LAPORAN UTS PRAKTIKUM STRUKTUR DATA DAN ALGORITMA

Nama / NPM : Aqila Ruqayyah / 2408107010070
Nama / NPM : Naila Hafizha / 2408107010026

A. Nama Fungsi yang Digunakan

1. initStack
   Fungsi untuk menginisialisasi stack.

2. isEmpty
   Fungsi untuk memeriksa apakah stack kosong.

3. push
   Fungsi untuk memasukkan elemen ke stack.

4. *pop
   Fungsi untuk mengeluarkan elemen dari stack.

5. *peek
   Fungsi untuk mendapatkan elemen teratas dari stack.

6. isOperator
   Fungsi untuk memeriksa apakah suatu karakter adalah operator.

7. precedence
   Fungsi untuk mendapatkan prioritas dari suatu operator.

8. infixToPostfix
   Mengonversi ekspresi infix ke postfix dengan menggunakan stack untuk mengatur urutan operator dan operand sesuai prioritas operator.

9. postfixToInfix
   Mengonversi ekspresi postfix ke infix dengan membaca setiap karakter dalam string postfix satu per satu, lalu operand langsung dimasukkan ke dalam stack. Dua operand teratas dikeluarkan dari stack, kemudian operator digabungkan dengan dua operand tersebut dalam format infix dan dimasukkan kembali ke stack. Setelah semua karakter diproses, hasil konversi infix akan diambil dari stack.

10. infixToPrefix
    Mengonversi ekspresi infix ke prefix dengan cara membalikkan ekspresi infix, mengganti tanda kurung, mengonversinya ke postfix, lalu membalikkan hasil postfix untuk mendapatkan prefix.

11. prefixToInfix
    Mengonversi ekspresi prefix ke infix dengan membaca ekspresi prefix dari kanan ke kiri. Lalu, memasukkan operand ke stack dan menggabungkan dua operand dengan operator saat menemukan operator. Setelah semua karakter diproses, hasil konversi infix diambil dari stack.

12. prefixToPostfix
    Mengonversi ekspresi prefix ke postfix dengan membaca ekspresi dari kanan ke kiri, menyusun operator dan operand, dan akhirnya menghasilkan postfix.

13. postfixToPrefix
    Mengonversi ekspresi postfix ke prefix dengan membaca setiap karakter dalam string postfix dari kiri ke kanan. Lalu, memasukkan operand ke stack dan menggabungkan dua operand dengan operator saat menemukan operator, dengan format prefix atau operator di depan. Setelah semua karakter diproses, hasil konversi prefix diambil dari stack.

14. int main()
    Fungsi main dengan menu yang menampilkan 7 pilihan konversi (infix-postfix-prefix) dan exit. User dapat melakukan input dan program akan memanggil fungsi konversi sesuai pilihan, serta adanya loop hingga user memilih exit (7).

B. Metode Struktur Data yang Digunakan

Program ini menggunakan stack berbasis linked list untuk konversi ekspresi aritmatika (infix, postfix, prefix). Sifat LIFO (Last In, First Out) pada stack memastikan operator dengan prioritas tinggi atau yang berada dalam kurung diproses lebih dahulu. Linked List memungkinkan alokasi memori dinamis, untuk menghindari batasan kapasitas statis. Algoritma seperti Shunting-yard untuk konversi infix ke postfix dan parsing rekursif untuk konversi postfix ke infix diimplementasikan dengan operasi push-pop stack, sementara konversi ke/dari prefix memanfaatkan pembalikan ekspresi.

C. Jumlah Fungsi yang Terdapat di Dalam Kode Program

Pada kode program ini terdapat 13 fungsi, dengan tanpa menghitung fungsi main. Berikut adalah 13 fungsi tersebut :

1. void initStack(Stack *s)
2. int isEmpty(Stack *s)
3. void push(Stack *s, char *c)
4. char *pop(Stack *s)
5. char *peek(Stack *s)
6. int isOperator(char c)
7. int precedence(char c)
8. void infixToPostfix(char infix[], char postfix[])
9. void postfixToInfix(char postfix[], char infix[])

10. void infixToPrefix(char infix[], char prefix[])
11. void prefixToInfix(char prefix[], char infix[])
12. void prefixToPostfix(char prefix[], char postfix[])
13. void postfixToPrefix(char postfix[], char prefix[])