

Name: Nimish Kushwaha

Reg. No.: CH.EN.U4CSE22073

Lab Exp.: 05

Aim: To implement symbol table.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

int main() {
    int x = 0, i = 0, j = 0;
    void *T4Tutorials_address[50]; // Symbol addresses
    char T4Tutorials_Array2[50]; // Input expression
    char T4Tutorials_Array3[50]; // Symbols stored
    char c;

    printf("Input the expression ending with $ sign: ");
    while ((c = getchar()) != '$') {
        T4Tutorials_Array2[i++] = c;
    }
    int n = i - 1;

    // Display the entered expression
    printf("\nGiven Expression: ");
    for (i = 0; i <= n; i++) {
        printf("%c", T4Tutorials_Array2[i]);
    }

    // Display Symbol Table
    printf("\n\nSymbol Table display\n");
    printf("Symbol \t Address \t Type\n");

    for (j = 0; j <= n; j++) {
        c = T4Tutorials_Array2[j];
        if (isalpha(c)) {
            // Allocate memory for identifier (1 byte per char)
            void *mypointer = malloc(sizeof(char));
            T4Tutorials_address[x] = mypointer;
```

```

T4Tutorials_Array3[x] = c;
printf("%c \t %p \t identifier\n", c, mypointer);
x++;
} else if (c == '+' || c == '-' || c == '*' || c == '=') {
    // Allocate memory for operator (1 byte)
    void *mypointer = malloc(sizeof(char));
    T4Tutorials_address[x] = mypointer;
    T4Tutorials_Array3[x] = c;
    printf("%c \t %p \t operator\n", c, mypointer);
    x++;
}
}

// Free allocated memory
for (i = 0; i < x; i++) {
    free(T4Tutorials_address[i]);
}

return 0;
}

```

Output:

```

asecomputerlab@ubuntu:~/22073$ gcc lab5.c
asecomputerlab@ubuntu:~/22073$ ./a.out
Input the expression ending with $ sign: a+b*c$

Given Expression: a+b*c

Symbol Table display
Symbol  Address          Type
a       0x5604e4856a80   identifier
+       0x5604e4856aa0   operator
b       0x5604e4856ac0   identifier
*       0x5604e4856ae0   operator
c       0x5604e4856b00   identifier
asecomputerlab@ubuntu:~/22073$

```