

Intermediate Code Generation

Compiler Design – Lab 9

Name: Ojas Patil
Reg No: 21BAI1106

AIM

Write a C program to generate 3-address code for the given expression.

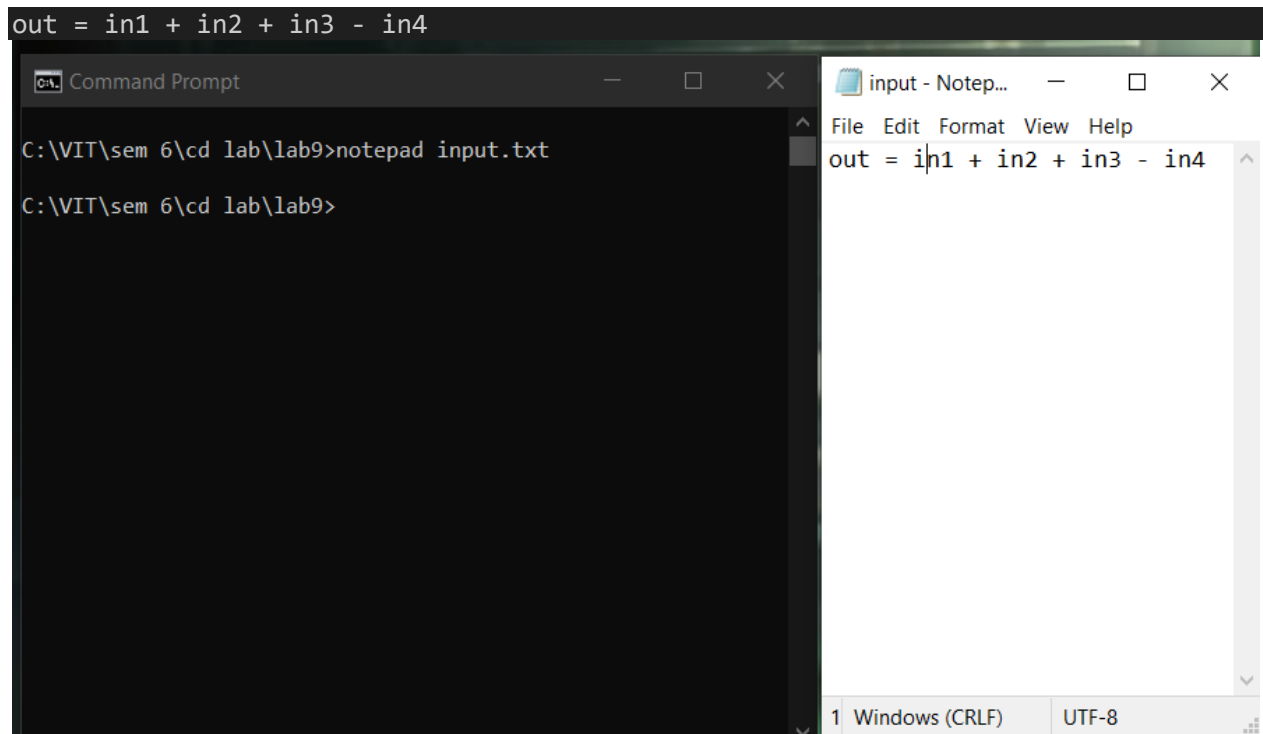
Algorithm

1. Include necessary header files: `stdio.h`, `conio.h`, `stdlib.h`, `string.h`.
2. Define structure three with data and temp arrays.
3. Declare variables and file pointers: `d1`, `d2`, `i`, `j`, `len`, `f1`, `f2`.
4. Open "sum.txt" in read mode and "out.txt" in write mode using `fopen()`.
5. Read data from "sum.txt" into the data array of structure `s`, increment `len`.
6. Process data:
 - a. If `s[3].data` is "+", write sum to output file using temporary variable.
 - b. If `s[3].data` is "-", write difference to output file using temporary variable.
7. Iterate through remaining elements of `s` and write sum or difference of previous temporary variable and current data element based on operator.
8. Write result to output file, close both input and output files using `fclose()`.

Explanation

The provided C code reads data from a file named "sum.txt" and processes mathematical expressions. It defines a structure to store data and temporary variables. The program calculates expressions using addition or subtraction operations and writes the results to an output file named "out.txt". Finally, it closes both input and output files.

