VARUN BHARGAVA – 241010282
DATA STRUCTURES TASK-6

Task 01: Postfix to Prefix:

(https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab6/lab6-1.c)
Example: Convert the following Postfix expression into Prefix expression:

AB + CD − ∗

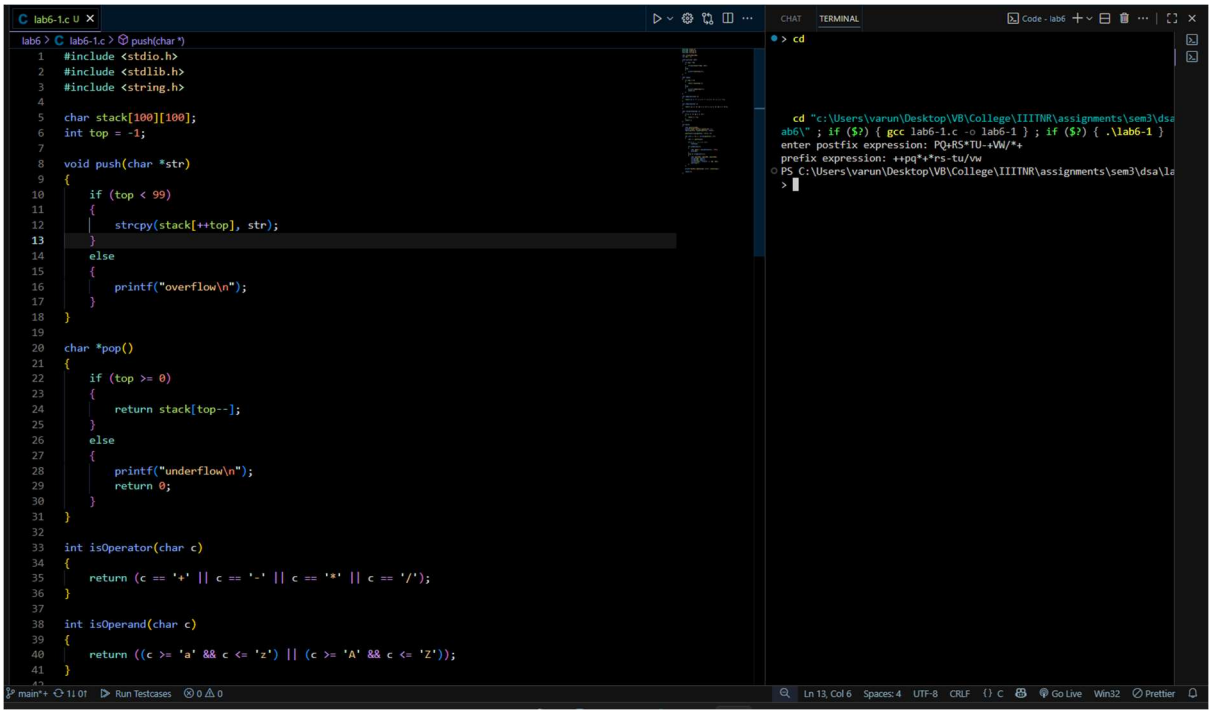
Prefix expression:

