Nama:M.AGUNG BALYA

NIM:1203230095

Kelas:IF 03-01

No.1

Code

```c
#include <stdio.h>

struct Stone {
    char* alphabet;
    struct Stone* link;
};

void initialize(struct Stone* stone, char* letter) {
    stone->link = NULL;
    stone->alphabet = letter;
}

int main() {
    struct Stone stone1, stone2, stone3, stone4, stone5, stone6, stone7, stone8, stone9;

    initialize(&stone1, "F");
    initialize(&stone2, "M");
    initialize(&stone3, "A");
    initialize(&stone4, "I");
    initialize(&stone5, "K");
    initialize(&stone6, "T");
    initialize(&stone7, "N");
    initialize(&stone8, "O");
    initialize(&stone9, "R");

    stone3.link = &stone6;
    stone6.link = &stone9;
    stone9.link = &stone4;
    stone4.link = &stone7;
    stone7.link = &stone1;
    stone1.link = &stone8;
    stone8.link = &stone2;
    stone2.link = &stone5;
    stone5.link = &stone3;

    printf("%s", stone3.link->link->link->alphabet);
    printf("%s", stone3.link->link->link->link->alphabet);
```

```c
    initialize(&stone8, "O");
    initialize(&stone9, "R");

    stone3.link = &stone6;
    stone6.link = &stone9;
    stone9.link = &stone4;
    stone4.link = &stone7;
    stone7.link = &stone1;
    stone1.link = &stone8;
    stone8.link = &stone2;
    stone2.link = &stone5;
    stone5.link = &stone3;

    printf("%s", stone3.link->link->link->alphabet);
    printf("%s", stone3.link->link->link->link->alphabet);
    printf("%s", stone3.link->link->link->link->link->alphabet);
    printf("%s", stone3.link->link->link->link->link->link->alphabet);
    printf("%s", stone3.link->link->alphabet);
    printf("%s", stone3.link->link->link->link->link->link->link->alphabet);
    printf("%s", stone3.alphabet);
    printf("%s", stone3.link->alphabet);
    printf("%s", stone3.link->link->link->alphabet);
    printf("%s", stone3.link->link->link->link->link->link->link->link->alphabet);
    printf("%s", stone3.alphabet);

    printf("\n\nStack Visualization:\n");

    struct Stone* current = &stone3;

    return 0;
}
```

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\agung\Februari\Week 7> cd "c:\agung\Februari\Week 7\" ; if ($?) { gcc OTH1.c -o OTH1 } ; if ($?) { .\OTH1 }
INFORMATIKA

