1.main.c

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "header.h"


int main(){

   char exp[100];

   printf("Enter an expression: ");

   scanf("%s", exp);

   if(paranthesisBalanced(exp)){

      printf("The expression is balanced.\n");

   }else{

      printf("The expression is not balanced.\n");

   }

   return 0;

}
```

2.header.h

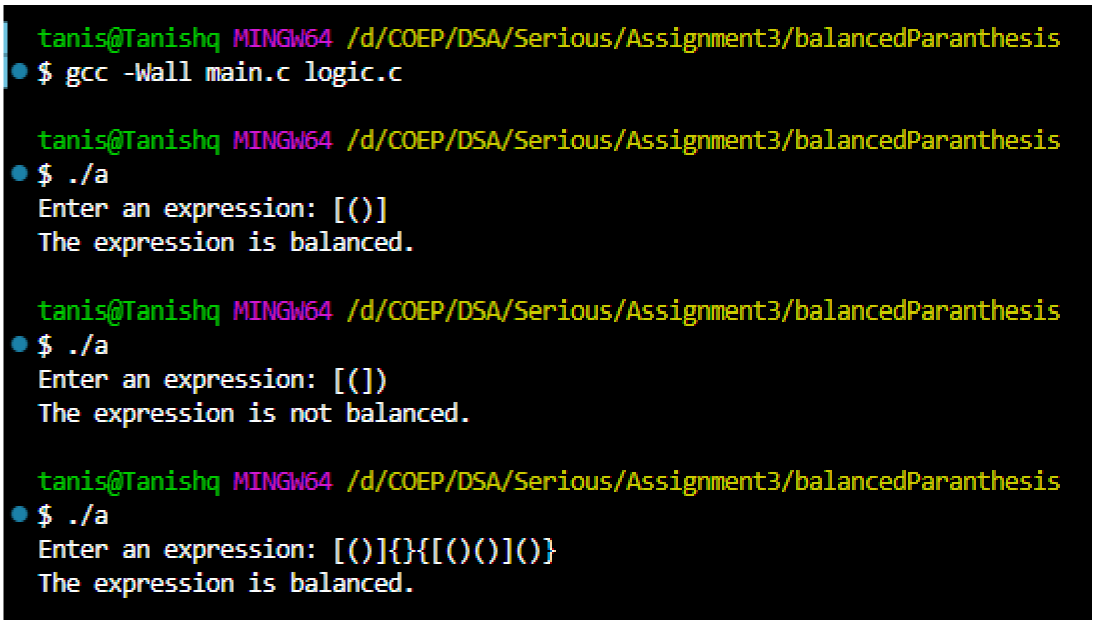
```
typedef struct {

   int top;

   char arr[100];

} Stack;


void init(Stack *s);

int isFull(Stack *s);

int isEmpty(Stack *s);

void push(Stack *s, int value);

void pop(Stack *s);

char peek(Stack *s);

int isParanthesisMatched(char char1, char char2);

int paranthesisBalanced(char exp[]);
```

3.logic.c

```
#include <stdio.h>
#include <stdlib.h>
#include "header.h"

void init(Stack *s) {
    s->top = -1;
}

int isFull(Stack *s) {
    return s->top == 99; // Adjusted for MAX = 100
}

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

void push(Stack *s, int value) {
    if (isFull(s)) {
        printf("Stack overflow\n");
        return;
    }
    s->arr[++s->top] = value;
}
void pop(Stack *s) {
    if (isEmpty(s)) {
        printf("Stack underflow\n");
    }
    s->top--;
}
```

```
char peek(Stack *s) {
    if (isEmpty(s)) {
        return 0;
    }
    return s->arr[s->top];
}
int isParanthesisMatched(char char1, char char2){
    if(char1 == '(' && char2 == ')') return 1;
    if(char1 == '[' && char2 == ']') return 1;
    if(char1 == '{' && char2 == '}') return 1;
    return 0;
}
int paranthesisBalanced(char exp[]){
    Stack Characters;
    init(&Characters);
    int i = 0;
    while(exp[i] != '\0'){
        if(exp[i] == '(' || exp[i] == '[' || exp[i] == '{'){
            push(&Characters, exp[i]);
        }
        if(exp[i] == ')' || exp[i] == ']' || exp[i] == '}'){
            if(isEmpty(&Characters) || !isParanthesisMatched(peek(&Characters), exp[i])){
                return 0;
            }else{
                pop(&Characters);
            }
        }
        i++;
    }
    return isEmpty(&Characters);
}
```

Output:



```
tanis@Tanishq MINGW64 /d/COEP/DSA/Serious/Assignment3/balancedParanthesis
$ gcc -Wall main.c logic.c

tanis@Tanishq MINGW64 /d/COEP/DSA/Serious/Assignment3/balancedParanthesis
$ ./a
Enter an expression: [()]
The expression is balanced.

tanis@Tanishq MINGW64 /d/COEP/DSA/Serious/Assignment3/balancedParanthesis
$ ./a
Enter an expression: [(])
The expression is not balanced.

tanis@Tanishq MINGW64 /d/COEP/DSA/Serious/Assignment3/balancedParanthesis
$ ./a
Enter an expression: [()]{}{[()()]()}
The expression is balanced.
```