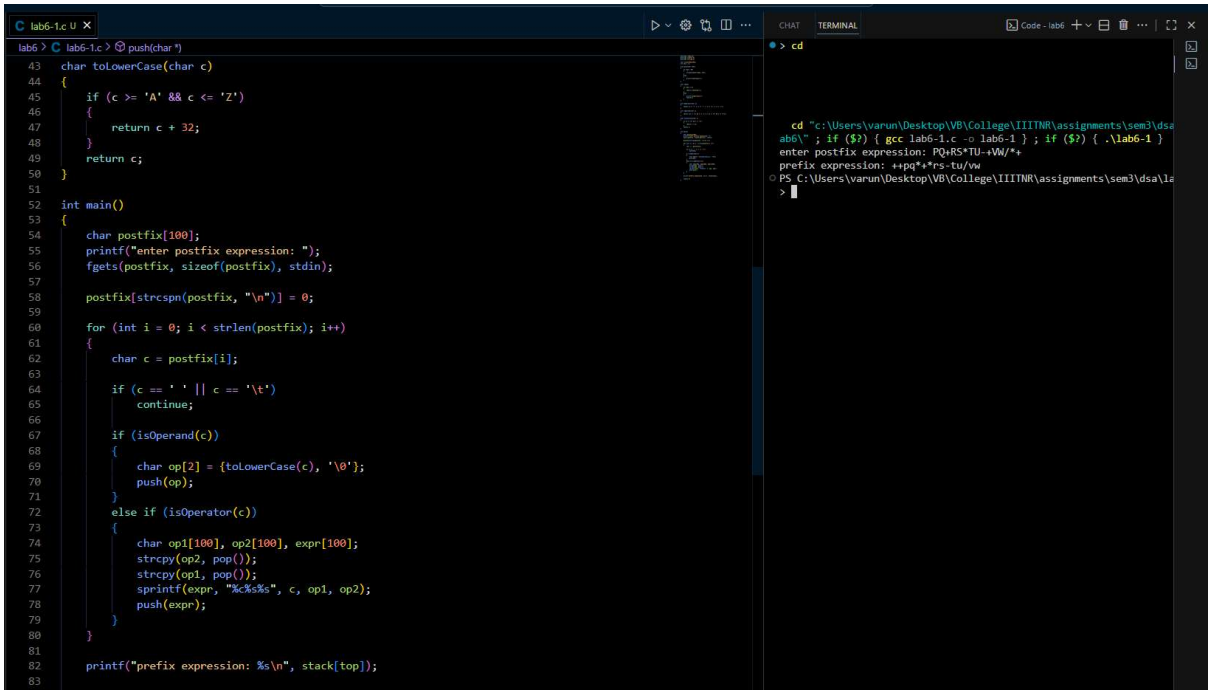
∗ + AB − CD

Question:
Convert the following Postfix expression into Prefix using stack operations:

P Q + RS ∗ T U − +V W/∗

SOLUTION:
++PQ*+*RS-TU/VW





Task 02: Stack-based Parentheses Checker:

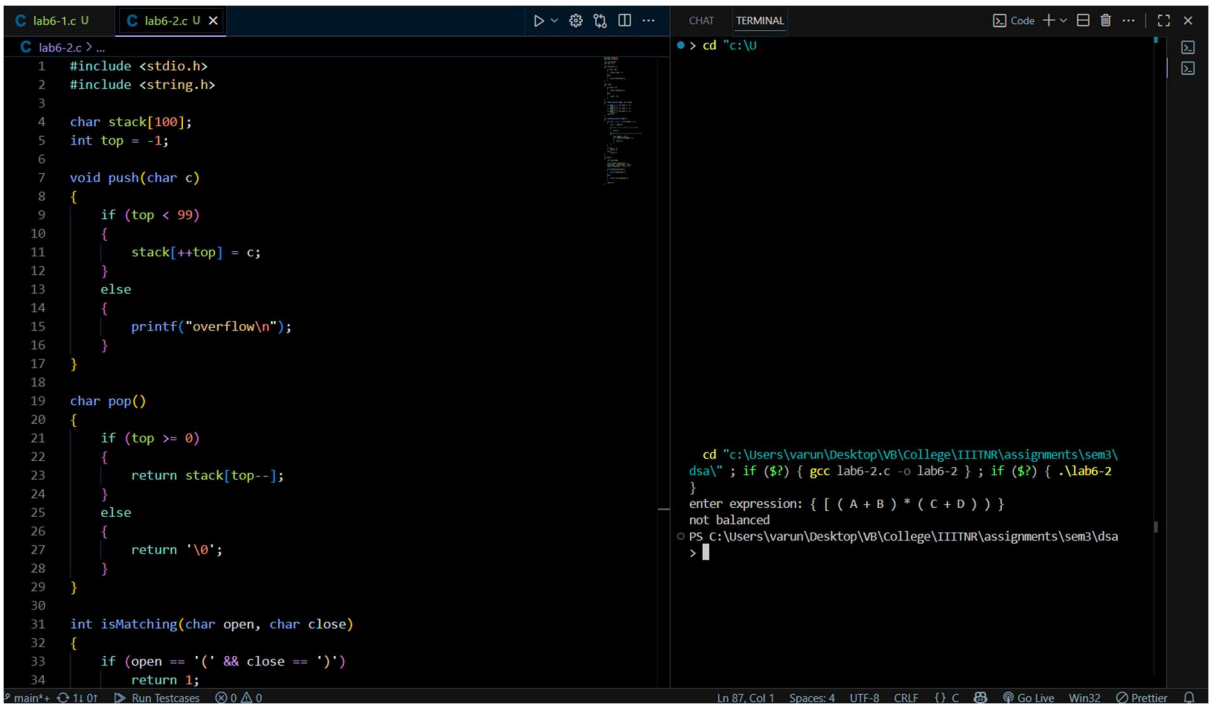
(https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab6/lab6-2.c)
Write a program using a stack to check whether a given expression has balanced parentheses or not. The expression may include the symbols: (), {}, and [].

