**8. Implement a C program to perform symbol table operations.**

**Code:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#define TABLE\_SIZE 10**

**typedef struct Symbol {**

**char name[50];**

**int address;**

**struct Symbol\* next;**

**} Symbol;**

**Symbol\* symbolTable[TABLE\_SIZE];**

**unsigned int hash(char\* name) {**

**unsigned int hashValue = 0;**

**while (\*name)**

**hashValue = (hashValue \* 31) + \*(name++);**

**return hashValue % TABLE\_SIZE;**

**}**

**void insertSymbol(char\* name, int address) {**

**unsigned int index = hash(name);**

**Symbol\* newSymbol = (Symbol\*)malloc(sizeof(Symbol));**

**if (!newSymbol) {**

**printf("Memory allocation failed!\n");**

**return;**

**}**

**strcpy(newSymbol->name, name);**

**newSymbol->address = address;**

**newSymbol->next = symbolTable[index];**

**symbolTable[index] = newSymbol;**

**}**

**Symbol\* searchSymbol(char\* name) {**

**unsigned int index = hash(name);**

**Symbol\* temp = symbolTable[index];**

**while (temp) {**

**if (strcmp(temp->name, name) == 0)**

**return temp;**

**temp = temp->next;**

**}**

**return NULL;**

**}**

**void displaySymbolTable() {**

**printf("Symbol Table:\n");**

**for (int i = 0; i < TABLE\_SIZE; i++) {**

**printf("[%d]: ", i);**

**Symbol\* temp = symbolTable[i];**

**while (temp) {**

**printf("(%s, %d) -> ", temp->name, temp->address);**

**temp = temp->next;**

**}**

**printf("NULL\n");**

**}**

**}**

**int main() {**

**insertSymbol("x", 100);**

**insertSymbol("y", 200);**

**insertSymbol("z", 300);**

**insertSymbol("a", 400);**

**insertSymbol("b", 500);**

**displaySymbolTable();**

**char searchKey[50];**

**printf("\nEnter symbol to search: ");**

**scanf("%s", searchKey);**

**Symbol\* result = searchSymbol(searchKey);**

**if (result)**

**printf("Symbol found: %s at address %d\n", result->name, result->address);**

**else**

**printf("Symbol not found!\n");**

**return 0;**

**}**

