//Ankit Kumar 2K20/CO/72 BalancedParenthesis.c

#include <stdio.h>

#include <stdlib.h>

#define MAX\_SIZE 1000

struct stack {

    char \*a;

    int size;

    int top;

};

int isempty(struct stack \*st) { return st->top == -1; }

void push(struct stack \*st, char x) {

    st->top++;

    st->a[st->top] = x;

}

void pop(struct stack \*st) {

    int x = st->a[st->top];

    st->top--;

}

int balanced(char str[]) {

    struct stack \*st = (struct stack \*)malloc(sizeof(struct stack));

    st->a = (char \*)malloc(sizeof(char) \* MAX\_SIZE);

    st->top = -1;

    st->size = MAX\_SIZE;

    for (int i = 0; str[i] != '\0'; i++) {

        if (str[i] == '(') push(st, str[i]);

        else if (str[i] == ')') {

            if (isempty(st)) return 0;

            pop(st);

        }

    }

    return isempty(st);

}

int main() {

    char str[MAX\_SIZE];

    printf("\nEnter Parenthesis:");

    scanf("%s", str);

    if (balanced(str)) printf("The Parenthesis are balanced\n");

    else

        printf("Parenthesis are not balanced!\n");

    return 0;

}

