

Date: 8/09/2025

Experiment No: 05

Aim: To implement Symbol Table.

Code:

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

int main(void) {
    int x = 0, i = 0, j = 0;
    char ch, c;
    char input_expr[15];
    char symbols[15]; // To store symbols (identifiers/operators)
    void *addresses[15]; // Corresponding memory addresses

    printf("Input the expression ending with $ sign: ");

    // Read expression until '$'
    while ((c = getchar()) != '$' && i < 14) {
        input_expr[i++] = c;
    }
    input_expr[i] = '\0';

    printf("Given Expression: %s\n", input_expr);

    printf("\nSymbol Table display\n");
    printf("Symbol\tAddress\tType\n");

    while (j < i) {
        c = input_expr[j];
        if (isalpha(c)) {
            // Identifier
            void *ptr = malloc(sizeof(char));
            symbols[x] = c;
            addresses[x] = ptr;
            printf("%c\t%p\tidentifier\n", c, ptr);
            x++;
        } else if (c == '+' || c == '-' || c == '*' || c == '=') {
            // Operator
            void *ptr = malloc(sizeof(char));
            symbols[x] = c;
            addresses[x] = ptr;
            printf("%c\t%p\toperator\n", c, ptr);
            x++;
        }
        j++;
    }

    // Free allocated memory

    j++;
}

// Free allocated memory
for (int k = 0; k < x; k++) {
    free(addresses[k]);
}

return 0;
}
```

Output:

```
asecomputerlab@asecomputerlab:~$ gcc ex5.c
asecomputerlab@asecomputerlab:~$ ./a.out
Input the expression ending with $ sign: a=b+c*d$
Given Expression: a=b+c*d

Symbol Table display
Symbol  Address      Type
a       0x561218937a80 identifier
=       0x561218937aa0 operator
b       0x561218937ac0 identifier
+       0x561218937ae0 operator
c       0x561218937b00 identifier
*       0x561218937b20 operator
d       0x561218937b40 identifier
asecomputerlab@asecomputerlab:~$
```

Results: The program for symbol table implementation has been executed successfully.