

```
} else {
```

```
        st->top++;
        st->S[st->top] = x;
    }
}
```

// Function to pop an element from the stack

```
int pop(Stack *st) {
    int x = -1;
    if (st->top == -1) {
        cout << "Stack Underflow." << endl;
    } else {
        x = st->S[st->top];
        st->top--;
    }
    return x;
}
```

// Function to peek at an element at a given position

```
int peek(Stack st, int pos) {
    int x = -1;
    if (st.top - pos + 1 < 0) {
        cout << "Invalid position." << endl;
    } else {
        x = st.S[st.top - pos + 1];
    }
    return x;
}
```

// Function to check if stack is empty

```
int isEmpty(Stack st) {
    return st.top == -1;
}
```

```
// Function to check if stack is full
int isFull(Stack st) {
    return st.top == st.size - 1;
}

int main() {
    Stack st; // Create a stack object
    create(&st); // Initialize the stack
    push(&st, 10); // Push elements
    push(&st, 20);
    push(&st, 30);
    push(&st, 40);
    push(&st, 50);

    pop(&st); // Pop one element
    display(st); // Display stack
    cout << peek(st, 3) << endl; // Peek at position 3

    delete[] st.S; // Free allocated memory
    return 0;
}
```

[illegible]

```

void push(Stack* st, char x) {
    if (st->top == st->size - 1) {
        cout << "Stack Overflow! Cannot push '" << x << "'." << endl;
    } else {
        st->top++;
        st->S[st->top] = x;
        cout << "Pushed '" << x << "' to stack." << endl;
    }
}

char pop(Stack* st) {
    char x = '\0';
    if (st->top == -1) {
        cout << "Stack Underflow! Nothing to pop." << endl;
    } else {
        x = st->S[st->top];
        st->top--;
        cout << "Popped '" << x << "' from stack." << endl;
    }
    return x;
}

void display(Stack st) {
    if (st.top == -1) {
        cout << "Stack is empty." << endl;
        return;
    }
    cout << "Stack contents (top to bottom): ";
    for (int i = st.top; i >= 0; i--) {
        cout << st.S[i];
    }
    cout << endl;
}

int main() {
    Stack st;
    char string[] = "DataStructure";
    int n = strlen(string);

    create(&st, n);

    for (int i = 0; i < n; i++) {
        push(&st, string[i]);
    }

    display(st);

    delete[] st.S;

    return 0;
}

```





Infix:  $a+b*c+d-e/f$   
Postfix:  $abc*+d+ef/-$

```

48
49 int isOperand(char x) {
50     if (x == '+' || x == '-' || x == '*' || x == '/') {
51         return 0;
52     } else {
53         return 1;
54     }
55 }
56
57 int precedence(char x) {
58     if (x == '+' || x == '-') {
59         return 1;
60     }
61     if (x == '*' || x == '/') {
62         return 2;
63     }
64     return 0;
65 }
66
67 int main() {
68     Stack st;
69     create(&st, 100);
70     char infix[] = "3+7*5+4-5/2";
71     char postfix[100] = {0};
72     int i = 0, j = 0;
73
74     while (infix[i] != '\0') {
75
76         while (infix[i] != '\0') {
77             if (isOperand(infix[i])) {
78                 postfix[j++] = infix[i++];
79             } else {
80                 while (!isEmpty(st) && precedence(st.S[st.top]) >= precedence(infix[i])) {
81                     postfix[j++] = pop(&st);
82                 }
83                 push(&st, infix[i++]);
84             }
85         }
86
87         while (!isEmpty(st)) {
88             postfix[j++] = pop(&st);
89         }
90         postfix[j] = '\0';
91
92         i = 0;
93         int x1, x2, r = 0;
94         for (i = 0; postfix[i] != '\0'; i++) {
95             if (isOperand(postfix[i])) {
96                 push(&st, postfix[i] - '0');
97             } else {
98                 x2 = pop(&st);
99                 x1 = pop(&st);
100                 switch (postfix[i]) {
101                     case '+': r = x1 + x2; break;
102                     case '-': r = x1 - x2; break;
103                     case '*': r = x1 * x2; break;
104                     case '/': r = x1 / x2; break;
105                 }
106                 push(&st, r);
107             }
108         }
109
110         cout << (int)st.S[st.top] << endl;
111
112         delete[] st.S;
113         return 0;
114     }

```

