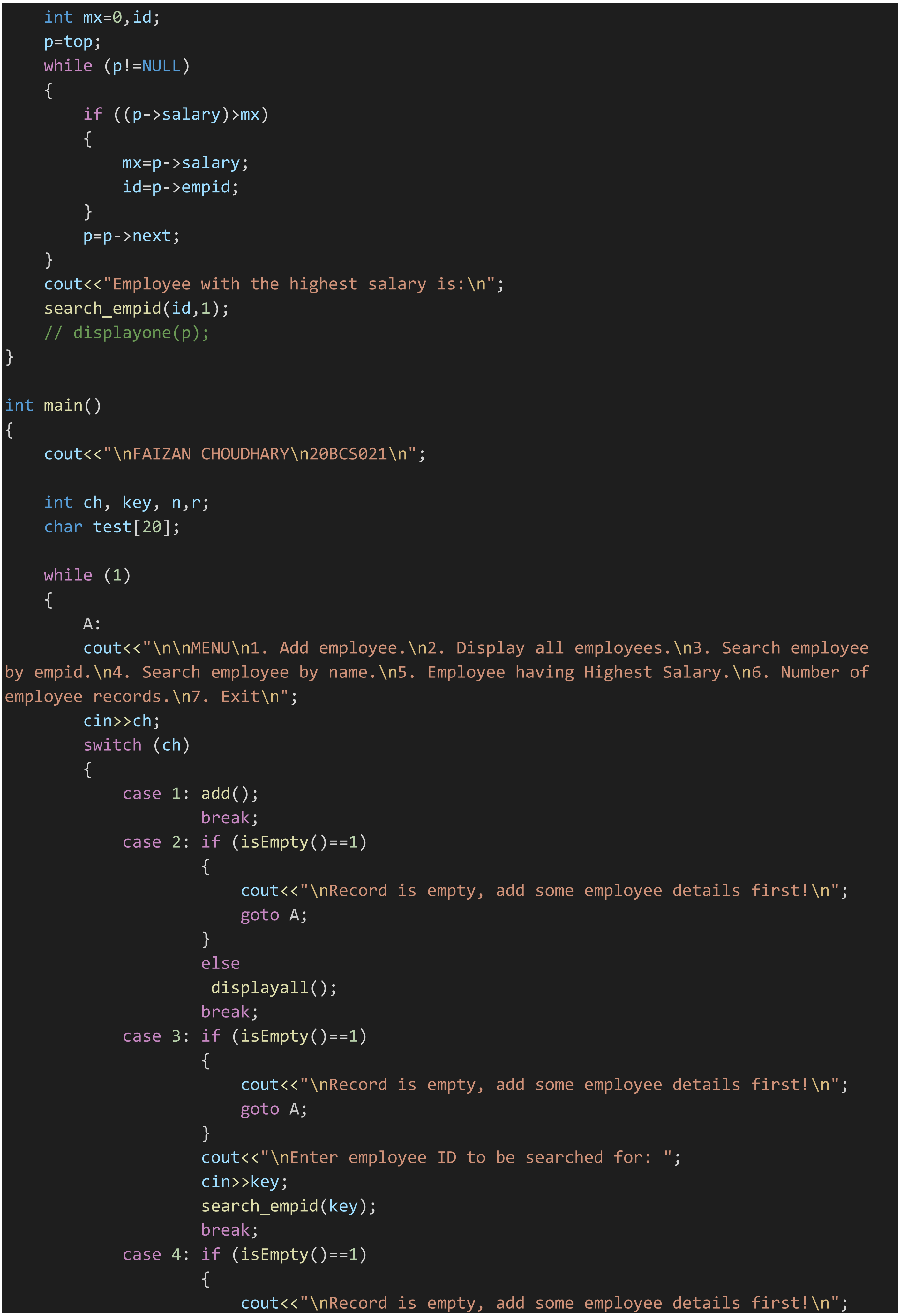
```
#include <iostream>
#include <string.h>
using namespace std;

struct employee {
    int empid;
    char name[20];
    int salary;
    struct employee *next;
};
struct employee *ptr,*top=NULL,*p;

int isEmpty ()
{
    if (top==NULL)
     return 1;
    else
     return 0;
}

int size ()
{
    if (isEmpty()==1)
     return 0;
    else
    {
        int count=0;
        for (p=top; p!=NULL; p=p->next)
         count++;
        return count;
    }
}

void displayall()
{
    cout<<"\nEMPLOYEE DETAILS:\n\n";
    cout<<"Employee ID\t\tEmployee name\t\tSalary\n";
    p=top;
    while (p!=NULL)
    {
```



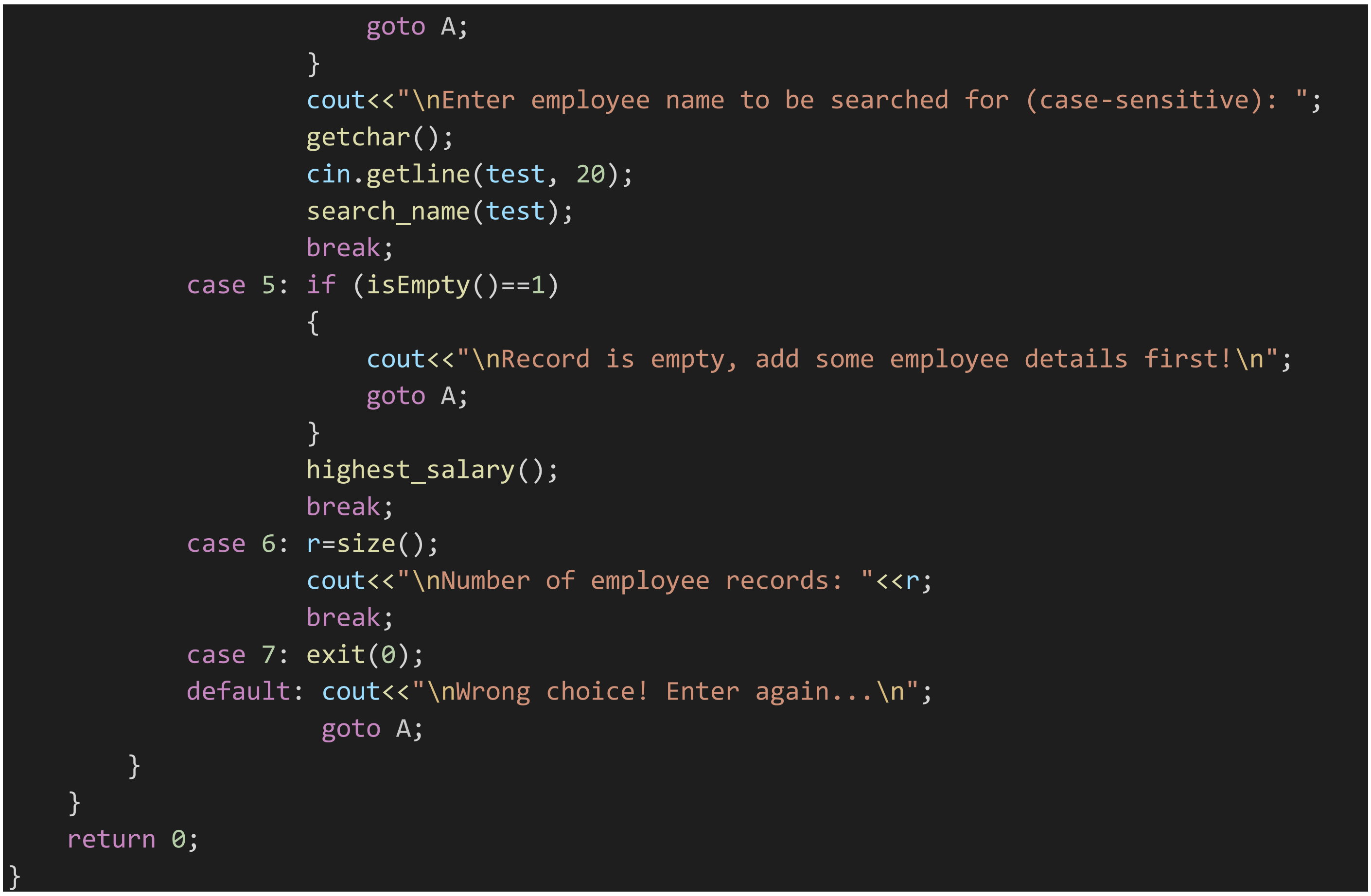
```
        cout<<p->empid;
        cout<<"\t\t\t"<<p->name;
        cout<<"\t\t\t"<<p->salary<<endl;
        p=p->next;
    }
}

void displayone(employee *p)
{
    cout<<"\nEMPLOYEE DETAILS:\n\n";
    cout<<"Employee ID\t\tEmployee name\t\tSalary\n";
    cout<<p->empid;
    cout<<"\t\t\t"<<p->name;
    cout<<"\t\t\t"<<p->salary;
}

