

main.c

```
1  #include<stdio.h>
2  #define N 5
3  int top1=-1;
4  int top2=-1;
5  int top3=-1;
6  void push1(int[], int);
7  int pop1(int[]);
8  void display1(int[]);
9  void push2(int[], int);
10 int pop2(int[]);
11 void display2(int[]);
12 void merge(int[],int[],int[]);
13 void push3(int[],int);
14 void display3(int[]);
15
16
17 int main(int argc, char **argv)
18 {
19     int stack1[N];
20     int stack2[N];
21     int stack3[2*N];
22     int choice1=0,choice2=0,element;
23     printf("Please choose options:\n");
24     printf("1 - Stack1\n");
25     printf("2 - Stack2\n");
26     printf("3 - Push merged elements to Stack3\n");
27     printf("4 - Display Stack3\n");
28     printf("5 - Quit\n");
29     while(choice1!=5)
```



main.c

```
30 {
31     printf("-----\n");
32     printf("Please enter your choice:");
33     scanf("%d",&choice1);
34     if(choice1 == 1)
35     {
36         printf("Stack 1\n");
37         printf("1 - Push\n");
38         printf("2 - Pop\n");
39         printf("3 - Display\n");
40         printf("4 - Quit from stack1\n");
41         while(choice2!=4)
42         {
43             printf("Please enter your choice:");
44             scanf("%d",&choice2);
45             if(choice2==1)
46             {
47                 printf("Enter an element to push to stack1:");
48                 scanf("%d",&element);
49                 push1(stack1,element);
50             }
51             else if(choice2==2)
52             {
53                 int num = pop1(stack1);
54                 printf("The popped element:%d",num);
55             }
56             else if(choice2==3)
57             {
58                 display1(stack1);
59             }
```



main.c

```
60         else if(choice2==4)
61             break;
62         else
63             printf("Invalid choice");
64     }
65 }
66
67 else if(choice1 == 2)
68 {
69     choice2 = 0;
70     printf("Stack 2\n");
71     printf("1 - Push\n");
72     printf("2 - Pop\n");
73     printf("3 - Display\n");
74     printf("4 - Quit from stack2\n");
75     while(choice2!=4)
76     {
77         printf("Please enter your choice:");
78         scanf("%d",&choice2);
79         if(choice2==1)
80         {
81             printf("Enter an element to push to stack2:");
82             scanf("%d",&element);
83             push2(stack2,element);
84         }
85         else if(choice2==2)
86         {
87             int num = pop2(stack2);
88             printf("The popped element:%d",num);
```



main.c

```
89         }
90         else if(choice2==3)
91         {
92             display2(stack2);
93         }
94         else if(choice2==4)
95             break;
96         else
97             printf("Invalid choice");
98     }
99 }
100 else if(choice1 == 3)
101 {
102     merge(stack1,stack2,stack3);
103 }
104 else if(choice1 == 4)
105 {
106     display3(stack3);
107     top3 = 0;
108 }
109 else if(choice1 == 5)
110 {
111     break;
112 }
113
114 }
115 return 0;
116 }
117
118 void push1(int stack[], int element)
```



main.c

```
115 | return 0;
116 | }
117
118 | void push1(int stack[], int element)
119 | {
120 |     if(top1==N-1)
121 |         printf("Stack Overflow\n");
122 |     else
123 |     {
124 |         top1++;
125 |         stack[top1]=element;
126 |     }
127 | }
128
129 | int pop1(int stack[])
130 | {
131 |     int num;
132 |     if(top1== -1){
133 |         printf("Stack Underflow\n");
134 |         return -1;}
135 |     else
136 |     {
137 |         num=stack[top1];
138 |         top1--;
139 |         return num;
140 |     }
141 | }
142 |
143 | void display1(int stack[])
144 | {
```

```
int i;
printf("The elements in stack1:\n");
for(i=top1;i>=0;i--)
{
    printf("%d\n",stack[i]);
}
```

```
void push2(int stack[], int element)
{
    if(top2==N-1)
        printf("Stack Overflow\n");
    else
    {
        top2++;
        stack[top2]=element;
    }
}
```

```
int pop2(int stack[])
{
    int num;
    if(top2==-1){
        printf("Stack Underflow\n");
        return -1;}
    else
    {
        num=stack[top2];
        top2--;
```



main.c

```
177 }
178 void display2(int stack[])
179 {
180     int i;
181     printf("The elements in stack2:\n");
182     for(i=top2;i>=0;i--)
183     {
184         printf("%d\n",stack[i]);
185     }
186 }
187
188 void merge(int stack1[], int stack2[], int stack3[])
189 {
190     int i;
191
192     if(top1 > top2)
193     {
194         int num = top1 - top2;
195         for(i=0;i<num;i++)
196         {
197             push3(stack3,stack1[i]);
198         }
199         for(i=num;i<=top1;i++)
200         {
201             push3(stack3,stack1[i]+stack2[top1-i]);
202         }
203     }
204     else if(top2 > top1)
205     {
206         int num = top2 - top1;
```



main.c

```
203 }
204 else if(top2 > top1)
205 {
206     int num = top2 - top1;
207
208     for(i=0;i<=top1;i++)
209     {
210         push3(stack3,stack1[i]+stack2[top2-i]);
211     }
212     for(i=num-1;i>=0;i--)
213     {
214         push3(stack3,stack2[i]);
215     }
216 }
217 else if(top2 == top1)
218 {
219     for(i=0;i<=top1;i++)
220     {
221         push3(stack3,stack1[i]+stack2[top1-i]);
222     }
223 }
224 }
225
226 void push3(int stack[], int element)
227 {
228     if(top3==(2*N-1))
229         printf("Stack Overflow\n");
230     else
231     {
232         top3++;
```



main.c

```
222     }
223 }
224 }
225
226 void push3(int stack[], int element)
227 {
228     if(top3==(2*N-1))
229         printf("Stack Overflow\n");
230     else
231     {
232         top3++;
233         stack[top3]=element;
234     }
235 }
236
237 void display3(int stack[])
238 {
239     int i;
240     printf("The elements in stack3:\n");
241     for(i=0;i<=top3;i++)
242     {
243         printf("%d\n",stack[i]);
244     }
245 }
```