

## Experiment F24

```
#include <bits/stdc++.h>
#define max 20
using namespace std;
struct employee {
    string name;
    long int code;
    string designation;
    int exp;
    int age;
};
int num;
void showMenu();
employee emp[max], tempemp[max],
    sortemp[max], sortemp1[max];
void build()
{
    cout << "Build The Table\n";
    cout << "Maximum Entries can be "
        << max << "\n";

    cout << "Enter the number of "
        << "Entries required";
    cin >> num;

    if (num > 20) {
        cout << "Maximum number of "
            << "Entries are 20\n";
        num = 20;
    }
    cout << "Enter the following data:\n";

    for (int i = 0; i < num; i++) {
        cout << "Name ";
        cin >> emp[i].name;
        cout << "Employee ID ";
        cin >> emp[i].code;
        cout << "Designation ";
        cin >> emp[i].designation;
        cout << "Experience ";
        cin >> emp[i].exp;
        cout << "Age ";
        cin >> emp[i].age;
    }
    showMenu();
}
```

```

void insert()
{
    if (num < max) {
        int i = num;
        num++;

        cout << "Enter the information "
             << "of the Employee\n";
        cout << "Name ";
        cin >> emp[i].name;

        cout << "Employee ID ";
        cin >> emp[i].code;

        cout << "Designation ";
        cin >> emp[i].designation;

        cout << "Experience ";
        cin >> emp[i].exp;

        cout << "Age ";
        cin >> emp[i].age;
    }
    else {
        cout << "Employee Table Full\n";
    }

    showMenu();
}

void deleteIndex(int i)
{
    for (int j = i; j < num - 1; j++) {
        emp[j].name = emp[j + 1].name;
        emp[j].code = emp[j + 1].code;
        emp[j].designation
            = emp[j + 1].designation;
        emp[j].exp = emp[j + 1].exp;
        emp[j].age = emp[j + 1].age;
    }
    return;
}

void deleteRecord()
{
    cout << "Enter the Employee ID "
         << "to Delete Record";

    int code;

```

```

        cin >> code;
        for (int i = 0; i < num; i++) {
            if (emp[i].code == code) {
                deleteIndex(i);
                num--;
                break;
            }
        }
        showMenu();
    }
    void searchRecord()
    {
        cout << "Enter the Employee"
              << " ID to Search Record";

        int code;
        cin >> code;
        for (int i = 0; i < num; i++) {
            if (emp[i].code == code) {
                cout << "Name "
                     << emp[i].name << "\n";

                cout << "Employee ID "
                     << emp[i].code << "\n";

                cout << "Designation "
                     << emp[i].designation << "\n";

                cout << "Experience "
                     << emp[i].exp << "\n";

                cout << "Age "
                     << emp[i].age << "\n";
                break;
            }
        }

        showMenu();
    }
    void showMenu()
    {
        cout << "-----"
              << "Employee"
              << " Management System"
              << "-----\n\n";

        cout << "Available Options:\n\n";
        cout << "Build Table      (1)\n";
    }
}

```

```

cout << "Insert New Entry  (2)\n";
cout << "Delete Entry      (3)\n";
cout << "Search a Record   (4)\n";
cout << "Exit              (5)\n";

int option;

// Input Options
cin >> option;

// Call function on the bases of the
// above option
if (option == 1) {
    build();
}
else if (option == 2) {
    insert();
}
else if (option == 3) {
    deleteRecord();
}
else if (option == 4) {
    searchRecord();
}
else if (option == 5) {
    return;
}
else {
    cout << "Expected Options"
         << " are 1/2/3/4/5";
    showMenu();
}
}

// Driver Code
int main()
{

    showMenu();
    return 0;
}

```