void add();
int check_id(int id)
{
    p=top;
    while(p!=NULL)
    {
        if (id==p->empid)
        {
            cout<<"\nID already in record! Enter again...\n";
            return -1;
        }
        p=p->next;
        return 0;
    }
}
void add ()
{
    int id;
    cout<<"\nEnter the details of the employee:\n";
    cout<<"Enter the employee id: ";
    cin>>id;
    if (check_id(id)==-1)
     add();
    else
    {
        ptr=(struct employee *) malloc (sizeof(struct employee));
        if (ptr==NULL)
        {
            cout<<"\nMemory could not be allocated!\n";
            return;
        }
        ptr->empid=id;
        cout<<"Enter the employee name: ";
        char g=getchar();                       //or cin.ignore();
        cin.getline(ptr->name, 20);
        cout<<"Enter salary: ";
        cin>>ptr->salary;
        ptr->next=NULL;
```



```
        if (top==NULL)          //if the stack is empty initally, directly assign top as
ptr
         top=ptr;
        else
        {
            ptr->next=top;      //otherwise assign the value that top points, to ptr and
then update the top to hold the address of the new ptr
            top=ptr;
        }
    }
}

void search_empid(int key, int f=0)
{
    int flag=0;
    p=top;
    while (p!=NULL)
    {
        if ((p->empid)==key)
        {
            flag=1;
            if (f==0)
             cout<<"\nEmployee found in record!";
            displayone(p);
            break;
        }
        p=p->next;
    }
    if (flag==0)
     cout<<"\nEmployee not found in record!";
}

void search_name(char test[])
{
    int flag=0;
    p=top;
    while (p!=NULL)
    {
        if (strcmp(test, p->name)==0)
        {
            flag=1;
            cout<<"\nEmployee found in record!";
            displayone(p);
            break;
        }
        p=p->next;
    }
    if (flag==0)
     cout<<"\nEmployee not found in record!";
}

void highest_salary()
{
```

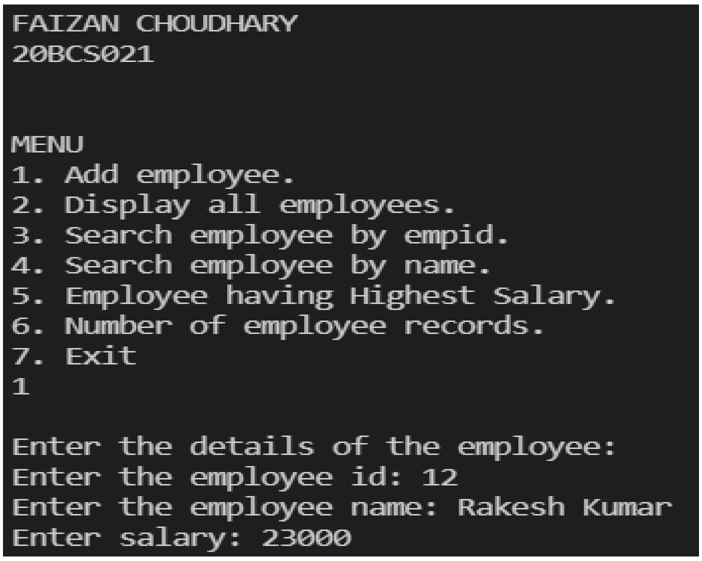


```
    int mx=0,id;
    p=top;
    while (p!=NULL)
    {
        if ((p->salary)>mx)
