

Nama: Ignatius Robert Cornelio S.

NIM / Kelas: 1203230071 / IF-0301

## Laporan Tugas Praktikum ASD Batu Sherlock dan Hackerrank

### 1) Batu Sherlock Holmes

#### Source Code Part 1

```
1  #include <stdio.h>
2
3  struct BatuHolmes {
4      struct BatuHolmes* link;
5      char alphabet;
6  };
7  int main() {
8      // Deklarasi/Inisialisasi Batu
9      struct BatuHolmes l1, l2, l3, l4, l5, l6, l7, l8, l9;
10     l1.link = NULL;
11     l1.alphabet = 'F';
12
13     l2.link = NULL;
14     l2.alphabet = 'M';
15
16     l3.link = NULL;
17     l3.alphabet = 'A';
18
19     l4.link = NULL;
20     l4.alphabet = 'I';
21
22     l5.link = NULL;
23     l5.alphabet = 'K';
24
25     l6.link = NULL;
26     l6.alphabet = 'T';
27
28     l7.link = NULL;
29     l7.alphabet = 'N';
30
31     l8.link = NULL;
32     l8.alphabet = 'O';
33 }
```

## Source Code Part 2

```
34     19.link = NULL;
35     19.alphabet = 'R';
36
37     // Menyambungkan link batu
38     17.link = &11; //N ke F
39     11.link = &18; //F ke O
40     18.link = &12; //O ke M
41     12.link = &15; //M ke K
42     15.link = &13; //K ke A
43     13.link = &16; //A ke T
44     16.link = &19; //T ke R
45     19.link = &14; //R ke i
46     14.link = &17; //i ke N
47
48     // Akses data huruf
49     printf("%c", 13.link->link->link->alphabet); //I
50     printf("%c", 13.link->link->link->link->alphabet); //N
51     printf("%c", 13.link->link->link->link->link->alphabet); //F
52     printf("%c", 13.link->link->link->link->link->link->alphabet); //O
53     printf("%c", 13.link->link->alphabet); //R
54     printf("%c", 13.link->link->link->link->link->link->link->alphabet); //M
55     printf("%c", 13.alphabet); //A
56     printf("%c", 13.link->alphabet); //T
57     printf("%c", 13.link->link->link->alphabet); //I
58     printf("%c", 13.link->link->link->link->link->link->link->link->alphabet); //K
59     printf("%c", 13.alphabet); //A
60
61     return 0;
62 }
```

## Output:

Menghasilkan output kata sebagai berikut:

```
PS E:\Semester2\CforASD\PRaktikum\W-4> cd "e:\Semester2\CforASD\PRaktikum\W-4\" ;
1tes }
INFORMATIKA
```

## Penjelasan:

```
1  #include <stdio.h>
2
3  struct BatuHolmes {
4      struct BatuHolmes* link;
5      char alphabet;
6  };
```

A)

- Struct yang berisi variabel character alphabet, dan pointer struct untuk link / menyambungkan batu antar batu.

```

7  ∨ int main() {
8      // Deklarasi/Inisialisasi Batu
9      struct BatuHolmes l1, l2, l3, l4, l5, l6, l7, l8, l9;
10     l1.link = NULL;
11     l1.alphabet = 'F';
12
13     l2.link = NULL;
14     l2.alphabet = 'M';
15
16     l3.link = NULL;
17     l3.alphabet = 'A';
18
19     l4.link = NULL;
20     l4.alphabet = 'I';
21
22     l5.link = NULL;
23     l5.alphabet = 'K';
24
25     l6.link = NULL;
26     l6.alphabet = 'T';
27
28     l7.link = NULL;
29     l7.alphabet = 'N';
30
31     l8.link = NULL;
32     l8.alphabet = 'O';
33
34     l9.link = NULL;
35     l9.alphabet = 'R';

```

B)

- Fungsi main, baris struct BatuHolmes untuk mendeklarasi variable batu batu.
- Baris 11 hingga 19.link mendefinisikan nilai dari link sebagai null / tidak ada sambungan.
- Baris 11 hingga 19.alphabet mendefinisikan nilai variable F, M, A, I, K, T, N, O, R, secara berturut.

```
// Menyambungkan link batu
17.link = &l1; //N ke F
11.link = &l8; //F ke O
18.link = &l2; //O ke M
12.link = &l5; //M ke K
15.link = &l3; //K ke A
13.link = &l6; //A ke T
16.link = &l9; //T ke R
19.link = &l4; //R ke i
14.link = &l7; //i ke N
```

C)

