

Nama: Insan Nafi' Alkalas
NIM: 1203230022
Kelas: IF 03-03

Tugas Self referential structure

source code

```
#include <stdio.h>
```

```
typedef struct Node {  
    struct Node* link;  
    char* alphabet;  
} Node;
```

```
int main() {  
    Node l1, l2, l3, l4, l5, l6, l7, l8, l9;
```

```
    l1.link = NULL;  
    l1.alphabet = "F";
```

```
    l2.link = NULL;  
    l2.alphabet = "M";
```

```
    l3.link = NULL;  
    l3.alphabet = "A";
```

```
    l4.link = NULL;  
    l4.alphabet = "I";
```

```
    l5.link = NULL;  
    l5.alphabet = "K";
```

```
    l6.link = NULL;  
    l6.alphabet = "T";
```

```
    l7.link = NULL;  
    l7.alphabet = "N";
```

```
    l8.link = NULL;  
    l8.alphabet = "O";
```

```
    l9.link = NULL;  
    l9.alphabet = "R";
```

```
// Hubungkan node sesuai dengan arah panah pada gambar dengan urutan N-F-O-M-K-A-T-R-I
```

```
    l7.link = &l1;  
    l1.link = &l8;  
    l8.link = &l2;  
    l2.link = &l5;  
    l5.link = &l3;  
    l3.link = &l6;  
    l6.link = &l9;
```

```
l9.link = &l4;  
l4.link = &l7;
```

```
// Print kata INFORMATIKA menggunakan linked list dimana l3 menjadi starting point dari linknya  
printf("%s", l3.link->link->link->alphabet);  
printf("%s", l3.link->link->link->link->alphabet);  
printf("%s", l3.link->link->link->link->link->alphabet);  
printf("%s", l3.link->link->link->link->link->link->alphabet);  
printf("%s", l3.link->link->link->link->link->link->link->alphabet);  
printf("%s", l3.alphabet);  
printf("%s", l3.link->alphabet);  
printf("%s", l3.link->link->link->alphabet);  
printf("%s", l3.link->link->link->link->link->link->link->link->alphabet);  
printf("%s", l3.alphabet);  
  
return 0;  
}
```

penjelasan

baris 3 – 6: baris ini digunakan untuk membuat struktur dan juga membuat alias struktur tersebut. Di dalam struktur tersebut ada 2 kode yaitu struktur dan tipe data Char.

Baris 8 – 63: baris ini adalah wadah untuk pemrograman.

Baris 9: pada baris ini digunakan untuk memberi suatu nama variabel dari data struktur.

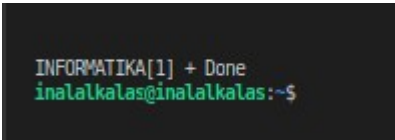
Baris 11,14,17,20,23,26,29,32,35: baris ini digunakan untuk data yang sudah mempunyai nama variabel pada baris 9 dan code tersebut menembak alamat link dari struktur mempunyai nilai NULL.

Baris 12,15,18,21,24,27,30,33,36: baris ini digunakan untuk data yang sudah mempunyai nama variabel pada baris 9 dan code tersebut menembak alamat alphabet dari struktur dan mempunyai nilai yang tertara pada code.

Baris 39 – 47: baris ini digunakan untuk menghubungkan alamat link yang mempunyai dan menyimpan alamat tersebut di alamat yang baru. Cth `l7.link = &l1;` dari code itu diketahui bahwa nama variabel l7 mempunyai nilai kemudian kita mengganti alamat tersebut menjadi l1 atau yang muali pertama.

Baris 50 – 60: baris ini digunakan untuk mengeluarkan output char dengan cara mengakses dari variabel l3 kemudian lanjut ke link selanjutnya hingga menemukan alphabet. Lebih tepatnya memberi akses untuk data yang diinginkan pada output tersebut sebanyak berapa kali link itu dikeluarkan.

Hasil program



```
INFORMATIKA[1] + Done  
inalalkalas@inalalkalas:~$
```

Tugas oth ke-2

pada tugas ke 2. soa tersebut berada di hackkerrank dan hanya menambahkan kode ini saja. Untuk melihat full kode bisa ke tugas_oth.c

```
int twoStacks(int maxSum, int a_count, int* a, int b_count, int* b) {
    int sum = 0, count = 0, i = 0, j = 0;

    // Mengambil sebanyak mungkin elemen dari stack a sampai totalnya melebihi maxSum atau
    // stack a habis.
    while (i < a_count && sum + a[i] <= maxSum) {
        sum += a[i];
        i++;
    }

    count = i; // Inisialisasi count dengan jumlah elemen yang diambil dari stack a

    // Mengambil elemen dari stack b satu per satu dan mengurangi elemen dari stack a jika
    // perlu
    while (j < b_count && i >= 0) {
        sum += b[j];
        j++;

        // Mengurangi elemen dari stack a jika total melebihi maxSum
        while (sum > maxSum && i > 0) {
            i--;
            sum -= a[i];
        }

        // Memperbarui count jika jumlah elemen yang diambil saat ini lebih besar dari sebelumnya
        if (sum <= maxSum && i + j > count) {
            count = i + j;
        }
    }

    return count;
}
```

penjelasan

baris 29: menginputkan tipe data integer = 0

baris 32 – 35: baris ini digunakan untuk mengambil sebanyak mungkin pada stack a lebih tepatnya array.

Baris 33: baris ini digunakan untuk menambahkan nilai elemen dari stack/array a ke variabel sum.

Baris 34: baris ini melakukan increment pada nilai i.

Baris 37: baris ini digunakan untuk menginisialisasi nilai count yang didapat pada stack/array a

baris 40 – 54: baris ini fungsinya persis dengan baris sebelumnya yang dimana hanya nama variabel tersebut saja yang berbeda. Code itu berada pada baris 41 – 42.

Baris 45 – 48: baris ini digunakan untuk mengurangi elemen pada stack/array a jika total data tersebut melebihi jumlah total yang sudah ditambahkan pada stack/array b. dan jika ada pengurangan pada stack/array a maka stack/array tersebut diambil hingga tidak melewati batas.

Baris 51 – 53: baris ini digunakan untuk memperbarui data count jika ada jumlah data yang diambil saat ini lebih besar dari sebelumnya

untuk output tersebut berada di file txt jadi saya mescreenshot secara terpisah

hasil

```
1
5 4 11
4 5 2 1 1
3 1 1 2
[1] + Done
inalalkalas@inalalkalas:~$
```

5

Visualisasi Data

4	
5	3
2	1
1	1
1	2

step 1

4	5	2
---	---	---

nilai tersebut 11 dari stack/array a dan count 3

step 2

4	5	2
---	---	---

stack B

3

nilai tersebut melebihi hi nilai sum yaitu 14

step 3

4

3

data tinggal 2 karna kelebihan nilai dan sekarang nilai sum tersebut 7

step 4

4

3	1	1
---	---	---

pada step 4 kita langsung saja meambhkan data b 2 kali yang bernilai 1. dan data sekrang terdapat 9 dan count tersebut bertambah menjadi 4

step 5

4

3	1	1	2
---	---	---	---

sekarang total data tersebut sudah 11 dan countny juga bertambah menjadi 5