        {
            mx=p->salary;
            id=p->empid;
        }
        p=p->next;
    }
    cout<<"Employee with the highest salary is:\n";
    search_empid(id,1);
    // displayone(p);
}

int main()
{
    cout<<"\nFAIZAN CHOUDHARY\n20BCS021\n";

    int ch, key, n,r;
    char test[20];

    while (1)
    {
        A:
        cout<<"\n\nMENU\n1. Add employee.\n2. Display all employees.\n3. Search employee by empid.\n4. Search employee by name.\n5. Employee having Highest Salary.\n6. Number of employee records.\n7. Exit\n";
        cin>>ch;
        switch (ch)
        {
            case 1: add();
                    break;
            case 2: if (isEmpty()==1)
                    {
                        cout<<"\nRecord is empty, add some employee details first!\n";
                        goto A;
                    }
                    else
                     displayall();
                    break;
            case 3: if (isEmpty()==1)
                    {
                        cout<<"\nRecord is empty, add some employee details first!\n";
                        goto A;
                    }
                    cout<<"\nEnter employee ID to be searched for: ";
                    cin>>key;
                    search_empid(key);
                    break;
            case 4: if (isEmpty()==1)
                    {
                        cout<<"\nRecord is empty, add some employee details first!\n";
```

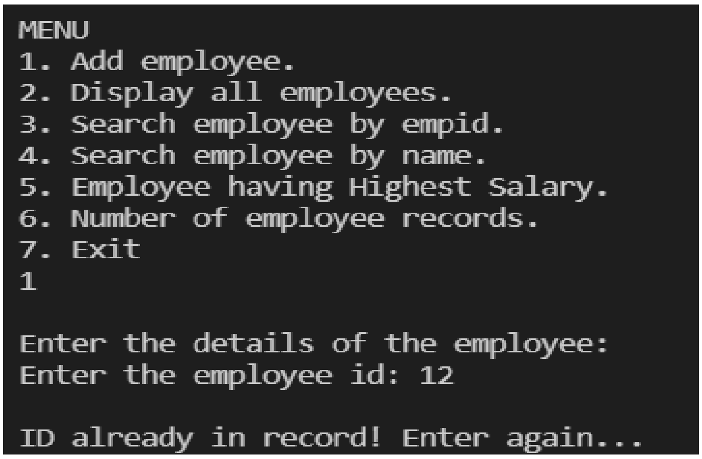


```
                        goto A;
                    }
                    cout<<"\nEnter employee name to be searched for (case-sensitive): ";
                    getchar();
                    cin.getline(test, 20);
                    search_name(test);
                    break;
            case 5: if (isEmpty()==1)
                    {
                        cout<<"\nRecord is empty, add some employee details first!\n";
                        goto A;
                    }
                    highest_salary();
                    break;
