LAPORAN UTS PRAKTIKUM SDA

KELOMPOK 20

Anggota :

Rahiqul Munadhil (2408107010049) dan Muhammad Irfan Qadafi (2408107010054)

A. Fungsi-fungsi yang digunakan :

1. clear()
   Membersihkan tampilan terminal. Fungsi ini mengecek apakah sistem operasi Windows atau sistem lain (misalnya Linux/Unix) dan memanggil perintah cls atau clear secara sesuai.

2. end()
   Menampilkan pesan "Press Enter to return...", menunggu input pengguna (dengan getchar()), lalu memanggil fungsi clear() untuk membersihkan layar.

3. initStack(Stack s)*
   Menginisialisasi struktur data stack dengan mengatur nilai top menjadi -1, yang berarti stack dalam keadaan kosong.

4. isEmpty(Stack s)*
   Mengecek apakah stack kosong dengan memeriksa apakah nilai top sama dengan -1. Jika ya, maka stack dianggap kosong.

5. push(Stack s, const char item)**
   Menambahkan (push) sebuah elemen ke dalam stack jika belum mencapai batas maksimal.

6. pop(Stack s)*
   Mengeluarkan (pop) elemen teratas dari stack. Jika stack kosong, fungsi mengembalikan string kosong.

7. precedence(char op)
   Mengembalikan nilai prioritas (precedence) dari operator aritmatika. Contohnya, operator '^' memiliki prioritas tertinggi (3), diikuti oleh '*' dan '/' (2), serta '+' dan '-' (1). Jika karakter bukan operator, fungsi mengembalikan -1.

8. reverseStr(char str)*
   Membalik urutan karakter dalam sebuah string. Fungsi ini digunakan, misalnya, pada konversi infix ke prefix.

9. removeSpaces(char str)*
   Menghapus semua spasi dalam sebuah string sehingga ekspresi tidak mengandung spasi yang tidak diperlukan.

10. infixToPostfix()
    Mengonversi ekspresi aritmatika dari format infix ke postfix. Fungsi ini membaca ekspresi infix, menghapus spasi, dan menggunakan stack untuk mengatur operator berdasarkan prioritasnya.

11. postfixToInfix()
    Mengonversi ekspresi aritmatika dari format postfix ke infix. Dengan menggunakan stack, fungsi ini mengambil operand dan operator untuk membentuk ekspresi infix yang valid.

12. infixToPrefix()
    Mengonversi ekspresi aritmatika dari format infix ke prefix. Langkah awal yang dilakukan adalah membalik string infix, kemudian mengubahnya ke bentuk postfix, dan akhirnya membalik hasil postfix tersebut menjadi prefix.

13. prefixToInfix()
    Mengonversi ekspresi aritmatika dari format prefix ke infix. Fungsi ini memproses ekspresi dari kanan ke kiri, menggunakan stack untuk menyusun ulang operand dan operator ke dalam format infix.

14. prefixToPostfix()
    Mengonversi ekspresi aritmatika dari format prefix ke postfix. Dengan membaca ekspresi dari kanan ke kiri, fungsi ini menggabungkan operand dan operator sesuai dengan aturan konversi.

15. postfixToPrefix()
    Mengonversi ekspresi aritmatika dari format postfix ke prefix. Proses konversi dilakukan dengan memanfaatkan stack untuk menyusun kembali urutan operator dan operand agar membentuk ekspresi prefix yang benar.

16. main()
    Fungsi utama yang menampilkan menu pilihan konversi ekspresi aritmatika (seperti infix ke postfix, postfix ke infix, dan seterusnya). Pengguna dapat memilih opsi yang diinginkan, dan berdasarkan pilihan tersebut, fungsi yang sesuai akan dipanggil. Fungsi ini juga mengatur loop utama program sehingga pengguna dapat melakukan beberapa konversi hingga memilih untuk keluar.

B. Metode struktur data yang digunakan

Menggunakan metode stack. Stack di sini diimplementasikan menggunakan array dua dimensi untuk menyimpan ekspresi (atau elemen string) dan sebuah variabel top yang menandai indeks elemen teratas dalam stack. Struktur data stack digunakan secara luas untuk operasi push dan pop yang sangat membantu dalam proses konversi antara notasi infix, postfix, dan prefix.

C. Jumlah fungsi yang terdapat di dalam kode program tidak termasuk fungsi main

Jumlah fungsi yang terdapat di dalam kode program (tidak termasuk fungsi main) adalah 15 fungsi, yaitu: clear(), end(), initStack(Stack* s), isEmpty(Stack* s), push(Stack* s, const char* item), pop(Stack* s), precedence(char op), reverseStr(char* str), removeSpaces(char* str), infixToPostfix(), postfixToInfix(), infixToPrefix(), prefixToInfix(, prefixToPostfix(), postfixToPrefix()