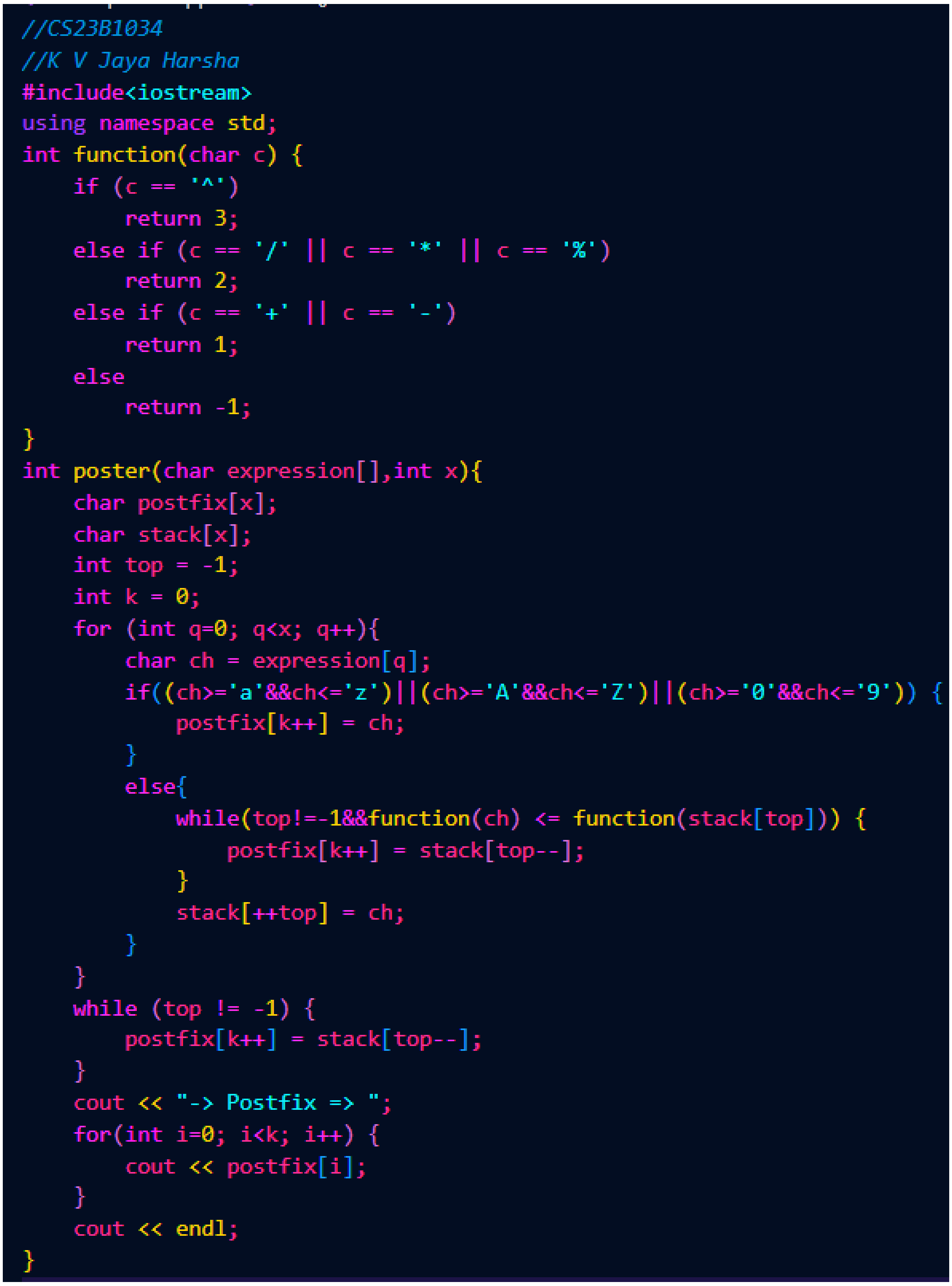
LAB-03

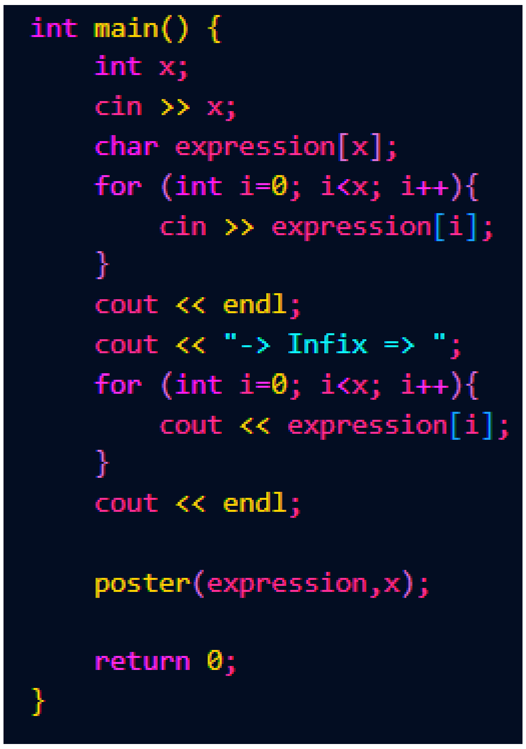
Name: K V Jaya Harsha

Roll no: CS23B1034 Date: 14-08-2024

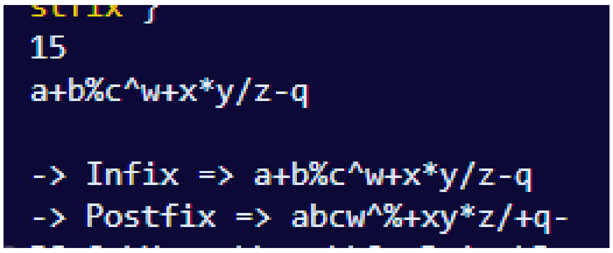
Q1. Infix to postfix. (in cpp)



```cpp
//CS23B1034
//K V Jaya Harsha
#include<iostream>
using namespace std;
int function(char c) {
    if (c == '^')
        return 3;
    else if (c == '/' || c == '*' || c == '%')
        return 2;
    else if (c == '+' || c == '-')
        return 1;
    else
        return -1;
}
int poster(char expression[],int x){
    char postfix[x];
    char stack[x];
    int top = -1;
    int k = 0;
    for (int q=0; q<x; q++){
        char ch = expression[q];
        if((ch>='a'&&ch<='z')||(ch>='A'&&ch<='Z')||(ch>='0'&&ch<='9')) {
            postfix[k++] = ch;
        }
        else{
            while(top!=-1&&function(ch) <= function(stack[top])) {