            case 6: r=size();
                    cout<<"\nNumber of employee records: "<<r;
                    break;
            case 7: exit(0);
            default: cout<<"\nWrong choice! Enter again...\n";
                     goto A;
        }
    }
    return 0;
}
```

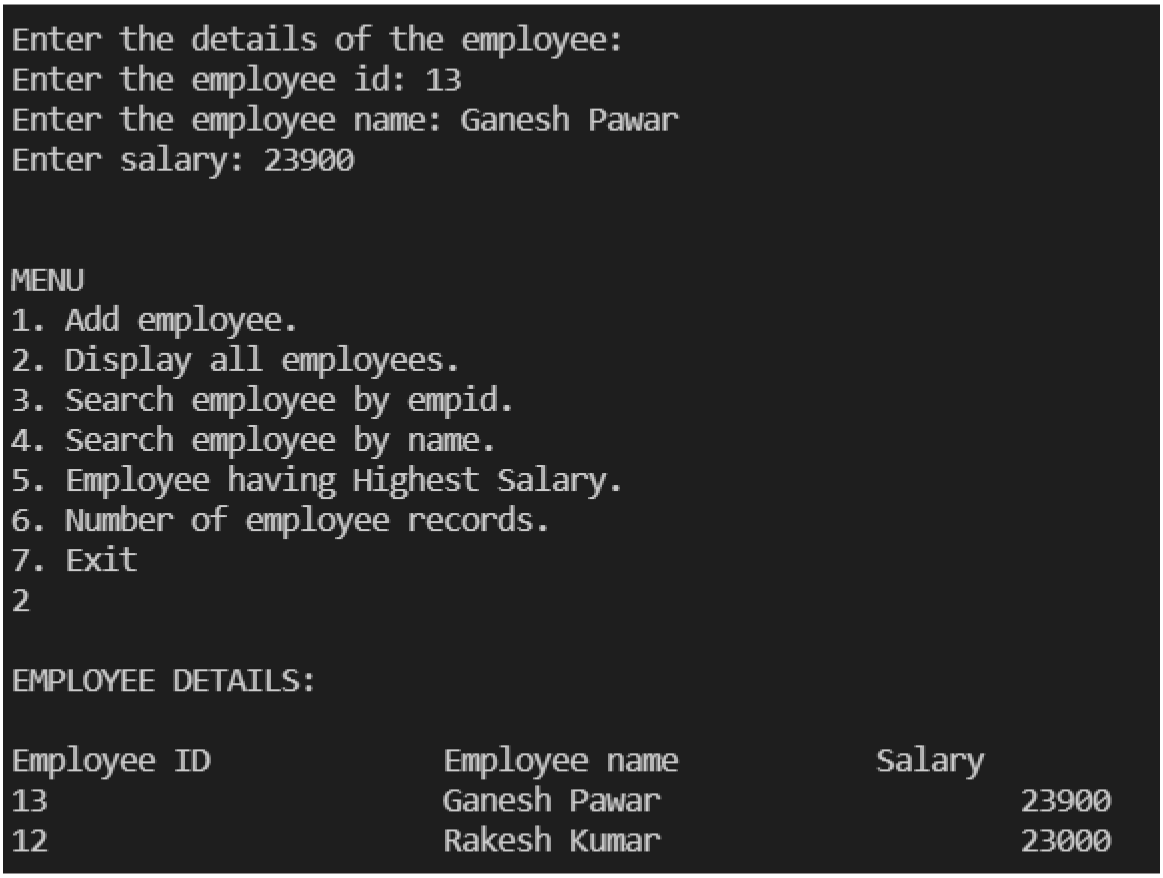
OUTPUT:



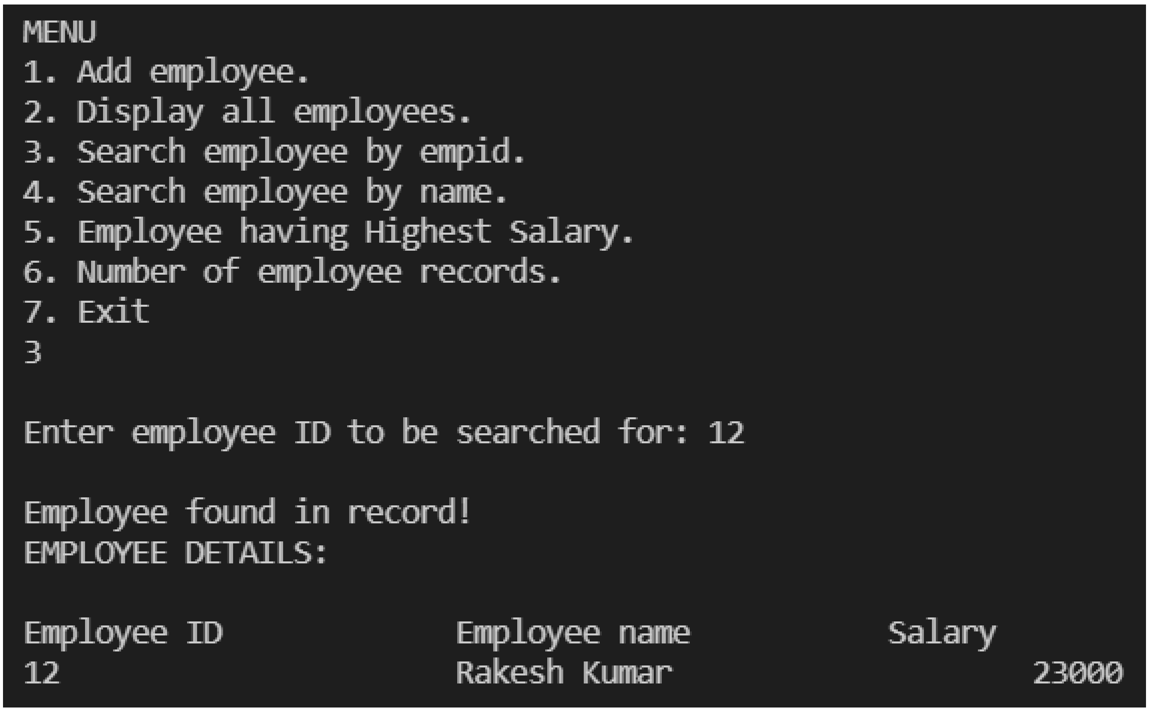
```
FAIZAN CHOUDHARY
20BCS021


MENU
1. Add employee.
2. Display all employees.
3. Search employee by empid.
4. Search employee by name.
5. Employee having Highest Salary.
6. Number of employee records.
7. Exit
1

Enter the details of the employee:
Enter the employee id: 12
Enter the employee name: Rakesh Kumar
Enter salary: 23000
```



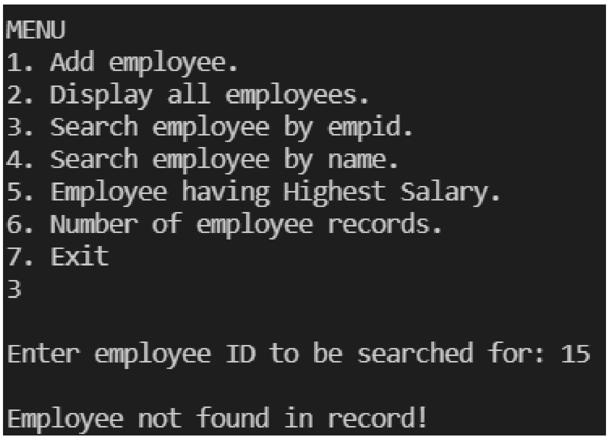
```
MENU
1. Add employee.
2. Display all employees.
3. Search employee by empid.
4. Search employee by name.
5. Employee having Highest Salary.
6. Number of employee records.
7. Exit
1

Enter the details of the employee:
Enter the employee id: 12

ID already in record! Enter again...
