COMPILER DESIGN LAB (CSL5404)

Name: Lakhan Kumawat

Roll: 1906055

Program: B.Tech CSE (5th Sem JUL-DEC 2021)

Assignment - 10

Name : Lakhan Kumawat Roll No : 1906055 Course : CSL5404

Question 1:

```
Q.1) In the syntax analyzer phase of the compiler, the parser
generates the abstract syntax tree (condensed form of the parse
tree).

This abstract syntax tree needs to be converted into machine
understandable
Format using the intermediate code generator.

Write a program in C to convert the given abstract syntaxes
into their equivalent machine codes. The following specific
machine instruction sets may
be considered:

-------------------------------------------
Following argument types may be used:
R →specifies a register in the form R0, R1, R2,etc.
L →specifies a numerical label.
V →specifies a 'variable location'pointed to by a register.
A →specifies a constant value.

-------------------------------------------

The instruction set may bedefined as follows:

LOAD A,R→loads the integer value specified by A into register
R.
STORE R,V→stores the value in register R to variable V.
OUT R→outputs the value in register R.
ADD A,R→adds the value specified by A to register R.
SUB A,R→subtracts the value specified by A from register R.
MUL A,R→multiplies the value specified by A by register R.
DIV A,R→divides register R by the value specified by A.
STOP→stops execution of the machine.
```

```
--------------------------------------------

Example:
Input:= t3 99
Output:  STORE t3, 99

--------------------------------------------

Input may be considered as:=
t1 2
[]= a 0 1
[]= a 1 2
[]= a 2 3
*t1 6 t2
+ a[2] t2 t3
-a[2] t1 t2
/ t3 t2 t2
print t2

--------------------------------------------
```

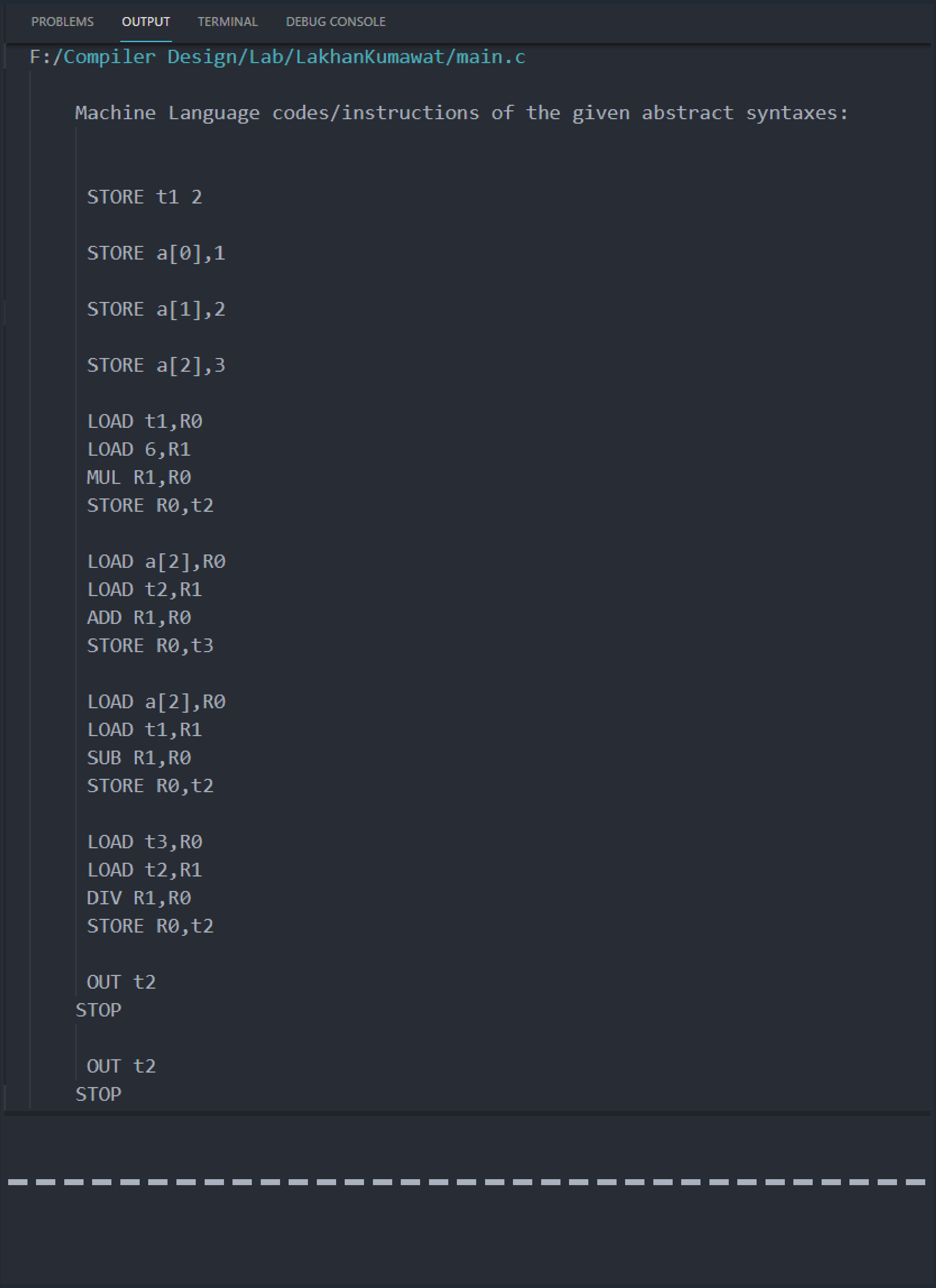
> Program Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int label[20];
int no = 0;
int check_label(int k)
{
    int i;
    for (i = 0; i < no; i++)
    {
        if (k == label[i])
            return 1;
    }
    return 0;
}
int main()
{
    FILE *fp1, *fp2;
    char fname[10], op[10], ch;
    char operand1[8], operand2[8], result[8];
    int i = 0, j = 0;
    printf("\n\n\tMachine Language codes/instructions of the 
given abstract syntaxes:\n");
    fp1 = fopen("input.txt", "r");
    if (fp1 == NULL)
    {
        printf("\n Error opening the file");
        exit(0);
    }
    while (!feof(fp1))
    {
        printf("\n");
        fscanf(fp1, "%s", op);
        i++;
        if (check_label(i))
            printf("\nlabel#%d", i);
        // for print
        if (strcmp(op, "print") == 0)
        {
            fscanf(fp1, "%s", result);
            printf("\n\t OUT %s", result);
            printf("\n\tSTOP");
        }
        //for array
        if (strcmp(op, "[]=") == 0)
        {
            fscanf(fp1, "%s %s %s", operand1, operand2, 
result);
```

