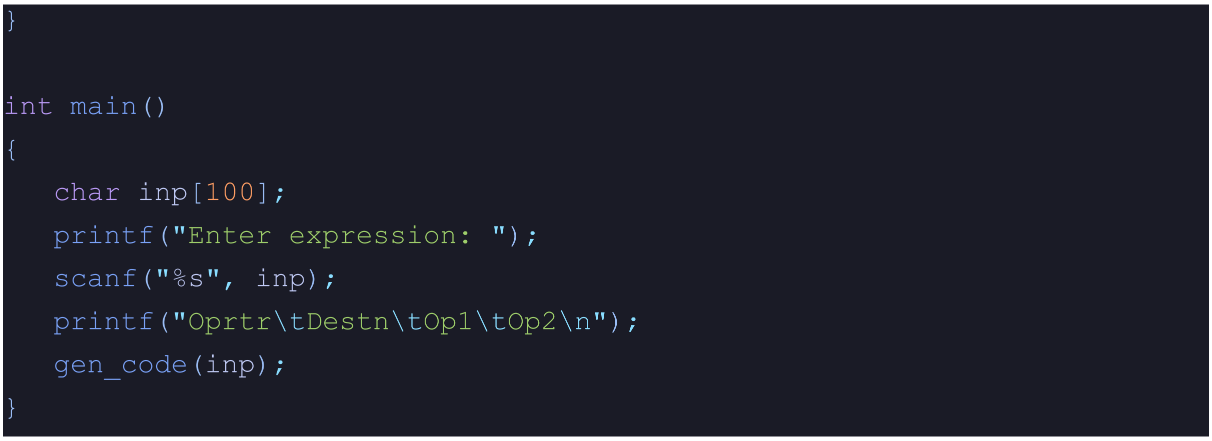
Name: Gokulnath M Prabhu
Class: CS7B
Roll No: 21

Lab Cycle 4 - Experiment 18

Implement Intermediate code generation for simple expressions

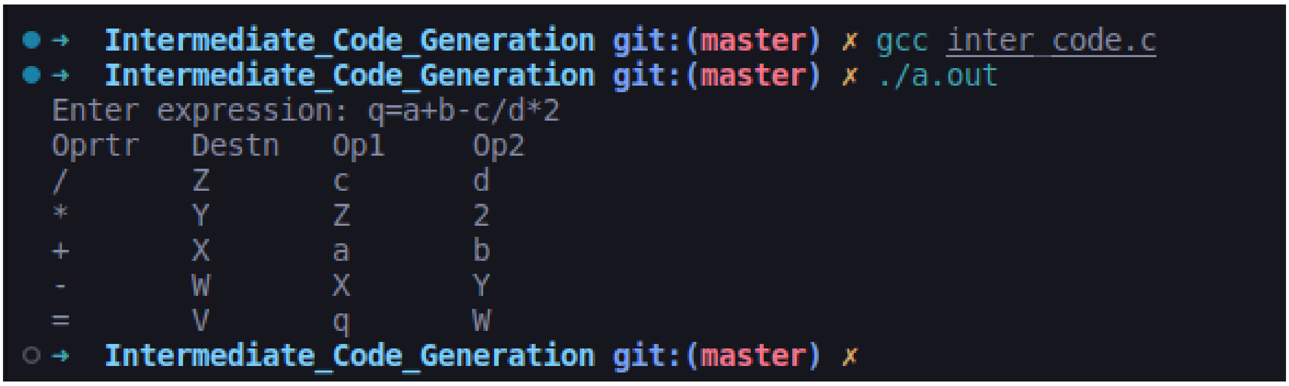
Code:

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

void gen_code_for_operator(char *inp, char operator, char * reg)
{
    int i = 0, j = 0; // j is used as an index of temp, i is used as an
index of inp
    char temp[100];
    while (inp[i] != '\0')
    {
        if (inp[i] == operator)
        {
            printf("%c\t%c\t%c\t%c\n", operator, * reg, inp[i - 1],
inp[i + 1]);
            temp[j - 1] = *reg; // Instead of copying a/b to the temp
string, copy the output register Z
                i += 2;
            (*reg)--; // Change register from Z to Y etc
            continue;
        }
        temp[j] = inp[i];
        i++;
        j++;
    }
    temp[++j] = '\0';
    strcpy(inp, temp);
}

void gen_code(char *inp)
{
    // Operator precendence - /, *, +, -, =
    char reg = 'Z'; // Decremented to get Z, Y etc
    gen_code_for_operator(inp, '/', &reg);
    gen_code_for_operator(inp, '*', &reg);
    gen_code_for_operator(inp, '+', &reg);
    gen_code_for_operator(inp, '-', &reg);
    gen_code_for_operator(inp, '=', &reg);
```



```
}

int main()
{
   char inp[100];
   printf("Enter expression: ");
   scanf("%s", inp);
   printf("Oprtr\tDestn\tOp1\tOp2\n");
   gen_code(inp);
}
```

Output:



```
● ➜  Intermediate_Code_Generation git:(master) ✗ gcc inter_code.c
● ➜  Intermediate_Code_Generation git:(master) ✗ ./a.out
  Enter expression: q=a+b-c/d*2
  Oprtr   Destn   Op1     Op2
  /       Z       c       d
  *       Y       Z       2
  +       X       a       b
  -       W       X       Y
  =       V       q       W
○ ➜  Intermediate_Code_Generation git:(master) ✗
```