                postfix[k++] = stack[top--];
            }
            stack[++top] = ch;
        }
    }
    while (top != -1) {
        postfix[k++] = stack[top--];
    }
    cout << "-> Postfix => ";
    for(int i=0; i<k; i++) {
        cout << postfix[i];
    }
    cout << endl;
}
```

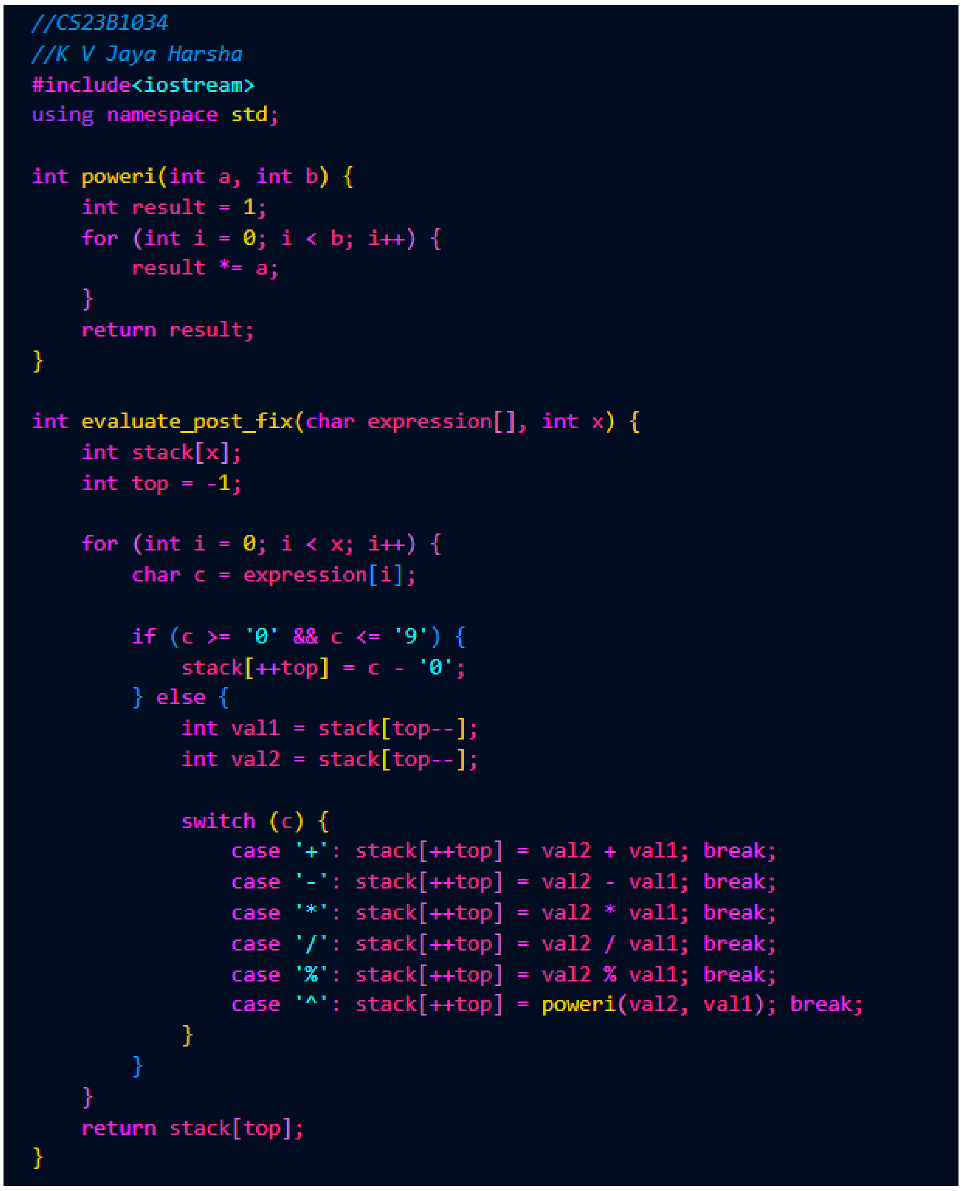


```cpp
int main() {
    int x;
    cin >> x;
    char expression[x];
    for (int i=0; i<x; i++){
        cin >> expression[i];
    }
    cout << endl;
    cout << "-> Infix => ";
    for (int i=0; i<x; i++){
        cout << expression[i];
    }
    cout << endl;

    poster(expression,x);

    return 0;
}
```

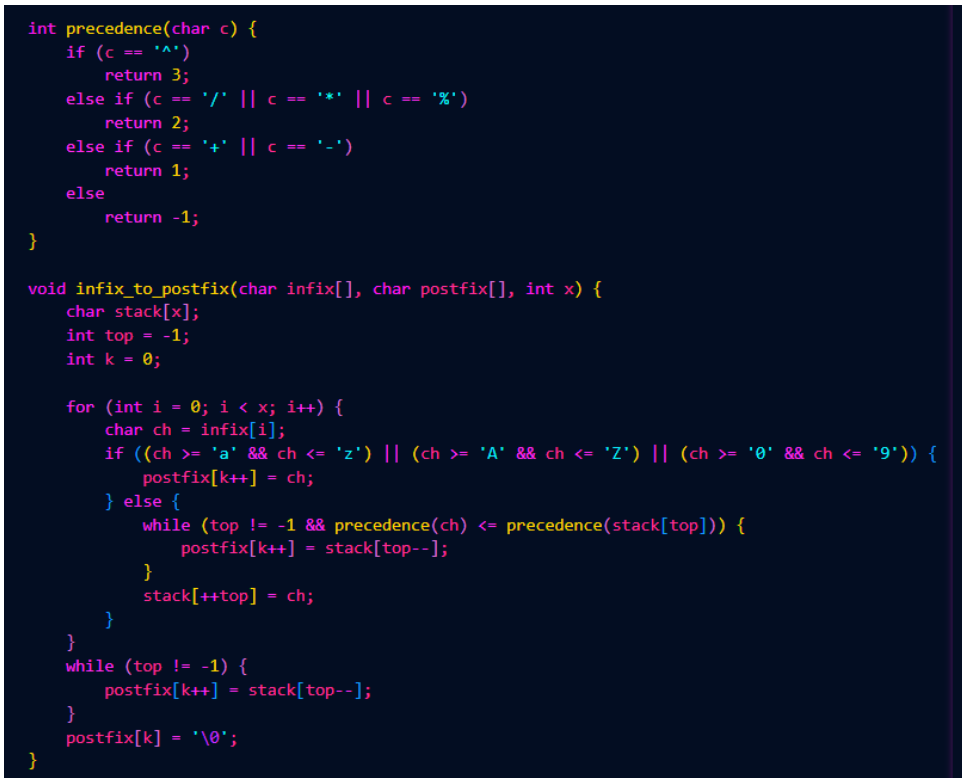


```
stfix }
15
a+b%c^w+x*y/z-q

-> Infix => a+b%c^w+x*y/z-q
-> Postfix => abcw^%+xy*z/+q-
```

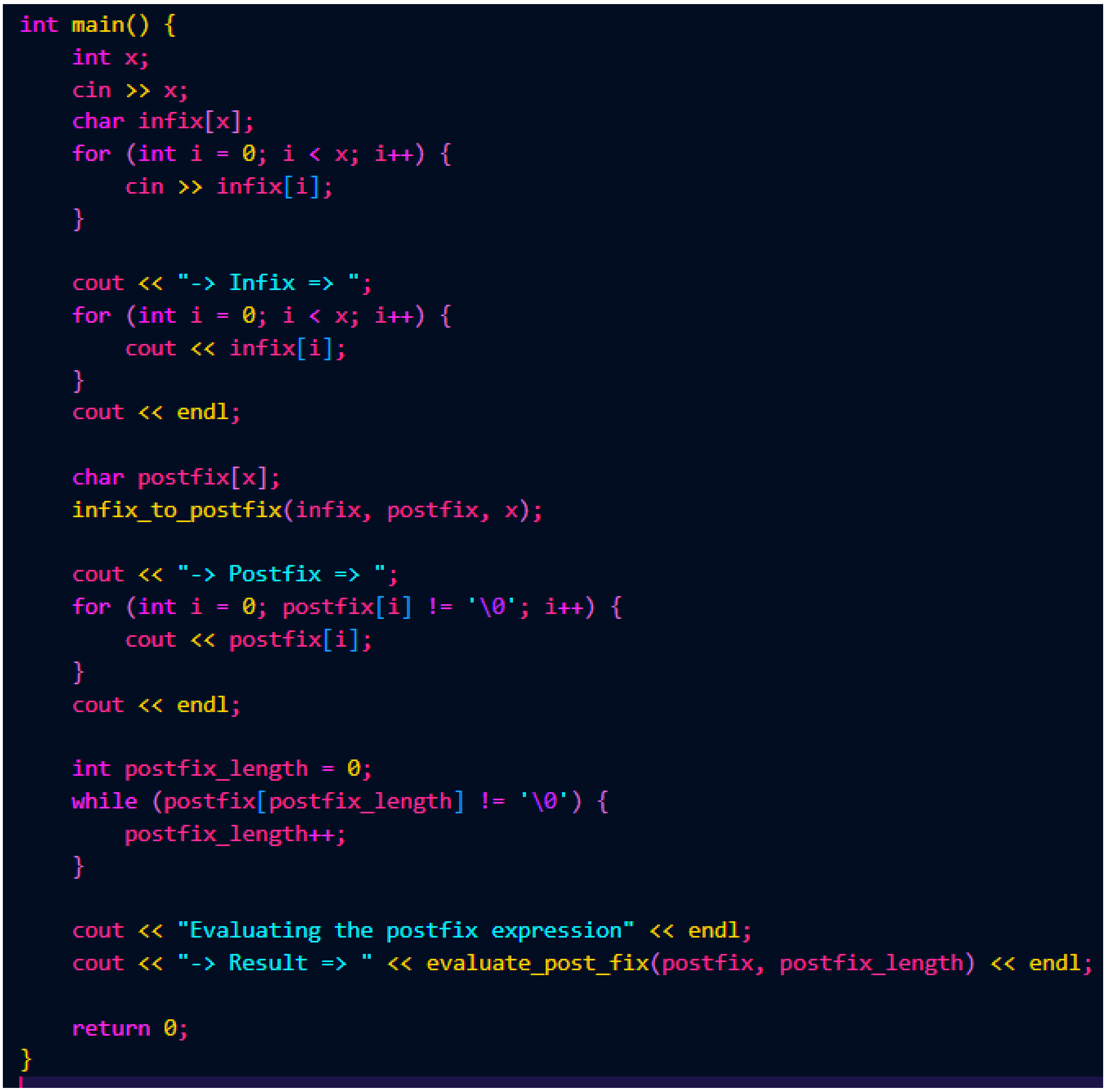
Q2. Evaluating postfix expression. (in cpp)



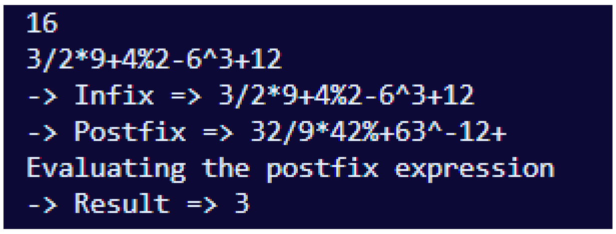