Sample Input



```
out = in1 + in2 + in3 - in4
```

```
C:\VIT\sem 6\lab\lab9>notepad input.txt
C:\VIT\sem 6\lab\lab9>
```

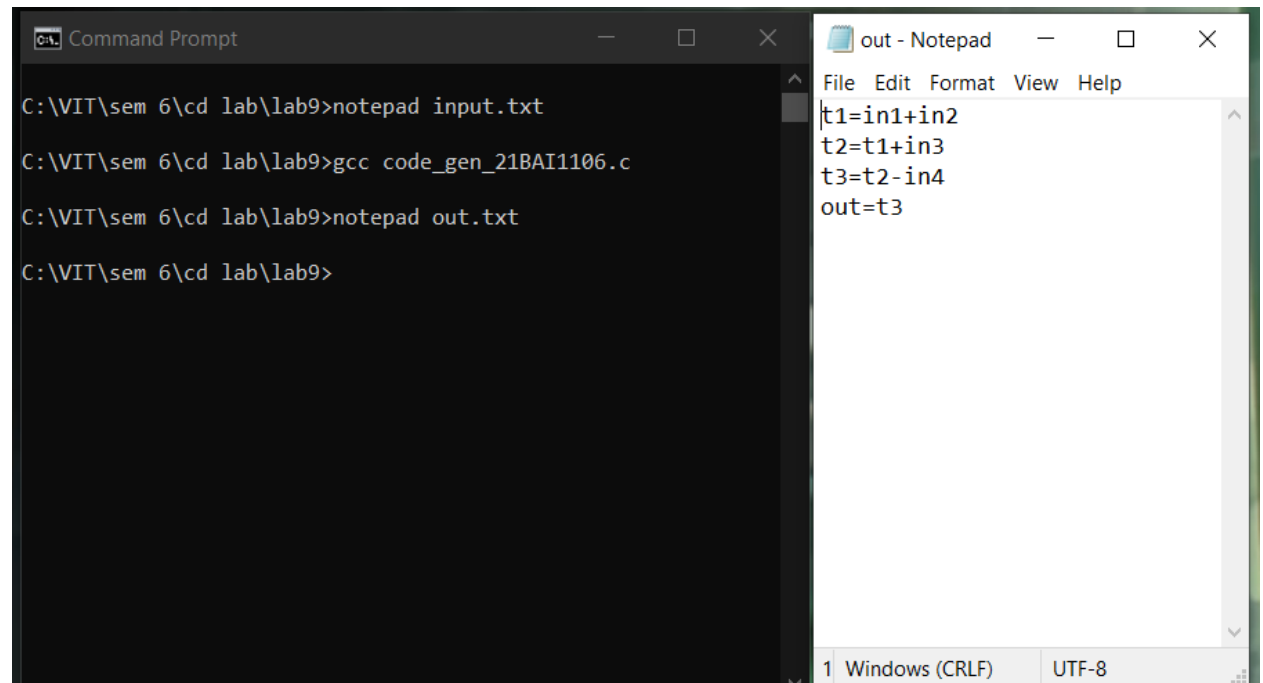
input - Notepad

File Edit Format View Help

out = in1 + in2 + in3 - in4

1 Windows (CRLF) UTF-8

Output Screenshot



```
C:\VIT\sem 6\lab\lab9>notepad input.txt
C:\VIT\sem 6\lab\lab9>gcc code_gen_21BAI1106.c
C:\VIT\sem 6\lab\lab9>notepad out.txt
C:\VIT\sem 6\lab\lab9>
```

out - Notepad

File Edit Format View Help

t1=in1+in2
t2=t1+in3
t3=t2-in4
out=t3

1 Windows (CRLF) UTF-8

Source Code

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

struct three
{
    char data[10], temp[7];
} s[30];
void main()
{
    char d1[7], d2[7] = "t";
    int i = 0, j = 1, len = 0;
    FILE *f1, *f2;
    f1 = fopen("sum.txt", "r");
    f2 = fopen("out.txt", "w");
    while (fscanf(f1, "%s", s[len].data) != EOF)
        len++;
    itoa(j, d1, 7);
    strcat(d2, d1);
    strcpy(s[j].temp, d2);
    strcpy(d1, "");
    strcpy(d2, "t");
    if (!strcmp(s[3].data, "+"))
    {
        fprintf(f2, "%s=%s+%s", s[j].temp, s[i + 2].data, s[i + 4].data);
        j++;
    }
    else if (!strcmp(s[3].data, "-"))
    {
        fprintf(f2, "%s=%s-%s", s[j].temp, s[i + 2].data, s[i + 4].data);
        j++;
    }
    for (i = 4; i < len - 2; i += 2)
    {
        itoa(j, d1, 7);
        strcat(d2, d1);
        strcpy(s[j].temp, d2);
        if (!strcmp(s[i + 1].data, "+"))
            fprintf(f2, "\n%s=%s+%s", s[j].temp, s[j - 1].temp, s[i + 2].data);
        else if (!strcmp(s[i + 1].data, "-"))
            fprintf(f2, "\n%s=%s-%s", s[j].temp, s[j - 1].temp, s[i + 2].data);
        strcpy(d1, "");
    }
}
```

```
        strcpy(d2, "t");  
        j++;  
    }  
    fprintf(f2, "\n%s=%s", s[0].data, s[j - 1].temp);  
    fclose(f1);  
    fclose(f2);  
}
```

Result

Thus, we have developed an C Code to perform intermediate code generation to generate 3-address code from the given expression.