

FAIZAN CHOUDHARY

20BCS021

DSA LAB

12th October 2021

CODE: (code pasted in this format for readability)

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

const int LIMIT=20;

struct employee {
    int empid;
    char name[20];
    int salary;
};

// struct employee *ptr = new employee[LIMIT];
struct employee *ptr = (struct employee *) malloc (LIMIT * sizeof (employee));

int count=-1;

void add();
int check_id(int id)
{
    for (int i=0; i<=count; i++)
    {
        if (id==(ptr+i)->empid)
        {
            cout<<"\nID already in record! Enter again...\n";
            count--;
            return -1;
        }
        return 0;
    }
}

void add ()
{
    int id;
    count++;
    cout<<"\nEnter the details of the employee:\n";
    cout<<"Enter the employee id: ";
    cin>>id;
    if (check_id(id)==-1)
        add();
}
```

```

else
{
    (ptr+count)->empid=id;
    cout<<"Enter the employee name: ";
    char g=getchar(); //or cin.ignore();
    cin.getline((ptr+count)->name, 20);
    cout<<"Enter salary: ";
    cin>>(ptr+count)->salary;
}
}

void display()
{
    cout<<"\nEMPLOYEE DETAILS:\n\n";
    cout<<"Employee ID\t\tEmployee name\t\tSalary\n";
    for (int i=0; i<=count; i++)
    {
        cout<<(ptr+i)->empid;
        cout<<"\t\t\t"<<(ptr+i)->name;
        cout<<"\t\t\t"<<(ptr+i)->salary<<endl;
    }
}

void search_empid(int key)
{
    int flag=0;
    for (int i=0; i<=count; i++)
    {
        if ((ptr+i)->empid==key)
        {
            flag=1;
            cout<<"\nEmployee found in record!";
            cout<<"\nEMPLOYEE DETAILS:\n\n";
            cout<<"Employee ID\t\tEmployee name\t\tSalary\n";
            cout<<(ptr+i)->empid;
            cout<<"\t\t\t"<<(ptr+i)->name;
            cout<<"\t\t\t"<<(ptr+i)->salary;
            if (flag==1)
                break;
            else
                cout<<"\nEmployee not found in record!";
        }
        if (flag==0)
        {
            cout<<"\nEmployee not found in record!";
            break;
        }
    }
}

void search_name(char test[])
{
    int flag=0;
    for (int i=0; i<=count; i++)

```

```

{
    if (strcmp(test, (ptr+i)->name)==0)
    {
        flag=1;
        cout<<"\nEmployee found in record!";
        cout<<"\nEMPLOYEE DETAILS:\n\n";
        cout<<"Employee ID\t\tEmployee name\t\tSalary\n";
        cout<<(ptr+i)->empid;
        cout<<"\t\t\t"<<(ptr+i)->name;
        cout<<"\t\t\t"<<(ptr+i)->salary;
        if (flag==1)
            break;
    }
    // if (flag==0)
    // cout<<"\nEmployee not found in record!";
}
if (flag==0)
    cout<<"\nEmployee not found in record!";
}

void highest_salary()
{
    int mx=0, index;
    for (int i=0; i<count; i++)
    {
        if ((ptr+i)->salary>mx)
        {
            mx=(ptr+i)->salary;
            index=i;
        }
    }
    cout<<"Employee with the highest salary is:\n";
    cout<<"Employee ID\t\tEmployee name\t\tSalary\n";
    cout<<(ptr+index)->empid;
    cout<<"\t\t\t"<<(ptr+index)->name;
    cout<<"\t\t\t"<<(ptr+index)->salary;
}

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

    int ch, key,n;
    char test[20], g1;

    cout<<"\nEnter the number of employees initially: ";
    cin>>n;
    // struct employee *ptr1 = new employee[n];
    struct employee *ptr1 =(struct employee *) malloc (n * sizeof (employee));
    ptr=ptr1;

    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. Exit\n";
        cin>>ch;
        switch (ch)
        {
            case 1: if (count==20)
                    {
                        cout<<"\nMaximum employee limit (20) reached!\n";
                        goto A;
                    }
                    else if (count<20)
                        add();
                    break;
            case 2: if (count==-1)
                    {
                        cout<<"\nRecord is empty, add some employee details first!\n";
                        goto A;
                    }
                    else
                        display();
                    break;
            case 3: cout<<"\nEnter employee ID to be searched for: ";
                    cin>>key;
                    search_empid(key);
                    break;
            case 4: cout<<"\nEnter employee name to be searched for (case-sensitive): ";
                    g1=getchar();
                    cin.getline(test, 20);
                    search_name(test);
                    break;
            case 5: highest_salary();
                    break;
            case 6: exit(0);
            default: cout<<"\nWrong choice! Enter again...\n";
                    goto A;
        }
    }
    return 0;
}

```

OUTPUT:

```
FAIZAN CHOUDHARY
20BCS021
```

```
Enter the number of employees initially: 3
```

```
MENU
```

1. Add employee.
 2. Display all employees.
 3. Search employee by empid.
 4. Search employee by name.
 5. Employee having Highest Salary.
 6. 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. 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. Exit
- ```
2
```

```
EMPLOYEE DETAILS:
```

| Employee ID | Employee name | Salary |
|-------------|---------------|--------|
| 12          | Rakesh Kumar  | 23000  |
| 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. 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. Exit
- 3

Enter employee ID to be searched for: 14

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. Exit
- 4

Enter employee name to be searched for (case-sensitive): Gajanan Anand

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. 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. Exit

5

Employee with the highest salary is:

| 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. Exit

6