Stack Visualization:
PS C:\agung\Februari\Week 7>
```

Penjelasan

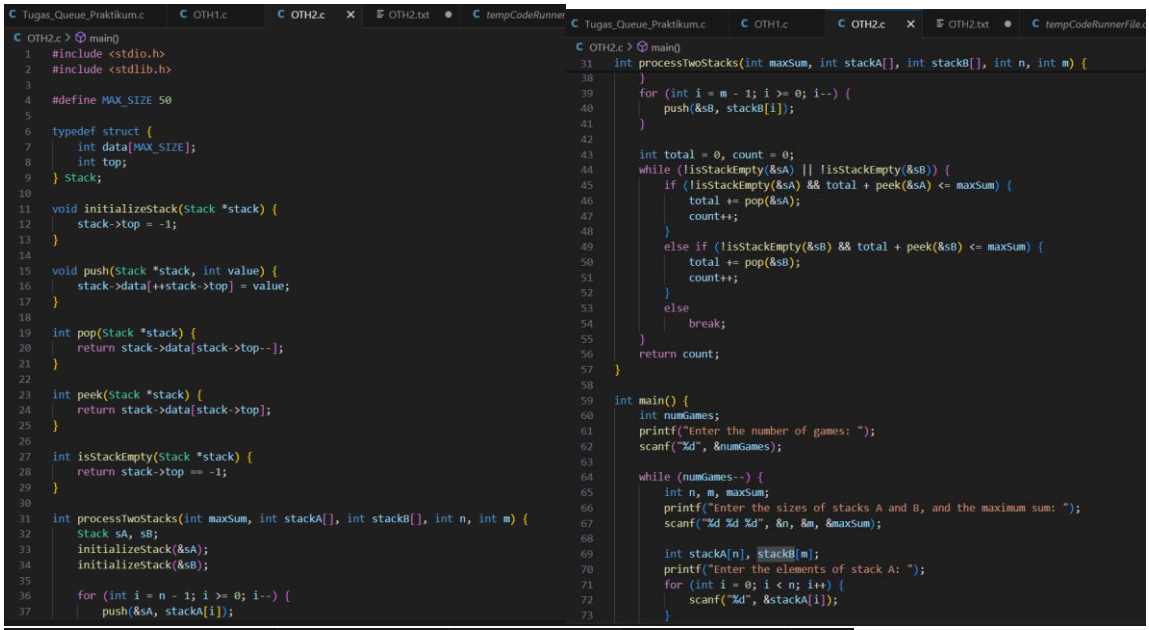
Mendeklarasikan sembilan variabel bertipe struct Stone yang mewakili sembilan batu.

Mencetak huruf-huruf yang terhubung dengan menggunakan pointer dan akses ke variabel alphabet dari batu-batu tersebut.

Menampilkan visualisasi stack dengan mendeklarasikan pointer ke struct Stone dan menginisialisasinya dengan alamat dari batu ketiga.

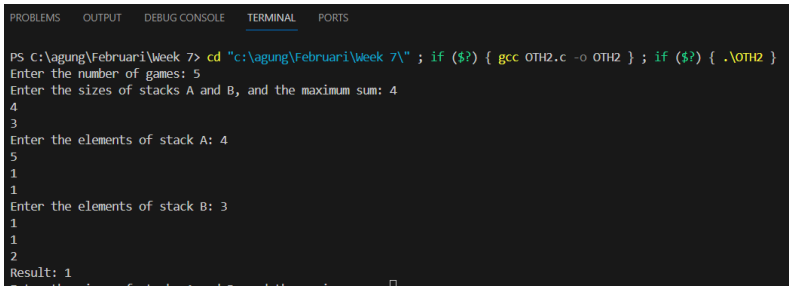
No.2

Kode



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

#define MAX_SIZE 50

typedef struct {
    int data[MAX_SIZE];
    int top;
} Stack;

void initializeStack(Stack *stack) {
    stack->top = -1;
}

void push(Stack *stack, int value) {
    stack->data[++stack->top] = value;
}

int pop(Stack *stack) {
    return stack->data[stack->top--];
}

int peek(Stack *stack) {
    return stack->data[stack->top];
}

int isStackEmpty(Stack *stack) {
    return stack->top == -1;
}

int processTwoStacks(int maxSum, int stackA[], int stackB[], int n, int m) {
    Stack sA, sB;
    initializeStack(&sA);
    initializeStack(&sB);

    for (int i = n - 1; i >= 0; i--) {
        push(&sA, stackA[i]);
    }
    for (int i = m - 1; i >= 0; i--) {
        push(&sB, stackB[i]);
    }

    int total = 0, count = 0;
    while (!isStackEmpty(&sA) || !isStackEmpty(&sB)) {
        if (!isStackEmpty(&sA) && total + peek(&sA) <= maxSum) {
            total += pop(&sA);
            count++;
        }
        else if (!isStackEmpty(&sB) && total + peek(&sB) <= maxSum) {
            total += pop(&sB);
            count++;
        }
        else
            break;
    }
    return count;
}

int main() {
    int numGames;
    printf("Enter the number of games: ");
    scanf("%d", &numGames);

    while (numGames--) {
        int n, m, maxSum;
        printf("Enter the sizes of stacks A and B, and the maximum sum: ");
        scanf("%d %d %d", &n, &m, &maxSum);

        int stackA[n], stackB[m];
        printf("Enter the elements of stack A: ");
        for (int i = 0; i < n; i++) {
            scanf("%d", &stackA[i]);
        }
        printf("Enter the elements of stack B: ");
        for (int i = 0; i < m; i++) {
            scanf("%d", &stackB[i]);
        }

        int result = processTwoStacks(maxSum, stackA, stackB, n, m);
        printf("Result: %d\n", result);
    }

    return 0;
}
```

Output



Penjelasan

typedef struct {} Ini mendefinisikan sebuah struktur data baru bernama Stack, yang terdiri dari array data dengan ukuran MAX_SIZE untuk menyimpan elemen-elemen tumpukan, dan variabel top yang menunjukkan indeks atas (terbaru) dari tumpukan.

void initializeStack Ini adalah fungsi yang digunakan untuk menginisialisasi tumpukan. Ketika dipanggil, variabel top dalam struktur Stack diatur menjadi -1, menunjukkan bahwa tumpukan kosong.

void push Ini adalah fungsi untuk menambahkan elemen baru ke dalam tumpukan. Nilai value ditambahkan ke dalam array data pada indeks top yang kemudian diinkremen, menunjukkan penambahan elemen baru ke dalam tumpukan.

int peek Ini adalah fungsi yang digunakan untuk melihat nilai teratas dari tumpukan tanpa menghapusnya. Ini hanya mengembalikan nilai dari array data pada indeks top.

int isStackEmpty Ini adalah fungsi yang mengembalikan nilai 1 jika tumpukan kosong (top sama dengan -1), dan 0 jika tidak.

Di dalam fungsi main(), program meminta input jumlah permainan (numGames), kemudian loop while akan berjalan sebanyak jumlah permainan tersebut.

Di dalam loop, program meminta input ukuran tumpukan dan nilai-nilai tumpukan A dan B untuk setiap permainan.

Setelah mendapatkan input, program memanggil fungsi processTwoStacks() untuk memproses kedua tumpukan, dan mencetak jumlah elemen yang diambil dari kedua tumpukan.