• Display “Balanced” if all parentheses are matched properly, otherwise “Not Balanced”.

Examples:
• Input: { [(A + B) * (C + D)] }
Output: Balanced
• Input: { (A + B] }
Output: Not Balanced
Expressions to check:
1. { [(A + B) * (C + D)) }
2. (A + B] { C + D }
3. { ([A + B] * C) - D }

SOLUTION:

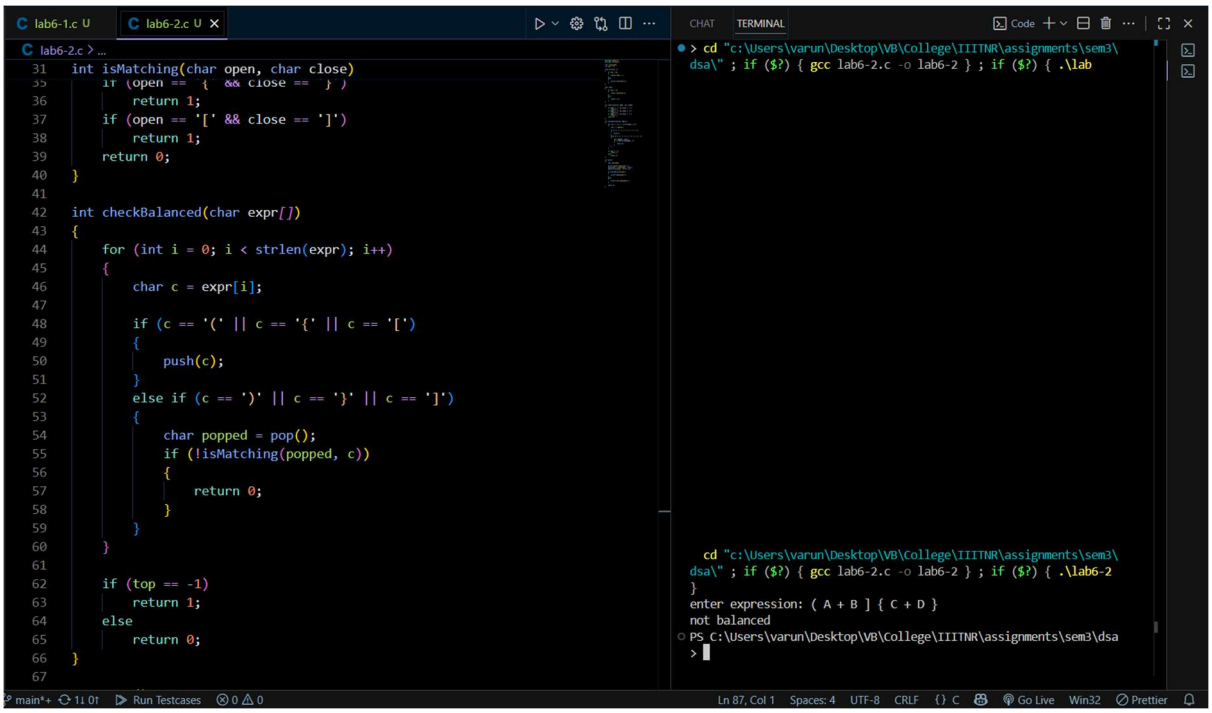
1. NOT BALANCED



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

char stack[100];
int top = -1;

void push(char c)
{
    if (top < 99)
    {
        stack[++top] = c;
    }
    else
    {
        printf("overflow\n");
    }
}

char pop()
{
    if (top >= 0)
    {
        return stack[top--];
    }
    else
    {
        return '\0';
    }
}

int isMatching(char open, char close)
{
    if (open == '(' && close == ')')
        return 1;
```

