```c
#include <stdio.h>
#include <stdlib.h>
#define HT_SIZE 10
typedef struct {
 int key;
 } Employee;
typedef struct {
 int key;
 Employee employee;
} HashEntry;
typedef struct {
 HashEntry *table;
 int size;
} HashTable;
int hashFunction(int key, int size) {
 return key % size;
}
void initializeHashTable(HashTable *ht, int size) {
 ht->size = size;
 ht->table = (HashEntry *)malloc(size * sizeof(HashEntry));
 for (int i = 0; i < size; i++) {
 ht->table[i].key = -1;
 }
}
void insert(HashTable *ht, int key, Employee employee) {
 int index = hashFunction(key, ht->size);
 while (ht->table[index].key != -1) {
 index = (index + 1) % ht->size;
 }
 ht->table[index].key = key;
 ht->table[index].employee = employee;
}
int search(HashTable *ht, int key) {
 int index = hashFunction(key, ht->size);
 int originalIndex = index;
 while (ht->table[index].key != key && ht->table[index].key != -1) {
 index = (index + 1) % ht->size;
 if (index == originalIndex)
 return -1;
 }
 if (ht->table[index].key == key) {
 return index;
 } else {
```

```
 return -1;
 }
}
int main() {
 HashTable ht;
 initializeHashTable(&ht, HT_SIZE);
 int numEmployees;
 printf("Enter the number of employees: ");
 scanf("%d", &numEmployees);
 for (int i = 0; i < numEmployees; i++) {
 Employee emp;
 printf("Enter key for employee %d: ", i+1);
 scanf("%d", &emp.key);
 insert(&ht, emp.key, emp);
 }
 int searchKey;
 printf("Enter key to search: ");
 scanf("%d", &searchKey);
 int resultIndex = search(&ht, searchKey);
 if (resultIndex != -1) {
 printf("Employee with key %d found at index %d.\n", searchKey,
resultIndex);
 } else {
 printf("Employee with key %d not found.\n", searchKey);
 }
 return 0;
}
```

```
PS C:\Users\kadab\OneDrive\Desktop\DS> gcc ninteen.c
PS C:\Users\kadab\OneDrive\Desktop\DS> .\a.exe
Enter the number of employees: 2
Enter key for employee 1: 234
Enter key for employee 2: 567
Enter key to search: 234
Employee with key 234 found at index 4.
```