```



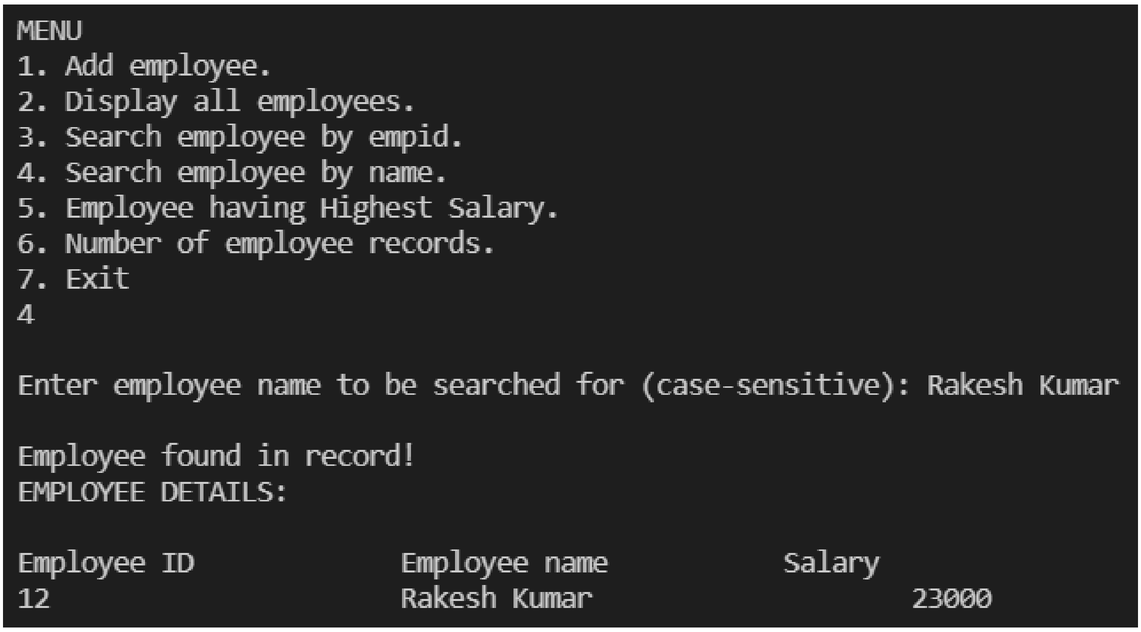
```
Enter the details of the employee:
Enter the employee id: 13
Enter the employee name: Ganesh Pawar
Enter salary: 23900


MENU
1. Add employee.
2. Display all employees.
3. Search employee by empid.
4. Search employee by name.
5. Employee having Highest Salary.
6. Number of employee records.
7. Exit
2

EMPLOYEE DETAILS:

Employee ID             Employee name           Salary
13                      Ganesh Pawar                    23900
12                      Rakesh Kumar                    23000
```



