

Nama : Richard Edgar G

NIM : 1203230084

Kelas : IF 03 – 03

Tugas Praktikum Struct dan Stack OTH

1. Tugas Link
 - Source Code

```
1  #include <stdio.h>
2
3  struct Node {
4      struct Node* link;
5      char alphabet;
6  };
7
8  int main() {
9      //inisialisasi batu - batu
10     struct Node l1, l2, l3, l4, l5, l6, l7, l8, l9;
11
12     l1.link = NULL;
13     l1.alphabet = 'F';
14
15     l2.link = NULL;
16     l2.alphabet = 'M';
17
18     l3.link = NULL;
19     l3.alphabet = 'A';
20
21     l4.link = NULL;
22     l4.alphabet = 'I';
23
24     l5.link = NULL;
25     l5.alphabet = 'K';
26
27     l6.link = NULL;
28     l6.alphabet = 'T';
29
30     l7.link = NULL;
31     l7.alphabet = 'N';
32
33     l8.link = NULL;
34     l8.alphabet = 'O';
35
36     l9.link = NULL;
37     l9.alphabet = 'R';
38
39     // Menghubungkan batu-batu
40     l7.link = &l1; //N ke F
41     l1.link = &l8; //F ke O
42     l8.link = &l2; //O ke M
43     l2.link = &l5; //M ke K
44     l5.link = &l3; //K ke A
45     l3.link = &l6; //A ke T
46     l6.link = &l9; //T ke R
47     l9.link = &l4; //R ke I
48     l4.link = &l7; //I ke N
49
50     printf("%c", l3.link->link->link->alphabet); //3x Link untuk output huruf I
51     printf("%c", l3.link->link->link->link->alphabet); //4x Link untuk output huruf N
52     printf("%c", l3.link->link->link->link->link->alphabet); //5x Link untuk output huruf F
53     printf("%c", l3.link->link->link->link->link->link->alphabet); //6x Link untuk output huruf O
54     printf("%c", l3.link->link->alphabet); //2x Link untuk output huruf R
55     printf("%c", l3.link->link->link->link->link->link->link->alphabet); //7x Link untuk output huruf M
56     printf("%c", l3.alphabet); //output huruf A karena starting point adalah huruf A
57     printf("%c", l3.link->alphabet); //1x Link untuk output huruf T
58     printf("%c", l3.link->link->link->alphabet); //3x Link untuk output huruf I
59     printf("%c", l3.link->link->link->link->link->link->link->alphabet); //8x Link untuk output huruf K
60     printf("%c", l3.alphabet); //output huruf A karena starting point adalah huruf A
61     return 0;
62 }
63
```

- Output

```
INFORMATIKA
Process returned 0 (0x0)   execution time : 0.443 s
Press any key to continue.
|
```

2. Tugas Hackerrank

- Source Code

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX_SIZE 100000
5
6  typedef struct {
7      int top;
8      int data[MAX_SIZE];
9  } Stack;
10
11 // Program untuk push atau memasukkan data kedalam stack
12 void push(Stack *stack, int value) {
13     stack->top++;
14     stack->data[stack->top] = value; // Memasukkan nilai ke dalam array pada posisi top
15 }
16
17 // Program untuk pop atau mengeluarkan data dari stack paling atas
18 int pop(Stack *stack) {
19     return stack->data[stack->top--]; // Mengembalikan nilai pada posisi top dan mengurangi indeks top
20 }
21
22 // Fungsi untuk mengambil nilai pada posisi top dari stack
23 int top(Stack *stack) {
24     return stack->data[stack->top]; // Mengembalikan nilai pada posisi top tanpa mengubah indeks top
25 }
26
27 // Fungsi duaStack untuk mengecek banyaknya operasi yang dilakukan
28 int duaStack(int total, int a[], int b[], int la, int lb) {
29     Stack stackA, stackB; // Mendefinisikan dua stack untuk menyimpan data dari array a dan b
30     stackA.top = stackB.top = -1; // Menginisialisasi indeks top stackA dan stackB menjadi stack kosong
31
32     // Memasukkan data dari array a ke dalam stackA
33     for (int i = la - 1; i >= 0; i--) {
34         push(&stackA, a[i]);
35     }
36
37     // Memasukkan data dari array b ke dalam stackB
38     for (int i = lb - 1; i >= 0; i--) {
39         push(&stackB, b[i]);
40     }
41
42     int sum = 0, counter = 0; // Inisialisasi variabel untuk menyimpan jumlah total dan jumlah operasi
43
44     // Menentukan panjang perulangan sesuai dengan array a atau b yang lebih pendek
45     int panjang = la >= lb ? lb : la;
46
47     // Melakukan perulangan untuk mengecek jumlah operasi yang dapat dilakukan
48     for (int i = 0; i < panjang; i++) {
49         // Jika penjumlahan total dengan nilai pada stackA masih kurang dari atau sama dengan total
50         if (sum + top(&stackA) <= total) {
51             sum += pop(&stackA); // Menambahkan nilai pada stackA ke dalam total
52             counter++; // Menambahkan counter
53         }
54         // Jika penjumlahan total dengan nilai pada stackB masih kurang dari atau sama dengan total
55         if (sum + top(&stackB) <= total) {
56             sum += pop(&stackB); // Menambahkan nilai pada stackB ke dalam total
57             counter++; // Menambahkan counter
58         }
59     }
60
61     return counter; // Mengembalikan jumlah operasi yang dapat dilakukan
62 }
```

