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

**Aim:**

To implement a symbol table using C that performs insertion, searching, and display operations.

**Algorithm:**

**1**. Insertion

* Accept the symbol (identifier name) and its attributes (e.g., address, type).
* Check if the symbol already exists in the table.
* If not found, add it to the table.

2. Search

* Accept the symbol to be searched.
* Traverse the table to find a match.
* If found, return the symbol’s details.
* If not found, display an error message.

3. Display

* Print all symbols and their attributes stored in the table.

**Code:**

#include <stdio.h>

#include <string.h>

#define MAX 100 // Maximum number of symbols

// Structure to store a symbol

struct Symbol {

char name[50];

char type[20];

int address;

} table[MAX];

int count = 0; // Count of symbols

// Function to insert a symbol

void insert() {

if (count >= MAX) {

printf("Symbol Table is Full!\n");

return;

}

char name[50], type[20];

int address, i;

printf("Enter Symbol Name: ");

scanf("%s", name);

printf("Enter Type: ");

scanf("%s", type);

printf("Enter Address: ");

scanf("%d", &address);

// Check if symbol already exists

for (i = 0; i < count; i++) {

if (strcmp(table[i].name, name) == 0) {

printf("Error: Symbol already exists!\n");

return;

}

}

// Insert the new symbol

strcpy(table[count].name, name);

strcpy(table[count].type, type);

table[count].address = address;

count++;

printf("Symbol Inserted Successfully!\n");

}

// Function to search for a symbol

void search() {

char name[50];

printf("Enter Symbol Name to Search: ");

scanf("%s", name);

for (int i = 0; i < count; i++) {

if (strcmp(table[i].name, name) == 0) {

printf("Symbol Found: Name: %s, Type: %s, Address: %d\n",

table[i].name, table[i].type, table[i].address);

return;

}

}

printf("Symbol Not Found!\n");

}

// Function to display the symbol table

void display() {

if (count == 0) {

printf("Symbol Table is Empty!\n");

return;

}

printf("\nSymbol Table:\n");

printf("---------------------------------------------------\n");

printf("Index\tName\t\tType\t\tAddress\n");

printf("---------------------------------------------------\n");

for (int i = 0; i < count; i++) {

printf("%d\t%-10s\t%-10s\t%d\n", i + 1, table[i].name, table[i].type, table[i].address);

}

printf("---------------------------------------------------\n");

}

// Main function

int main() {

int choice;

while (1) {

printf("\nSymbol Table Operations:\n");

printf("1. Insert Symbol\n");

printf("2. Search Symbol\n");

printf("3. Display Symbol Table\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1: insert(); break;

case 2: search(); break;

case 3: display(); break;

case 4: return 0;

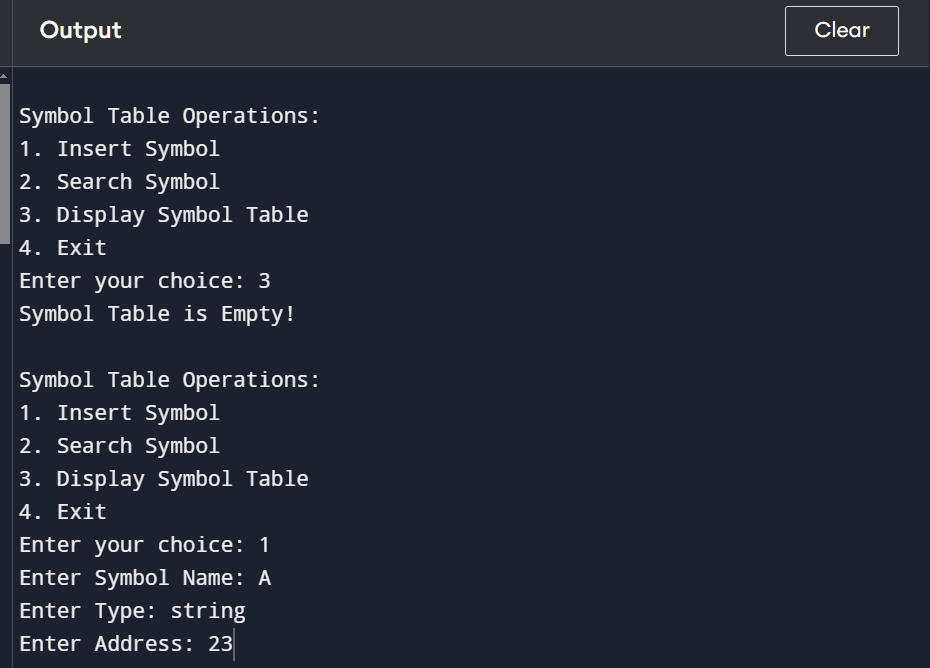
default: printf("Invalid Choice! Try Again.\n");

}

}

}

**OUTPUT:**

****