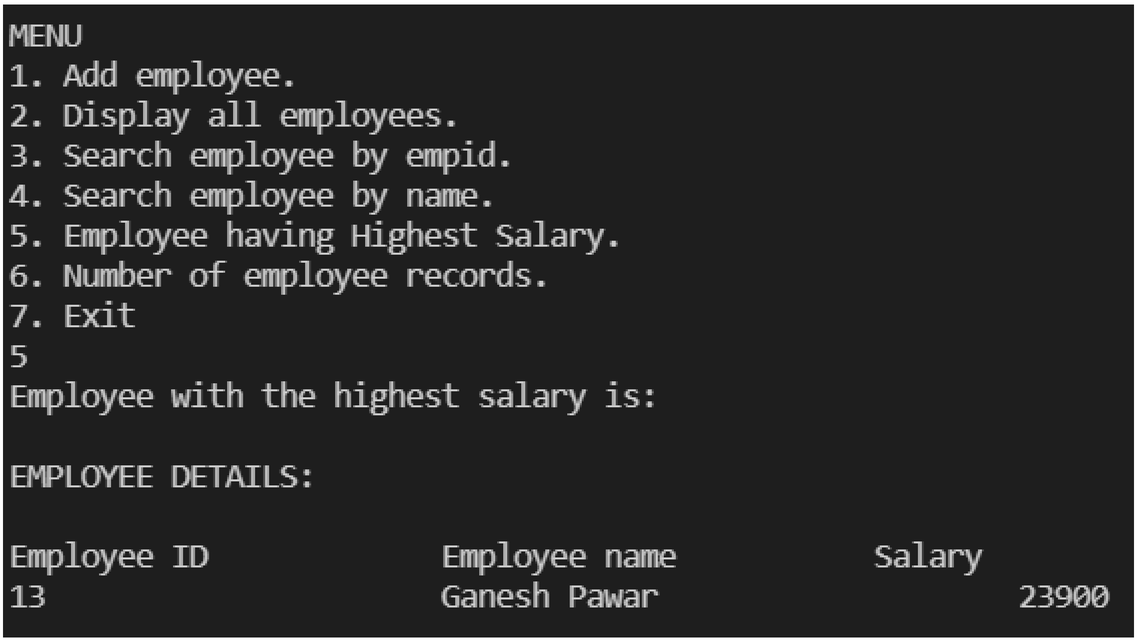
```
 MENU
 1. Add employee.
 2. Display all employees.
 3. Search employee by empid.
 4. Search employee by name.
 5. Employee having Highest Salary.
 6. Number of employee records.
 7. Exit
 3

 Enter employee ID to be searched for: 12

 Employee found in record!
 EMPLOYEE DETAILS:

 Employee ID             Employee name           Salary
 12                      Rakesh Kumar                    23000
```



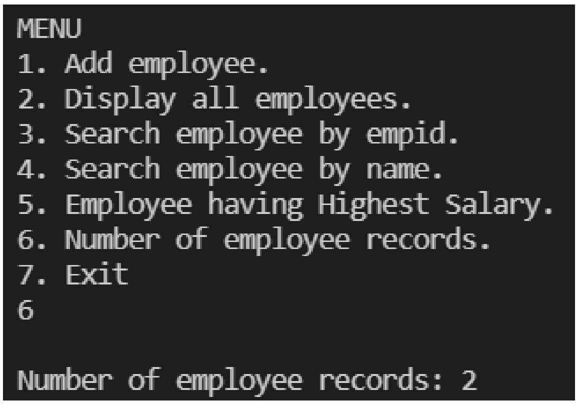
```
MENU
1. Add employee.
2. Display all employees.
3. Search employee by empid.
4. Search employee by name.
5. Employee having Highest Salary.
6. Number of employee records.
7. Exit
3

Enter employee ID to be searched for: 15

Employee not found in record!
```



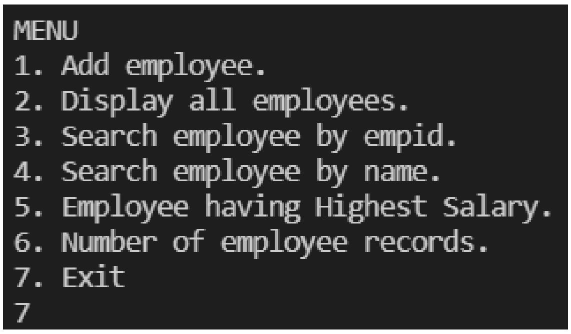
```
MENU
1. Add employee.
2. Display all employees.
3. Search employee by empid.
4. Search employee by name.
5. Employee having Highest Salary.
6. Number of employee records.
7. Exit
4

Enter employee name to be searched for (case-sensitive): Rakesh Kumar

Employee found in record!
EMPLOYEE DETAILS:

Employee ID             Employee name           Salary
12                      Rakesh Kumar                    23000
```



```
MENU
1. Add employee.
2. Display all employees.
3. Search employee by empid.
4. Search employee by name.
5. Employee having Highest Salary.
6. Number of employee records.
7. Exit
5
Employee with the highest salary is:

EMPLOYEE DETAILS:

Employee ID             Employee name           Salary
13                      Ganesh Pawar                    23900
```



```
MENU
1. Add employee.
2. Display all employees.
3. Search employee by empid.
4. Search employee by name.
5. Employee having Highest Salary.
6. Number of employee records.
7. Exit
6

Number of employee records: 2
```



```
MENU
1. Add employee.
2. Display all employees.
3. Search employee by empid.
4. Search employee by name.
5. Employee having Highest Salary.
6. Number of employee records.
7. Exit
7
```