```c
#include<stdio.h>
#include<stdlib.h>

struct stackArray
{
    int top1;
    int top2;
    int capacity;
    int *stack1;
};

struct stackArray * createStack(int capacity)
{
    struct stackArray *S=(struct stackArray *)malloc(sizeof(struct stackArray));
    S->top1=-1;
    S->top2=capacity;
    S->capacity=capacity;
    S->stack1=(int *)malloc(sizeof(int)*capacity);
    return S;
}

void pushStack1(struct stackArray *s,int value)
{
    if(s->top1==s->top2-1)
    {
        printf("stack is full");
        return;
    }
    else
    {
        s->top1++;
        s->stack1[s->top1]=value;
        return;
    }

}

void pushStack2(struct stackArray *s,int value)
{
    if(s->top2==s->top1+1)
    {
        printf("stack is full");
        return;
    }
    else
    {
        s->top2--;
        s->stack1[s->top2]=value;
        return;
    }

}

int pop1(struct stackArray *s)
{
    if(s->top1==-1)
    {
        printf("stack Underflow");
        return NULL;
    }
    else
    {
        return s->stack1[s->top1];
        s->top1--;

    }
```

```c
}

int pop2(struct stackArray *s)
{
    if(s->top2==s->capacity)
    {
        printf("stack Underflow");
        return NULL;
    }
    else
    {
        return s->stack1[s->top2];
        s->top1++;

    }
}



int main()
{
    struct stackArray *s=createStack(5);
     pushStack1(s,5);
     pushStack1(s,4);
     pushStack1(s,3);
     pushStack2(s,2);
     pushStack2(s,1);
     printf("element poped from 1st stack is %d",pop1(s));
     printf("\nelement poped from 2nd stack is %d",pop2(s));

}
```