

CS6612 – Compiler Lab

Ex no : 8

Name : Sreedhar V

Date : 23.04.2021

Reg no: 185001161

Programming Assignment-8 - Code Optimization

Code:

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

int main()
{
    FILE *fin = fopen("input.txt", "r");
    char line[100];
    while (fgets(line, 100, fin))
    {
        printf("\nInput: %s", line);
        if (line[3] == '+' && (line[2] == '0' || line[4] == '0'))
        {
            if (line[4] == '0')
            {
                if (line[0] != line[2])
                    printf("The code had been optimized to %c=%c\n", line[0], line[2]);
                else
                    printf("The code had been optimized\n");
            }
            else if (line[2] == '0')
            {
                if (line[0] != line[4])
                    printf("The code had been optimized to %c=%c\n", line[0], line[4]);
                else
                    printf("The code had been optimized\n");
            }
            else
            {
                printf("The code had been optimized\n");
            }
        }
        else if (line[3] == '*' && (line[2] == '1' || line[4] == '1'))
        {
            if (line[4] == '1')
            {
                if (line[0] != line[2])
                    printf("The code had been optimized to %c=%c\n", line[0], line[2]);
                else
                    printf("The code had been optimized\n");
            }
            else if (line[2] == '1')
            {
                if (line[0] != line[4])
                    printf("The code had been optimized to %c=%c\n", line[0], line[4]);
                else
                    printf("The code had been optimized\n");
            }
            else
            {
                printf("The code had been optimized\n");
            }
        }
    }
}
```

```

        if (line[4] == '1')
        {
            if (line[0] != line[2])
                printf("The code had been optimized to %c=%c\n", li
ne[0], line[2]);
            else
                printf("The code had been optimized\n");
        }
        else if (line[2] == '1')
        {
            if (line[0] != line[4])
                printf("The code had been optimized to %c=%c\n", li
ne[0], line[4]);
            else
                printf("The code had been optimized\n");
        }
        else
        {
            printf("The code had been optimized\n");
        }
    }
    else if (line[3] == '-' && (line[2] == '0' || line[4] == '0'))
    {
        if (line[4] == '0')
        {
            if (line[0] != line[2])
                printf("The code had been optimized to %c=%c\n", li
ne[0], line[2]);
            else
                printf("The code had been optimized\n");
        }
        else if (line[2] == '0')
        {
            printf("The code had been optimized to %s", line);
        }
        else
        {
            printf("The code had been optimized\n");
        }
    }
    else if (line[3] == '/' && (line[2] == '0' || line[4] == '1'))
    {
        if (line[4] == '1')
        {
            if (line[0] != line[2])

```

```

        printf("The code had been optimized to %c=%c\n", line[0], line[2]);
    else
        printf("The code had been optimized\n");
    }
    else if (line[4] == '1')
    {
        printf("The code had been optimized to %c=0\n", line[0]);
    };
    }
    else
        printf("The code had been optimized\n");
}
else if (line[0] == 'p' && line[1] == 'o' && line[2] == 'w')
{
    if (line[6] == '2')
    {
        printf("The code had been optimized to %c=%c*%c\n", line[4], line[4], line[4]);
    }
    else
    {
        printf("The code had been optimized\n");
    }
}
else
{
    printf("The code had been optimized\n");
}
}
fclose(fin);
return 0;

```

Output:

```
C:\Windows\System32\cmd.exe
G:\Academics\SSN\6th Sem\Compiler Design\Ex8>gcc v3.c -o a
G:\Academics\SSN\6th Sem\Compiler Design\Ex8>a
Input: x=x+0;
The code had been optimized

Input: x=0+x;
The code had been optimized

Input: y=x+0;
The code had been optimized to y=x

Input: x=0+y;
The code had been optimized to x=y

Input: x=x*1;
The code had been optimized

Input: x=1*x;
The code had been optimized

Input: y=x*1;
The code had been optimized to y=x

Input: x=1*y;
The code had been optimized to x=y

Input: pow(x,2);
The code had been optimized to x=x*x

Input: x=x-0;
The code had been optimized

Input: x=0-x;
The code had been optimized to x=0-x;

Input: y=x-0;
The code had been optimized to y=x

Input: y=0-x;
The code had been optimized to y=0-x;

Input: x=x/1;
The code had been optimized

Input: x=0/x;
The code had been optimized
```

Learning Outcome:

- I've learnt how to implement the code optimizer and analyse the code written in C and tried to optimize the code and memory by removing the useless arithmetic operations.
- I've learnt how to remove the unwanted assignment statement which affects the time and space complexity of the program.
- I've learnt to convert a C code which has redundant and unoptimized statements into an optimized program.