Akshaya Deep Prasad
1BM19CS013

Lab Test-1 ~~(C++)~~ (DS)

```
#include<stdio.h>
#include<stdlib.h>
#define maxsize 5
#define maxsize2 5
int
```

Q1.b The first element in stack1 should be merged with the last element of stack2, the second element of stack1 to the second last element of stack2 and so on. If an element in stack1 or stack2 is none, then the corresponding element in the other stack should be kept as it is in ~~the other~~ the merged stack.

Sol:-

```
#include<stdio.h>
#include<stdlib.h>
#define size 5
int top1=-1, top2=-1
int stack1[size1], stack2[size2];
void push1(int ele)
{
    if(top1==size-1)
        printf("Stack is in overflow condition\n");
    else
    {
        top1++;
        stack[top1]=ele;
    }
}
int pop1()
{
    if(top1==-1)
```

```
6
    top1++;
    stack[top1]=ele;
}
else
{
printf("stack underflow\n");
return -1;
}
else
{
    int x = stack[top];
    top1--;
    return x;
}
}
void display1()
{
    int i;
    if(top1==-1)
    {
      printf("stack is empty \n");}
    else
    {
        for(i=top1;i>=0;i--)
        {
          printf("%d \n", stack1[i]);
        }
    }
}
void push2(int ele)
{
    if(top2==size-1)
        printf("stack is in overflow condition\n");
    else
    {
        top2++;
        stack2[top2]= ele
    }
}
```

```
int pop2()
{
    if(top2 == -1)
    {
      printf("stack underflow \n");
      return -1;
    }
    else
    {
       int x = stack2[top2];
       top2--;
       return x;
    }
}
void display2()
{
  int i;
  if (top2 == -1)
  {    printf("stack is empty \n");
  else
  {
      for (i = top2; i >= 0; i--);
          printf ("%d \n", stack2[i]);
      }
  }
void merge()
{
    int t1, t2, i;
    int s1(size), s2[size];
    t1 = 0;
    t2 = -1;
    for( i = top2; i >= 0; i--)
    {
       t2++;
       s2[t2] = stack2[i]
    }
    int mergedlist[size]
```

```
t1 = top1;
i = 0;
while (t1 >= 0 && t2 >= 0)
{
        mergedlist[i] = stack1[t1] + s2[t2];
        t1--;
        t2--;
        i++
}
while (t1 >= 0)
{
    mergedlist[i] = stack1[t1]
    t1--;
    i++
}
while (t2 >= 0)
{
    mergedlist[i] = s2[t2];
    t2--;
    i++;
}
printf("Merged list is: \n");
for (int j = 0; j < i; j++)
    printf("%d \n", mergedlist[j]);
}
int main()
{
    int x = 1, ele;
    int i = 0;
    printf("Enter the elements of stack1: \n");
    while (x != 0)
    {
```

```
printf("\n Menu\n");
printf("\n 1. Push \n 2. Pop \n 3. Display \n 4. Exit ");
printf("Enter your choice\n");
scanf("%d", &x);
if (x==4)
    break;
if (x==1)
{
    printf("Enter the element:\n");
    scanf("%d", &ele);
    push1(ele);
}
else if(x==2)
{
    int popele = pop1();
    printf("Element popped is : %d", popele);
}
else if(x==3)
{
    printf("The stack is:\n");
    display1();
}
}
x=1;
printf("Enter the elements of stack 2 \n");
while (x!=4)
{
    printf("\n menu\n");
    printf("1. Push \n 2. Pop \n 3. Diplay \n 4. Exit ");
    printf("Enter your choice\n");
    scanf("%d", &x);
    if(x==4)
        break;
```

```
if(n==1)
{
    printf("Enter the element\n");
    scanf("%d",&ele);
    push2(ele);
}
else if(n==2)
{
    int popele=pop2(stack1);
    printf("Popped element is:\n%d",popele);
}
else if(n==3)
{
    printf("The stack is:\n");
    display2();
}
}
merged();
return 0;
}
```