```

1  int main() {
2      int g, n, m, total;
3      scanf("%d", &g); // Input jumlah game
4      while (g-->0) {
5          scanf("%d %d %d", &n, &m, &total); // Input jumlah data untuk stack a, stack b, dan total
6          int a[n], b[m];
7          for (int i = 0; i < n; i++) {
8              scanf("%d", &a[i]); // Input data untuk stack a
9          }
10         for (int i = 0; i < m; i++) {
11             scanf("%d", &b[i]); // Input data untuk stack b
12         }
13         int hasil = duaStack(total, a, b, n, m); // Memanggil fungsi duaStack untuk menghitung hasil
14         printf("%d\n", hasil); // Menampilkan hasil
15     }
16     return 0;
17 }

```

- Output

```

1
5 4 10
4 2 4 6 1
2 1 8 5
4

Process returned 0 (0x0)   execution time : 22.723 s
Press any key to continue.
|

```

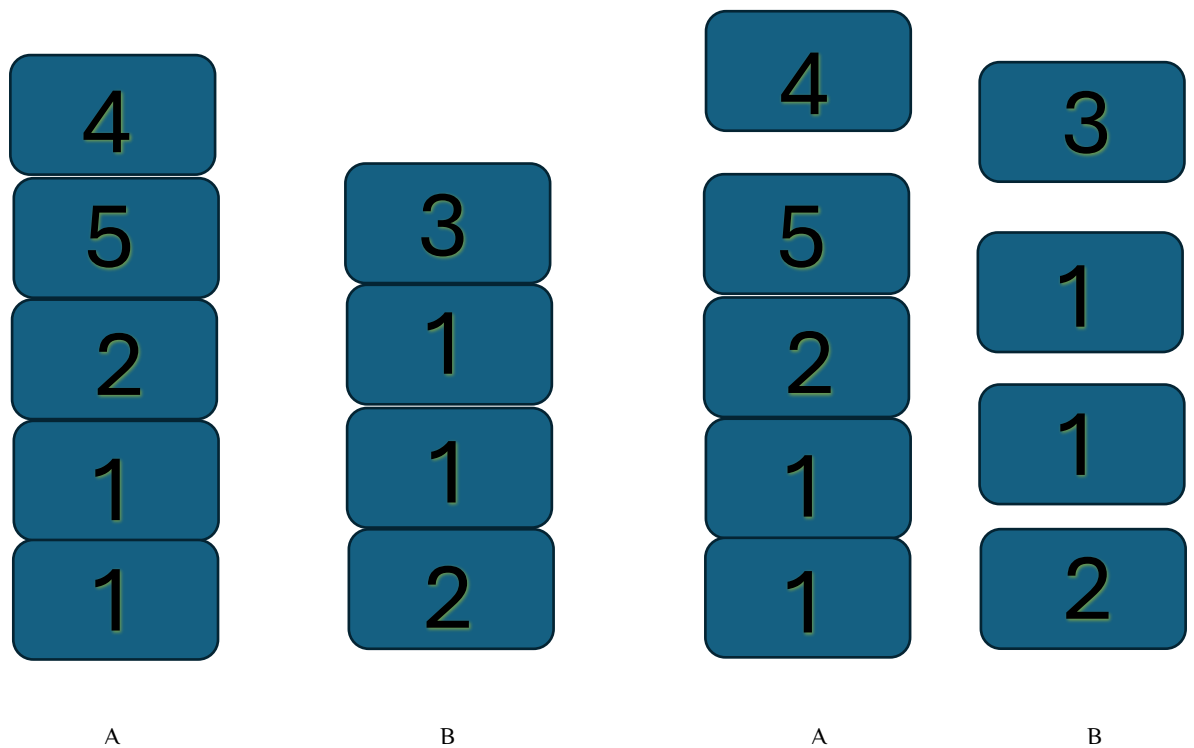
```

1
5 4 11
4 5 2 1 1
3 1 1 2
5

Process returned 0 (0x0)   execution time : 14.429 s
Press any key to continue.
|

```

- Visualisasi



Setelah melakukan operasi pop pada top A pada gambar di atas, langkah berikutnya adalah melakukan operasi pop pada top B. Kemudian, meskipun melakukan operasi pop pada top A lagi, karena jumlahnya melebihi batas maksimum, operasi pop dilanjutkan pada top B, dan seterusnya. Hal ini terus dilakukan hingga total jumlah yang diambil menjadi 11, yaitu hasil dari penjumlahan $4 + 3 + 1 + 1 + 2$.