Name: Spandan Mukherjee

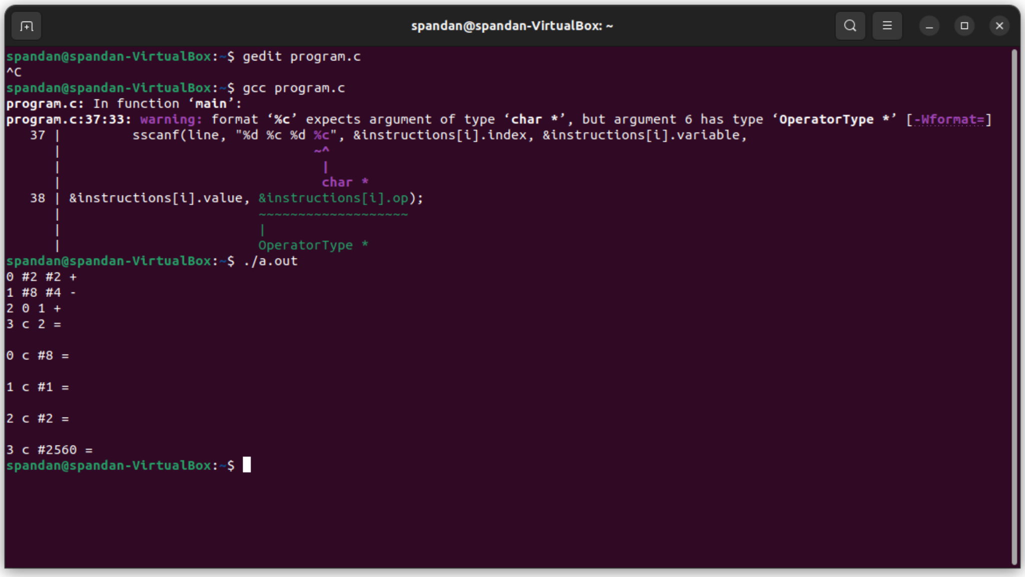
Subject: Compiler Design Lab

Experiment 7: Implementation of Code Optimization (Constant Folding)

Code:

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

#define MAX_LINE_LENGTH 50



typedef enum {
    OPERATOR_ADD,
    OPERATOR_SUB,
    OPERATOR_ASSIGN,
} OperatorType;



typedef struct {
    int index;
    char variable;
    int value;
    OperatorType op;
} Instruction;



int main() {
    Instruction instructions[4];
    char line[MAX_LINE_LENGTH];



    // Read input instructions
    for (int i = 0; i < 4; i++) {
        fgets(line, MAX_LINE_LENGTH, stdin);
        sscanf(line, "%d %c %d %c", &instructions[i].index, &instructions[i].variable,
&instructions[i].value, &instructions[i].op);
    }



    // Perform constant folding
    int a, b;
    for (int i = 0; i < 4; i++) {
        if (instructions[i].op == OPERATOR_ADD || instructions[i].op == OPERATOR_SUB) {
            // Check if both operands are constants
            if (instructions[i].variable == '#' && instructions[i + 1].variable == '#') {
                a = instructions[i].value;
                b = instructions[i + 1].value;
                // Evaluate the operation
                if (instructions[i].op == OPERATOR_ADD) {
                    instructions[i + 1].value = a + b;
                } else {
                    instructions[i + 1].value = a - b;
                }
                // Remove the current instruction
                instructions[i].op = OPERATOR_ASSIGN;
                instructions[i + 1].variable = 'c';
            }
        }
    }



    // Output optimized instructions
    for (int i = 0; i < 4; i++) {
        if (instructions[i].op == OPERATOR_ASSIGN) {
            printf("\n");
            printf("%d c #%d =", instructions[i].index, instructions[i + 1].value);
        } else {
            printf("\n");
            printf("%d c #%d =\n", instructions[i].index,instructions[i + 1].value);
        }
    }



    return 0;
}
```

Output:



```
spandan@spandan-VirtualBox:~$ gedit program.c
^C
spandan@spandan-VirtualBox:~$ gcc program.c
program.c: In function ‘main’:
program.c:37:33: warning: format ‘%c’ expects argument of type ‘char *’, but argument 6 has type ‘OperatorType *’ [-Wformat=]
   37 |         sscanf(line, "%d %c %d %c", &instructions[i].index, &instructions[i].variable,
      |                                ~^
      |                                 |
      |                                 char *
   38 | &instructions[i].value, &instructions[i].op);
      |                         ~~~~~~~~~~~~~~~~~~~
      |                         |
      |                         OperatorType *
spandan@spandan-VirtualBox:~$ ./a.out
0 #2 #2 +
1 #8 #4 -
2 0 1 +
3 c 2 =

0 c #8 =

1 c #1 =

2 c #2 =

3 c #2560 =
spandan@spandan-VirtualBox:~$ 
```