

# Assignment 3

Two stack

## Header.h

```
typedef struct twoStacks{
    char *data;
    int size;
    int top1;
    int top2;
} twoStacks;

void init(twoStacks *s, int size);
void push1(twoStacks *s, char value);
int pop1(twoStacks *s);
void push2(twoStacks *s, char value);
int pop2(twoStacks *s);
int peek(twoStacks *s);
void display1(twoStacks *s);
void display2(twoStacks *s);
```

## Logic.c

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

void init(twoStacks *s, int size) {
    s->data = (char *)malloc(size * sizeof(char));
    s->size = size;
    s->top1 = -1;
    s->top2 = size;
}

void push1(twoStacks *s, char value) {
    if (s->top1 < s->top2 - 1) {
        s->data[++(s->top1)] = value;
    } else {
```

```

        printf("Stack Overflow: No space to push %c in Stack 1\n",
value);
    }
}

void push2(twoStacks *s, char value) {
    if (s->top1 < s->top2 - 1) {
        s->data[--(s->top2)] = value;
    } else {
        printf("Stack Overflow: No space to push %c in Stack 2\n",
value);
    }
}

int pop1(twoStacks *s) {
    if (s->top1 >= 0) {
        return s->data[(s->top1)--];
    } else {
        printf("Stack Underflow: No element to pop from Stack
1\n");
        return INT_MIN;
    }
}

int pop2(twoStacks *s) {
    if (s->top2 < s->size) {
        return s->data[(s->top2)++];
    } else {
        printf("Stack Underflow: No element to pop from Stack
2\n");
        return INT_MIN;
    }
}

int peek1(twoStacks *s) {
    if (s->top1 >= 0) {
        return s->data[s->top1];
    } else {
        printf("Stack Underflow: No element to peek in Stack
1\n");
        return INT_MIN;
    }
}

int peek2(twoStacks *s) {
    if (s->top2 < s->size) {
        return s->data[s->top2];
    } else {
        printf("Stack Underflow: No element to peek in Stack
2\n");
        return INT_MIN;
    }
}

```

```

}

void display1(twoStacks *s) {
    if (s->top1 >= 0) {
        for (int i = 0; i <= s->top1; i++) {
            printf("%d ", s->data[i]);
        }
        printf("\n");
    } else {
        printf("Stack 1 is empty\n");
    }
}

void display2(twoStacks *s) {
    if (s->top2 < s->size) {
        for (int i = s->top2; i < s->size; i++) {
            printf("%d ", s->data[i]);
        }
        printf("\n");
    } else {
        printf("Stack 2 is empty\n");
    }
}

```

## Main.c

```

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

int main(){
    twoStacks s;
    init(&s,10);
    push1(&s, 5);
    push2(&s, 10);
    push1(&s, 15);
    push2(&s, 20);
    display1(&s);
    display2(&s);
    printf("Popped element from stack1: %d\n", pop1(&s));
    printf("Popped element from stack2: %d\n", pop2(&s));
}

```