

Latihan UAS ProgLan 2024

Isian Singkat

1. Apa output dari program berikut? (Jawaban: 4)

```
#include <stdio.h>

int main() {
    int i, j, cnt = 0;
    for (i = 0; i < 10; i++) {
        for (j = 0; j < 7; j++) {
            if (i + j % 2 == 0) {
                cnt ++;
            }
        }
    }
    printf("%d", cnt);
};
```

2. Apa output dari program berikut? (Jawaban: 125)

```
#include <stdio.h>

int main() {
    int a = 5;
    int *b = &a;
    int *c = b;

    printf("%d", a**b**c);
};
```

3. Apa output dari program berikut? (Jawaban: 3.00)

```
#include <stdio.h>

int main() {
    int angka1 = 10;
    int angka2 = 3;

    float hasil = angka1 / angka2;

    printf("%.2f", hasil);
};
```

4. Apa output dari program berikut? (Jawaban: 13)

```
#include <stdio.h>

int main() {
    int i;
    for (i = 0; i < 10; i += 5) {
        i--;
    }
    printf("%d", ++i);
};
```

5. Apa output dari program berikut? (Jawaban: 70)

```
#include <stdio.h>

int functionA(int a, int b);
int functionB(int a, int b);
int functionC(int a, int b);

int main() {
    printf("%d", functionA(3, 0));
};

int functionA(int a, int b) {
    if (a == 0) return b;
    return functionB(a, b + 5);
}

int functionB(int a, int b) {
    return functionC(a, 2 * b);
}

int functionC(int a, int b) {
    return functionA(a - 1, b);
}
```

Pemrograman

*Untuk soal pemrograman memang dirancang susah karena biasanya bersifat open book (**Belum Tentu**). Jangan lupa juga untuk belajar Multithreading & Paralel Programming karena di Latihan Soal ini tidak ada.*

1. Buatlah sebuah **Circular Doubly Linked List** dalam bahasa C. Program akan meminta input sebanyak N bilangan bulat. Angka tersebut akan disimpan pada Circular Doubly Linked List. Setiap node akan berisikan data integer, pointer menuju node berikutnya, dan pointer menuju node sebelumnya.
2. Lakukan **Pengurutan** pada Circular Doubly Linked List tersebut, dan hasilnya disimpan pada **Linked List biasa** yang baru. Dengan kata lain, Circular Doubly Linked List tidak perlu diurutkan, namun buatlah Linked List baru yang berisikan data terurut dari Circular Doubly Linked List tersebut.

Hint: Anda dapat melakukan iterasi setiap elemen pada Circular Doubly Linked List dari Head sampai Tail, dan di setiap elemen lakukan **Insertion Sort** untuk data tersebut ke Linked List yang baru.

3. Setelah Linked List terurut terbentuk, program akan meminta input sebuah angka bilangan bulat pada User, dan kemudian program tersebut akan melakukan **Binary Search** untuk data tersebut pada Linked List yang terurut. Hasil pencarian akan mengembalikan indeks dari data yang dicari (indeks dimulai dari 0), dan akan mengembalikan angka -1 apabila data yang dicari tidak terdapat pada Linked List.
4. Setelah pencarian selesai dilakukan, buatlah sebuah Binary Search Tree yang optimal dari Linked List terurut tersebut.

Referensi: <https://www.geeksforgeeks.org/sorted-linked-list-to-balanced-bst/>

5. Di akhir program, cetaklah isi dari Circular Doubly Linked List, Sorted Linked List, dan Binary Search Tree tersebut dalam format berikut.

a. Circular Doubly Linked List

```
Indeks-0 \t addr: %p \t next-addr: %p \t prev-addr: %p \t data: %d \n
Indeks-1 \t addr: %p \t next-addr: %p \t prev-addr: %p \t data: %d \n
...
Indeks-N \t addr: %p \t next-addr: %p \t prev-addr: %p \t data: %d \n
```

b. Sorted Linked List

```
Indeks-0 \t addr: %p \t next-addr: %p \t data: %d \n
Indeks-1 \t addr: %p \t next-addr: %p \t data: %d \n
...
Indeks-N \t addr: %p \t next-addr: %p \t data: %d \n
```

c. Binary Search Tree

Untuk BST, cetaklah dengan menggunakan traversal BFS (Breadth-First Search)