employee

Write a C program that creates an array of structures to hold the employee information below:

```
typedef struct {
   char name[40];
   char telno[40];
   int id;
   double salary;
} Employee;
```

You are required to implement the following three functions:

- The function `readin()` reads a number of persons' names and their corresponding telephone numbers, passes the data to the caller via the parameter *p*, and returns the number of names that have entered. The character `'#'` is used to indicate the end of user input.
- The function `search()` allows the user to query the array using the name field. If the name is found, the program displays the message “`Employee found at index location: x`”. The function `search()` finds the employee data of an input name `target`, and then prints the name, telephone number, id and salary on the screen. If the input name cannot be found, then it will print the error message “`Name not found`” on the screen.
- If the name is not found and the array is not full, the user can then add the name as a new record into the array. Assume that the maximum size of the array is 100. The function `addEmployee()` adds a new employee record into the array. The message “`Added at position: x`” will be printed. If the database is fill, an error message “`Database is full`” will be printed on the screen. The function returns the updated size of the array after adding the new employee record.

The prototypes of the functions are given below:

```
int readin(Employee *emp);
int search(Employee *emp, int size, char *target);
int addEmployee(Employee *emp, int size, char *target);
```

A sample program template is given below to test the functions:

```
#include <stdio.h>
#include <string.h>
#define MAX 100
typedef struct {
   char name[40];
   char telno[40];
   int id;
   double salary;
} Employee;
void printEmp(Employee *emp, int size);
int readin(Employee *emp);
int search(Employee *emp, int size, char *target);
int addEmployee(Employee *emp, int size, char *target);
int main()
{
   Employee emp[MAX];
   char name[40], *p;
   int size, choice, result;

   printf("Select one of the following options: \n");
   printf("1: readin() \n");
   printf("2: search() \n");
   printf("3: addEmployee() \n");
   printf("4: print() \n");
   printf("5: exit() \n");
   do {
      printf("Enter your choice: \n");
      scanf("%d", &choice);
      switch (choice) {
         case 1:
            size = readin(emp);
            break;
         case 2:
            printf("Enter search name: \n");
            scanf("\n");
            fgets(name, 40, stdin);
            if (p=strchr(name,'\n')) *p = '\0';
            result = search(emp,size,name);
            if (result != 1)
               printf ("Name not found! \n");
            break;
         case 3:
            printf("Enter new name: \n");
            scanf("\n");
            fgets(name, 40, stdin);
            if (p=strchr(name,'\n')) *p = '\0';
            result = search(emp,size,name);
            if (result != 1)
               size = addEmployee(emp, size, name);
            else
               printf("The new name has already existed in the database\n");
            break;
         case 4:
            printEmp(emp, size);
            break;
         default:
            break;
      }
   } while (choice < 5);
   return 0;
}
void printEmp(Employee *emp, int size)
{
   int i;

   printf("The current employee list: \n");
   if (size==0)
      printf("Empty array\n");
   else
   {
      for (i=0; i<size; i++)
         printf("%s %s %d %.2f\n",emp[i].name,emp[i].telno,emp[i].id,
                        emp[i].salary);
   }
}
int readin(Employee *emp)
{
   /* write your code here */
}
int search(Employee *emp, int size, char *target)
{
   /* write your code here */
}
int addEmployee(Employee *emp, int size, char *target)
{
   /* write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Select one of the following options:
1: readin()
2: search()
3: addEmployee()
4: print()
5: exit()
Enter your choice:
1
Enter name:
Hui Siu Cheung
Enter tel:
12345678
Enter id:
11
Enter salary:
123.45
Enter name:
#
Enter your choice:
4
The current employee list:
Hui Siu Cheung 12345678 11 123.45
Enter your choice:
5
```

(2) Test Case 2:

```
Select one of the following options:
1: readin()
2: search()
3: addEmployee()
4: print()
5: exit()
Enter your choice:
1
Enter name:
Hui Siu Cheung
Enter tel:
12345678
Enter id:
11
Enter salary:
123.45
Enter name:
Kenny B
Enter tel:
23456789
Enter id:
12
Enter salary:
1234.45
Enter name:
#
Enter your choice:
4
The current employee list:
Hui Siu Cheung 12345678 11 123.45
Kenny B 23456789 12 1234.45
Enter your choice:
2
Enter search name:
Kenny B
```

```
Employee found at index location: 1
Kenny B 23456789 12 1234.45
Enter your choice:
5
```

(3) Test Case 3:

```
Select one of the following options:
1: readin()
2: search()
3: addEmployee()
4: print()
5: exit()
Enter your choice:
1
Enter name:
Hui Siu Cheung
Enter tel:
12345678
Enter id:
11
Enter salary:
123.45
Enter name:
Kenny B
Enter tel:
23456789
Enter id:
12
Enter salary:
1234.45
Enter name:
#
Enter your choice:
4
The current employee list:
Hui Siu Cheung 12345678 11 123.45
Kenny B 23456789 12 1234.45
Enter your choice:
2
Enter search name:
Kenny BB
Name not found!
Enter your choice:
5
```

(4) Test Case 4:

```
Select one of the following options:
1: readin()
2: search()
3: addEmployee()
4: print()
5: exit()
Enter your choice:
1
Enter name:
Hui Siu Cheung
Enter tel:
12345678
Enter id:
11
Enter salary:
123.45
Enter name:
Kenny B
```

```
Enter tel:
23456789
Enter id:
12
Enter salary:
1234.45
Enter name:
#
Enter your choice:
4
The current employee list:
Hui Siu Cheung 12345678 11 123.45
Kenny B 23456789 12 1234.45
Enter your choice:
3
Enter new name:
Kenny Tan
Enter tel:
12344321
Enter id:
13
Enter salary:
2345.67
Added at position: 2
Enter your choice:
4
The current employee list:
Hui Siu Cheung 12345678 11 123.45
Kenny B 23456789 12 1234.45
Kenny Tan 12344321 13 2345.67
Enter your choice:
5
```