

Roll No. and Name: 21BCE021 Vani Balani

Course Code and Name: 2CS701 Compiler Construction

Practical No: 10

AIM :To implement Code Optimization techniques

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

#define MAX_STATEMENTS 50
#define MAX_LEN 20

typedef struct {
    char lhs[MAX_LEN];
    char rhs[MAX_LEN];
    int isOptimized;
} Statement;

Statement statements[MAX_STATEMENTS];
int count = 0;

void addStmt(char* lhs, char* rhs) {
    strcpy(statements[count].lhs, lhs);
    strcpy(statements[count].rhs, rhs);
    statements[count].isOptimized = 0;
    count++;
}

void constFold() {
    for (int i = 0; i < count; i++) {
        int num1, num2, res;
        char op;

        if (sscanf(statements[i].rhs, "%d %c %d", &num1, &op, &num2)
== 3) {
            if (op == '+') res = num1 + num2;
            else if (op == '-') res = num1 - num2;
            else if (op == '*') res = num1 * num2;
            else if (op == '/' && num2 != 0) res = num1 / num2;
            else continue;

            sprintf(statements[i].rhs, "%d", res);
        }
    }
}
```

```

        statements[i].isOptimized = 1;
    }
}

void constProp() {
    for (int i = 0; i < count; i++) {
        if (statements[i].isOptimized) continue;

        for (int j = i + 1; j < count; j++) {
            if (strstr(statements[j].rhs, statements[i].lhs)) {
                char newRhs[MAX_LEN];
                snprintf(newRhs, sizeof(newRhs), "%s",
statements[i].rhs);
                strcpy(statements[j].rhs, newRhs);
                statements[j].isOptimized = 1;
            }
        }
    }
}

void commonExprElim() {
    for (int i = 0; i < count; i++) {
        for (int j = i + 1; j < count; j++) {
            if (strcmp(statements[i].rhs, statements[j].rhs) == 0) {
                strcpy(statements[j].rhs, statements[i].lhs);
                statements[j].isOptimized = 1;
            }
        }
    }
}

void printOptimized() {
    printf("\nOptimized Code:\n");
    for (int i = 0; i < count; i++) {
        if (!statements[i].isOptimized) {
            printf("%s = %s\n", statements[i].lhs,
statements[i].rhs);
        }
    }
}

int main() {
    addStmt("a", "5 + 3");
    addStmt("b", "a * 2");
}

```

```

    addStmt("c", "a + b");
    addStmt("d", "5 + 3");
    addStmt("e", "d + b");

    printf("Original Code:\n");
    for (int i = 0; i < count; i++) {
        printf("%s = %s\n", statements[i].lhs, statements[i].rhs);
    }

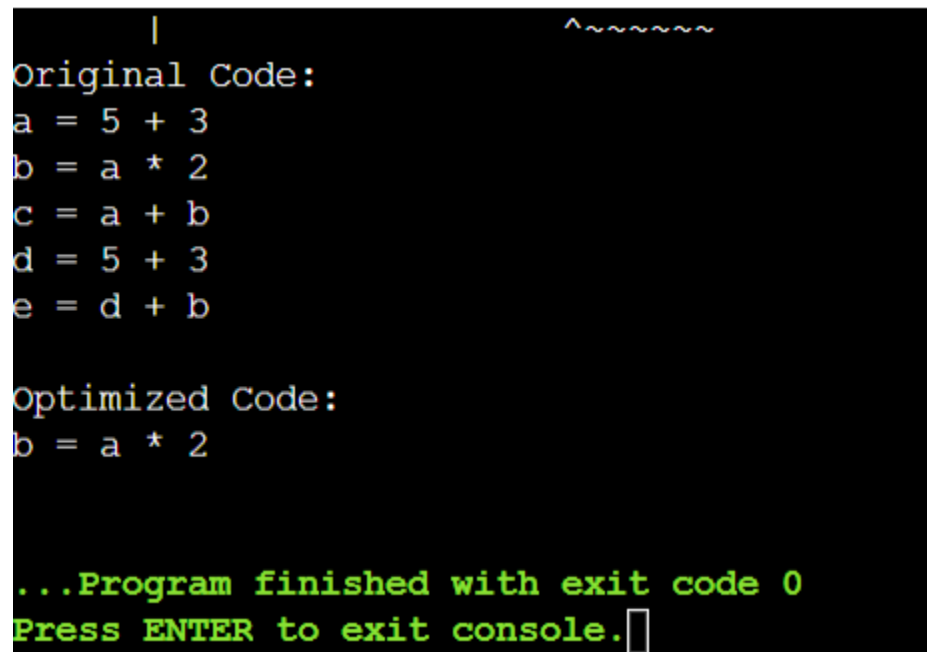
    constFold();
    constProp();
    commonExprElim();

    printOptimized();

    return 0;
}

```

OUTPUT



```

| ^~~~~~
Original Code:
a = 5 + 3
b = a * 2
c = a + b
d = 5 + 3
e = d + b

Optimized Code:
b = a * 2

...Program finished with exit code 0
Press ENTER to exit console.

```