```
> cd "c:\U

  cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\
dsa\" ; if ($?) { gcc lab6-2.c -o lab6-2 } ; if ($?) { .\lab6-2
}
enter expression: { [ ( A + B ) * ( C + D ) ) }
not balanced
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa
>
```

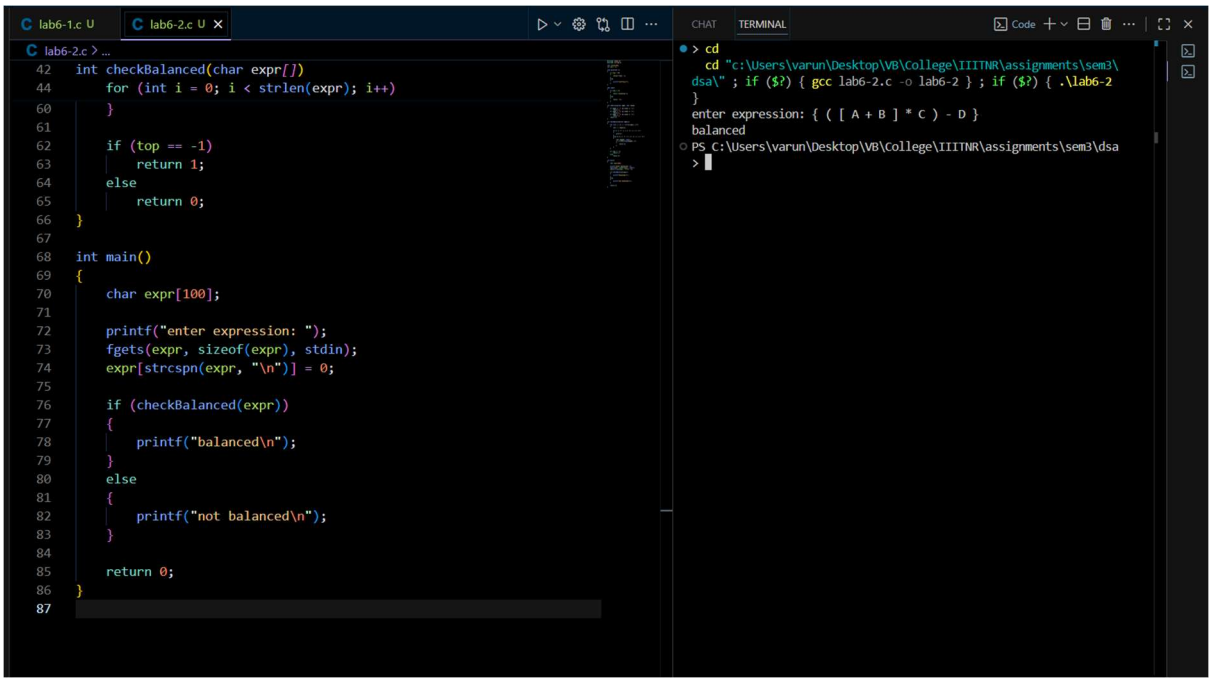
2. NOT BALANCED



```
int isMatching(char open, char close)
    if (open == '{' && close == '}')
        return 1;
    if (open == '[' && close == ']')
        return 1;
    return 0;
}

int checkBalanced(char expr[])
{
    for (int i = 0; i < strlen(expr); i++)
    {
        char c = expr[i];

        if (c == '(' || c == '{' || c == '[')
        {
            push(c);
        }
        else if (c == ')' || c == '}' || c == ']')
        {
            char popped = pop();
            if (!isMatching(popped, c))
            {
                return 0;
            }
        }
    }

    if (top == -1)
        return 1;
    else
        return 0;
}
```

```
> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\
dsa\" ; if ($?) { gcc lab6-2.c -o lab6-2 } ; if ($?) { .\lab

  cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\
dsa\" ; if ($?) { gcc lab6-2.c -o lab6-2 } ; if ($?) { .\lab6-2
}
enter expression: ( A + B ] { C + D }
not balanced
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa
>
```

3. BALANCED


lab6-1.c U
lab6-2.c U
lab6-2.c > ...
42 int checkBalanced(char expr[])
44 for (int i = 0; i < strlen(expr); i++)
60 }
61
62 if (top == -1)
63 return 1;
64 else
65 return 0;
66 }
67
68 int main()
69 {
70 char expr[100];
71
72 printf("enter expression: ");
73 fgets(expr, sizeof(expr), stdin);
74 expr[strcspn(expr, "\n")] = 0;
75
76 if (checkBalanced(expr))
77 {
78 printf("balanced\n");
79 }
80 else
81 {
82 printf("not balanced\n");
83 }
84
85 return 0;
86 }
87
CHAT
TERMINAL
Code
> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\" ; if ($?) { gcc lab6-2.c -o lab6-2 } ; if ($?) { .\lab6-2 }
enter expression: { ([A + B] * C) - D }
balanced
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa
>