

EX NO-10

GENERATE THREE ADDRESS CODES FOR A GIVEN EXPRESSION (ARITHMETIC EXPRESSION, FLOW OF CONTROL)

AIM: The aim is to generate Three-Address Code (TAC) for a given arithmetic expression and flow of control (e.g., if-else, loops). TAC is an intermediate representation used in compilers to simplify the task of code generation. It consists of simple instructions that make it easier to translate into machine-level code.

PROGRAM

```
#include <stdio.h>
#include <string.h>
void generateArithmeticTAC(const char* expr) {
    char result[10], op1[10], op[2], op2[10];
    sscanf(expr, "%s = %s %s %s", result, op1, op, op2);

    if (strcmp(op, "+") == 0 || strcmp(op, "-") == 0 ||
        strcmp(op, "*") == 0 || strcmp(op, "/") == 0) {
        printf("t1 = %s %s %s\n", op1, op, op2);
        printf("%s = t1\n", result);
    } else {
        printf("Unsupported operation: %s\n", op);
    }
}
void generateIfElseTAC(const char* condition, const char* trueStmt, const char* falseStmt) {
    printf("if %s goto L1\n", condition);
    printf("goto L2\n");
    printf("L1: %s\n", trueStmt);
    printf("goto L3\n");
    printf("L2: %s\n", falseStmt);
    printf("L3:\n");
}
void generateWhileLoopTAC(const char* condition, const char* body) {
    printf("L1: if %s goto L2\n", condition); // exit condition
    printf("%s\n", body); // loop body
    printf("goto L1\n"); // repeat loop
    printf("L2:\n");
}
int main() {
    printf("TAC for arithmetic expression:\n");
    generateArithmeticTAC("a = b + c");
    printf("\nTAC for if-else statement:\n");
    generateIfElseTAC("a < b", "x = 1", "x = 2");
    printf("\nTAC for while loop:\n");
    generateWhileLoopTAC("a >= b", "x = x + 1");
    return 0;
}
```

KAMALI K A- 220701118

OUTPUT

```
TAC for arithmetic expression:
```

```
t1 = b + c
```

```
a = t1
```

```
TAC for if-else statement:
```

```
if a < b goto L1
```

```
goto L2
```

```
L1: x = 1
```

```
goto L3
```

```
L2: x = 2
```

```
L3:
```

```
TAC for while loop:
```

```
L1: if a >= b goto L2
```

```
x = x + 1
```

```
goto L1
```

```
L2:
```

RESULT: Thus the above program is the simplified example and a complete implementation and it would need to handle more complex expressions, nested control structures, and ensure proper parsing of the input.

KAMALI K A- 220701118