

10. Write a program that can parse a simple arithmetic expression of type (x op y op z). The program can make use of all the available string manipulation functions to parse the string efficiently. The arithmetic expression is given as an input by the user. 'x', 'y', 'z' are all integers. 'op' can be one of '+', '-', '*', '/'. (the syntax for the expression has the space character between any operand and operator)

Description:

This program **parses and evaluates a simple arithmetic expression** of the form:

x op y op z

where:

- x, y, and z are **integers**.
- op is an **operator** (+, -, *, or /).
- The expression contains **spaces** between numbers and operators.

Example:

Enter an expression: 5 + 3 * 2

Result: 11

Algorithm:

Step 1: Start

Step 2: Declare a string variable to store the user-input expression.

Step 3: Prompt the user to enter an arithmetic expression in the format x op y op z (with spaces between numbers and operators).

Step 4: Read the input string.

Step 5: Parse the string to extract:

- The first integer (x)
- The first operator (op1)
- The second integer (y)
- The second operator (op2)
- The third integer (z)

Step 6: Perform the calculations based on operator precedence:

- If op2 is * or /, evaluate y op2 z first, then compute x op1 result.
- Otherwise, compute x op1 y first, then apply op2 to z.

Step 7: Display the final result.

Step 8: Stop.

Source Code:

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

int main() {
    int x, y, z, result;
    char op1, op2;

    // Taking input from the user
    printf("Enter an arithmetic expression (x op y op z): ");
    scanf("%d %c %d %c %d", &x, &op1, &y, &op2, &z);

    // Performing the first
    operation switch(op1) {
        case '+': result = x + y;
        break; case '-': result = x
        - y; break; case '*':
        result = x * y; break;
        case '/': result = (y != 0) ? x / y : 0; break;
        default: printf("Invalid operator!\n"); return 1;
    }

    // Performing the second
    operation switch(op2) {
        case '+': result += z;
        break; case '-': result
        -= z; break; case '*':
        result *= z; break;
        case '/': result = (z != 0) ? result / z : 0;
        break; default: printf("Invalid operator!\n");
        return 1;
    }

    // Printing the result
    printf("Result: %d\n", result);
    return 0;
}
```

Sample Output:

Enter an arithmetic expression of the form (x op y op z): 1 + 2 + 3

Result : 6

Enter an arithmetic expression of the form (x op y op z): 2 * 5 / 2

Result : 5

Enter an arithmetic expression of the form (x op y op z): 201 / 2 * 6

Result : 600