Nama : Khusnia Fitri

Kelas : IF-03-01

NIM : 1203230030

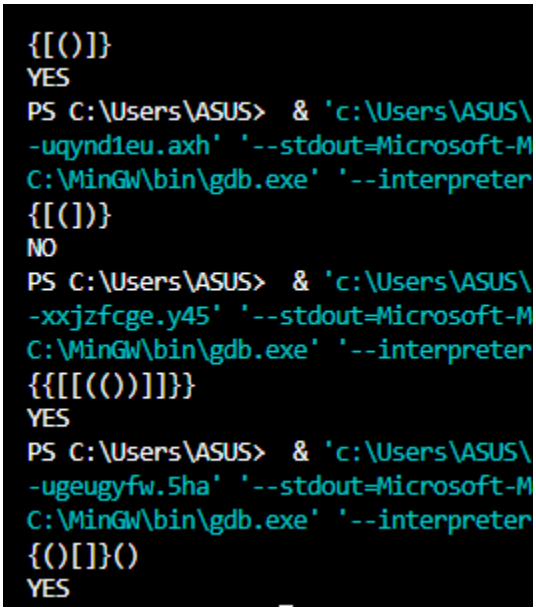
Kode :

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

struct Stack {
    int top;
    unsigned capacity;
    int* array;
};

struct Stack* createStack(unsigned capacity) {
    struct Stack* stack = (struct Stack*) malloc(sizeof(struct Stack));
    if (!stack) return NULL;
    stack->top = -1;
    stack->capacity = capacity;
    stack->array = (int*) malloc(stack->capacity * sizeof(int));
    if (!stack->array) return NULL;
    return stack;
}

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

void push(struct Stack* stack, char item) {
    stack->array[++stack->top] = item;
}

char pop(struct Stack* stack) {
    if (!isEmpty(stack))
        return stack->array[stack->top--];
    return '$';
}

int isOpening(char ch) {
    return ch == '(' || ch == '{' || ch == '[';
}
```

```c
int isClosing(char ch) {
    return ch == ')' || ch == '}' || ch == ']';
}

int isMatch(char a, char b) {
    return (a == '(' && b == ')') || (a == '{' && b == '}') || (a == '[' && b
== ']');
}

int isBalanced(char* expr) {
    struct Stack* stack = createStack(strlen(expr));
    if (!stack) return 0;

    for (int i = 0; i < strlen(expr); i++) {
        if (isOpening(expr[i])) {
            push(stack, expr[i]);
        } else if (isClosing(expr[i])) {
            if (isEmpty(stack) || !isMatch(pop(stack), expr[i])) {
                free(stack->array);
                free(stack);
                return 0;
            }
        }
    }

    if (!isEmpty(stack)) {
        free(stack->array);
        free(stack);
        return 0;
    }

    free(stack->array);
    free(stack);
    return 1;
}

int main() {
    char expression[100];
    scanf("%s", expression);

    if (isBalanced(expression)) {
        printf("YES\n");
    } else {
        printf("NO\n");
    }
```

```
    return 0;
}
```

Output :



Penjelasan :

1. Struktur Stack: struktur data yang disebut Stack, digunakan untuk menyimpan tanda kurung saat diperiksa. Terdapat: **top** (menunjukkan indeks elemen teratas), **capacity** (menunjukkan kapasitas maksimum tumpukan), dan **array** (array untuk menyimpan elemen tumpukan).
2. Fungsi:
    1. Membuat Stack: createStack adalah fungsi yang digunakan untuk membuat tumpukan baru, mengalokasikan memori untuk stack dan menginisialisasi nilai-nilai awalnya.
    2. Mengecek Apakah Stack Kosong: isEmpty adalah fungsi yang memeriksa apakah stack kosong atau tidak. Jika top adalah -1, maka tumpukan dianggap kosong.
    3. Menambahkan Elemen ke Stack: push adalah fungsi yang menambahkan elemen ke dalam stack. Elemen ditambahkan pada indeks **top+1.**
    4. Mengeluarkan Elemen dari Stack: pop adalah fungsi yang mengeluarkan elemen teratas dari stack, mengurangi **top** dan mengembalikan elemen yang dihapus.
    5. Fungsi lainnya: **isOpening, isClosing, dan isMatch**. Ini digunakan untuk memeriksa apakah suatu karakter merupakan tanda kurung pembuka atau penutup, dan apakah pasangan tanda kurung tersebut cocok atau tidak.

    a. **isOpening** digunakan untuk memeriksa apakah karakter adalah tanda kurung pembuka ('(', '{', atau '[').
    b. **isClosing** digunakan untuk memeriksa apakah karakter adalah tanda kurung penutup (')', '}', atau ']').
    c. **isMatch** digunakan untuk memeriksa apakah tanda kurung pembuka dan penutup cocok (berpasangan).
6. isBalanced: Fungsi utama dalam program, fungsi untuk mengecek apakah setiap tanda kurung yang dibuka sesuai atau cocok dengan tanda kurung penutup. ini adalah isBalanced. Ini melakukan pemeriksaan keseimbangan tanda kurung.
7. main: Fungsi main menerima input, yaitu string yang berisi tanda kurung. Kemudian, dengan memanggil fungsi isBalanced untuk memeriksa tanda kurung dalam string tersebut. Hasilnya kemudian dicetak sebagai "YES" jika tanda kurung sudah sesuai, dan "NO" jika tidak.