K. Nikhilsai
AP19110010218
CSE-E

2)Hashtable_symboltable

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#define SIZE 20
struct symtab
{
char *data;
int key;
};
struct symtab* hashArray[SIZE];
struct symtab* dummyItem;
struct symtab* item;
int hashCode(int key) {
return key % SIZE;
}
struct symtab *search(char* key) {
//get the hash
int hashIndex = 1;
//move in array until an empty
while(hashArray[hashIndex] != NULL) {
if(hashArray[hashIndex]->data == key)
return hashArray[hashIndex];
//go to next cell
++hashIndex;
//wrap around the table
hashIndex %= SIZE;
}
return 0;
}
void insert(int key,char* data) {
struct symtab *item = (struct symtab*) malloc(sizeof(struct symtab));
item->data = data;
item->key = key;
struct symtab *temp = (struct symtab*) malloc(sizeof(struct symtab));
temp = search(data);
if (temp == NULL){
//get the hash
int hashIndex = key;
//move in array until an empty or deleted cell
while(hashArray[hashIndex] != NULL && hashArray[hashIndex]->key != -1) {
//go to next cell
++hashIndex;
//wrap around the table
hashIndex %= SIZE;
}
hashArray[hashIndex] = item;
}
}
void display() {
int i = 0;
printf("\n\n");
printf("info    id");
for(i = 0; i<SIZE; i++) {
if(hashArray[i] != NULL)
printf("\n %d  %s",hashArray[i]->key,hashArray[i]->data);
}
printf("\n");
}
int main() {
char id[20][20];
int i = 0;
printf("enter 0 to stop inputing identifiers\n\n");
while (i < 20){
scanf("%s",&id[i]);
if (!strcmp(id[i],"0")){
break;
}
insert(i,id[i]);
i++;
}
display();
}
```