```
            printf("\n\t STORE %s[%s],%s", operand1, operand2,
result);
        }
        //for operation
        switch (op[0])
        {
        case '*':
            fscanf(fp1, "%s %s ", operand1, operand2);
            printf("\n \t LOAD t1,R0");
            printf("\n \t LOAD %s,R1", operand1);
            printf("\n \t MUL R1,R0");
            printf("\n \t STORE R0,%s", operand2);
            break;
        case '+':
            fscanf(fp1, "%s %s %s", operand1, operand2,
result);
            printf("\n \t LOAD %s,R0", operand1);
            printf("\n \t LOAD %s,R1", operand2);
            printf("\n \t ADD R1,R0");
            printf("\n \t STORE R0,%s", result);
            break;
        case '-':
            fscanf(fp1, "%s %s %s", operand1, operand2,
result);
            printf("\n \t LOAD %s,R0", operand1);
            printf("\n \t LOAD %s,R1", operand2);
            printf("\n \t SUB R1,R0");
            printf("\n \t STORE R0,%s", result);
            break;
        case '/':
            fscanf(fp1, "%s %s %s", operand1, operand2,
result);
            printf("\n \t LOAD %s,R0", operand1);
            printf("\n \t LOAD %s,R1", operand2);
            printf("\n \t DIV R1,R0");
            printf("\n \t STORE R0,%s", result);
            break;
        case '=':
            fscanf(fp1, "%s %s", operand1, result);
            printf("\n\t STORE %s %s", operand1, result);
            break;
        }
    }
    fclose(fp1);
    return 0;
}
```

Output screenshot:

Name : Lakhan Kumawat   Roll No : 1906055   Course : CSL5404



```
F:/Compiler Design/Lab/LakhanKumawat/main.c

    Machine Language codes/instructions of the given abstract syntaxes:


     STORE t1 2

     STORE a[0],1

     STORE a[1],2

     STORE a[2],3

     LOAD t1,R0
     LOAD 6,R1
     MUL R1,R0
     STORE R0,t2

     LOAD a[2],R0
     LOAD t2,R1
     ADD R1,R0
     STORE R0,t3

     LOAD a[2],R0
     LOAD t1,R1
     SUB R1,R0
     STORE R0,t2

     LOAD t3,R0
     LOAD t2,R1
     DIV R1,R0
     STORE R0,t2

     OUT t2
    STOP

     OUT t2
    STOP
```

End Of Assignment