- Baris tersebut untuk menyambungkan link dari batu ke batu selanjutnya sesuai gambar soal sehingga semua batu terhubung satu sama lain.

```
// Akses data huruf
printf("%c", l3.link->link->link->alphabet); //I
printf("%c", l3.link->link->link->link->alphabet); //N
printf("%c", l3.link->link->link->link->link->alphabet); //F
printf("%c", l3.link->link->link->link->link->link->alphabet); //O
printf("%c", l3.link->link->alphabet); //R
printf("%c", l3.link->link->link->link->link->link->link->alphabet); //M
printf("%c", l3.alphabet); //A
printf("%c", l3.link->alphabet); //T
printf("%c", l3.link->link->link->alphabet); //I
printf("%c", l3.link->link->link->link->link->link->link->link->alphabet); //K
printf("%c", l3.alphabet); //A

return 0;
```

D)

- Baris printf untuk menghasilkan output yang diapit oleh tanda kutip, dengan menggunakan awalan l3 sebagai starting point untuk menghasilkan huruf yang sesuai “INFORMATIKA” dengan berapa kali link terjadi dari l3 dan diakhiri alphabet untuk mendefinisikan huruf.

## 2) Hackerrank Two Stacks

### Source Code part 1

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define maxSize 50
5
6  typedef struct {
7      int data[maxSize];
8      int top;
9  } Stack;
10
11 void initStack(Stack *stack) {
12     stack->top = -1;
13 }
14
15 void push(Stack *stack, int value) {
16     stack->data[++stack->top] = value;
17 }
18
19 int pop(Stack *stack) {
20     return stack->data[stack->top--];
21 }
22
23 int peek(Stack *stack) {
24     return stack->data[stack->top];
25 }
26
27 int isEmpty(Stack *stack) {
28     return stack->top == -1;
29 }
30
```

## Source Code Part 2

```
31 int isStackFull(Stack *stack) {
32     return stack->top == maxSize - 1;
33 }
34
35 int twoStacks(int maxSum, int a[], int b[], int x, int y) {
36     Stack stackA, stackB;
37     initStack(&stackA);
38     initStack(&stackB);
39
40     // Push elements to stacks
41     for (int i = x - 1; i >= 0; i--) {
42         push(&stackA, a[i]);
43     }
44     for (int i = y - 1; i >= 0; i--) {
45         push(&stackB, b[i]);
46     }
47
48     int total = 0, count = 0;
49     int length = x >= y ? y : x;
50
51     for (int i = 0; i < length; i++) {
52         if (total + peek(&stackA) <= maxSum) {
53             total += pop(&stackA);
54             count++;
55         }
56         if (total + peek(&stackB) <= maxSum) {
57             total += pop(&stackB);
58             count++;
59         }
60     }
```

### Source Code Part 3

```
61     return count;
62 }
63
64 int main() {
65     int total;
66     printf("Masukan berapa total game: ");
67     scanf("%d", &total);
68
69     while (total-->0) {
70         int x, y, maxSum;
71         printf("Masukan maksimal array a, b, dan maksimal pertambahan secara berturut:\n");
72         scanf("%d %d %d", &x, &y, &maxSum);
73
74         int a[x], b[y];
75         printf("Masukan angka yang ada di array a:\n");
76         for (int i = 0; i < x; i++) {
77             scanf("%d", &a[i]);
78         }
79         printf("Masukan angka yang ada di array b:\n");
80         for (int i = 0; i < y; i++) {
81             scanf("%d", &b[i]);
82         }
83         int result = twoStacks(maxSum, a, b, x, y);
84         printf("Total pengoerasian yang terjadi: %d\n", result);
85     }
86     return 0;
87 }
```

### Hasil Output

```
PS E:\Semester2\CforASD\PRaktikum\W-4> cd "e:\Semester2\CforASD\PRaktikum"
f ($?) { .\oth2game2stack }
Masukan berapa total game: 1
Masukan maksimal array a, b, dan maksimal pertambahan secara berturut:
5 4 11
Masukan angka yang ada di array a:
4 5 2 1 1
Masukan angka yang ada di array b:
3 1 1 2
Total pengoerasian yang terjadi: 5
```

### Penjelasan:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define maxSize 50
5
6  typedef struct {
7      int data[maxSize];
8      int top;
9  } Stack;
10
11 void initStack(Stack *stack) {
12     stack->top = -1;
13 }
```

a.