```cpp
//CS23B1034
//K V Jaya Harsha
#include<iostream>
using namespace std;

int poweri(int a, int b) {
    int result = 1;
    for (int i = 0; i < b; i++) {
        result *= a;
    }
    return result;
}

int evaluate_post_fix(char expression[], int x) {
    int stack[x];
    int top = -1;

    for (int i = 0; i < x; i++) {
        char c = expression[i];

        if (c >= '0' && c <= '9') {
            stack[++top] = c - '0';
        } else {
            int val1 = stack[top--];
            int val2 = stack[top--];

            switch (c) {
                case '+': stack[++top] = val2 + val1; break;
                case '-': stack[++top] = val2 - val1; break;
                case '*': stack[++top] = val2 * val1; break;
                case '/': stack[++top] = val2 / val1; break;
                case '%': stack[++top] = val2 % val1; break;
                case '^': stack[++top] = poweri(val2, val1); break;
            }
        }
    }
    return stack[top];
}
```



```cpp
int precedence(char c) {
    if (c == '^')
        return 3;
    else if (c == '/' || c == '*' || c == '%')
        return 2;
    else if (c == '+' || c == '-')
        return 1;
    else
        return -1;
}

void infix_to_postfix(char infix[], char postfix[], int x) {
    char stack[x];
    int top = -1;
    int k = 0;

    for (int i = 0; i < x; i++) {
        char ch = infix[i];
        if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') || (ch >= '0' && ch <= '9')) {
            postfix[k++] = ch;
        } else {
            while (top != -1 && precedence(ch) <= precedence(stack[top])) {
                postfix[k++] = stack[top--];
            }
            stack[++top] = ch;
        }
    }
    while (top != -1) {
        postfix[k++] = stack[top--];
    }
    postfix[k] = '\0';
}
```



```cpp
int main() {
    int x;
    cin >> x;
    char infix[x];
    for (int i = 0; i < x; i++) {
        cin >> infix[i];
    }

    cout << "-> Infix => ";
    for (int i = 0; i < x; i++) {
        cout << infix[i];
    }
    cout << endl;

    char postfix[x];
    infix_to_postfix(infix, postfix, x);

    cout << "-> Postfix => ";
    for (int i = 0; postfix[i] != '\0'; i++) {
        cout << postfix[i];
    }
    cout << endl;

    int postfix_length = 0;
    while (postfix[postfix_length] != '\0') {
        postfix_length++;
    }

    cout << "Evaluating the postfix expression" << endl;
    cout << "-> Result => " << evaluate_post_fix(postfix, postfix_length) << endl;

    return 0;
}
```



```
16
3/2*9+4%2-6^3+12
-> Infix => 3/2*9+4%2-6^3+12
-> Postfix => 32/9*42%+63^-12+
Evaluating the postfix expression
-> Result => 3
```