- Line 3 Mendeklarasi maksimal array dari stack.
- Line 6 Mendefinisikan struktur stack yaitu array int data dengan ukuran maxSize, dan int top.
- Line 11, menginisialisasi stack dengan ukuran -1 yang menandakan stack kosong.

```
15 void push(Stack *stack, int value) {
16     stack->data[++stack->top] = value;
17 }
18
19 int pop(Stack *stack) {
20     return stack->data[stack->top--];
21 }
22
23 int peek(Stack *stack) {
24     return stack->data[stack->top];
25 }
26
27 int isEmpty(Stack *stack) {
28     return stack->top == -1;
29 }
30
31 int isStackFull(Stack *stack) {
32     return stack->top == maxSize - 1;
33 }
```

b.

- Fungsi push untuk menambahkan elemen value ke dalam stack dengan menaikkan top terlebih dahulu.
- Fungsi pop untuk menghapus dan mengembalikan elemen teratas dari stack, serta menurunkan nilai top setelah penghapusan.
- fungsi peek untuk mengembalikan nilai elemen teratas dari stack tanpa menghapus.
- Mendefinisikan fungsi isEmpty untuk memeriksa apakah stack kosong dengan memeriksa apakah nilai top sama dengan -1.



- Mendefinisikan fungsi `isStackFull` untuk memeriksa apakah stack penuh dengan memeriksa apakah `top` sama dengan `maxSize - 1`.

```

34
35 int twoStacks(int maxSum, int a[], int b[], int x, int y) {
36     Stack stackA, stackB;
37     initStack(&stackA);
38     initStack(&stackB);
39
40     for (int i = x - 1; i >= 0; i--) {
41         push(&stackA, a[i]);
42     }
43     for (int i = y - 1; i >= 0; i--) {
44         push(&stackB, b[i]);
45     }
46
47     int total = 0, count = 0;
48     int length = x >= y ? y : x;
49
50     for (int i = 0; i < length; i++) {
51         if (total + peek(&stackA) <= maxSum) {
52             total += pop(&stackA);
53             count++;
54         }
55         if (total + peek(&stackB) <= maxSum) {
56             total += pop(&stackB);
57             count++;
58         }
59     }
60     return count;
61 }

```

c.

Mendefinisikan fungsi `twoStacks` untuk menghitung jumlah elemen yang dapat diambil dari kedua stack sehingga totalnya tidak melebihi `maxSum`. Dalam fungsi ini, stack `stackA` dan `stackB` diinisialisasi dan elemen-elemen dari array `a` dan `b` dimasukkan ke dalamnya. Kemudian, elemen-elemen diambil dari kedua stack secara bergantian hingga total pengoperasian tidak melebihi dari `maxSum`.

```

63 int main() {
64     int total;
65     printf("Masukan berapa total game: ");
66     scanf("%d", &total);
67
68     while (total-->0) {
69         int x, y, maxSum;
70         printf("Masukan maksimal array a, b, dan maksimal pertambahan secara berturut:\n");
71         scanf("%d %d %d", &x, &y, &maxSum);
72
73         int a[x], b[y];
74         printf("Masukan angka yang ada di array a:\n");
75         for (int i = 0; i < x; i++) {
76             scanf("%d", &a[i]);
77         }
78         printf("Masukan angka yang ada di array b:\n");
79         for (int i = 0; i < y; i++) {
80             scanf("%d", &b[i]);
81         }
82         int result = twoStacks(maxSum, a, b, x, y);
83         printf("Total pengoerasian yang terjadi: %d\n", result);
84     }
85     return 0;
86 }

```

d.

- Mendeklarasi int total, dan menerima inputan dari user.
- Loop pengurangan total, mendeklarasikan nilai x,y,maxsum,a[x],b[y].
- Line 75, Memasukkan elemen array a dari input user menggunakan loop for.
- Line 78, Memasukkan elemen array b dari input user menggunakan loop for.
- Line 82, Memanggil fungsi twoStacks dengan parameter maxSum, a, b, x, dan y, dan menyimpan hasilnya ke dalam variabel result.

4
5
2
1
1

3
1
1
2

- Dalam 1 game, terdapat dua array dengan ukuran maksimal size  $a = 5$ ,  $b = 4$ , dan maksimal pengoperasian = 11.
- Array a berisi = 4,5,2,1,1
- Array b berisi = 3,1,1,2
- Pengoperasian berjalan mengambil top A, 4 ditambah ke sum yaitu 0 sehingga menjadi 4. Selanjutnya ditambahkan top b yaitu 3, menjadi 7. Lalu ditambahkan lagi ke 5 tetapi karena melebihi maksimal operasi maka lanjut 7 tambah 1 sehingga 8, lalu ditambah 2 menjadi 10, dan ditambah 1 menjadi 11 sehingga jumlah operasi yang terjadi ada 5.