Stack Implementation

Example

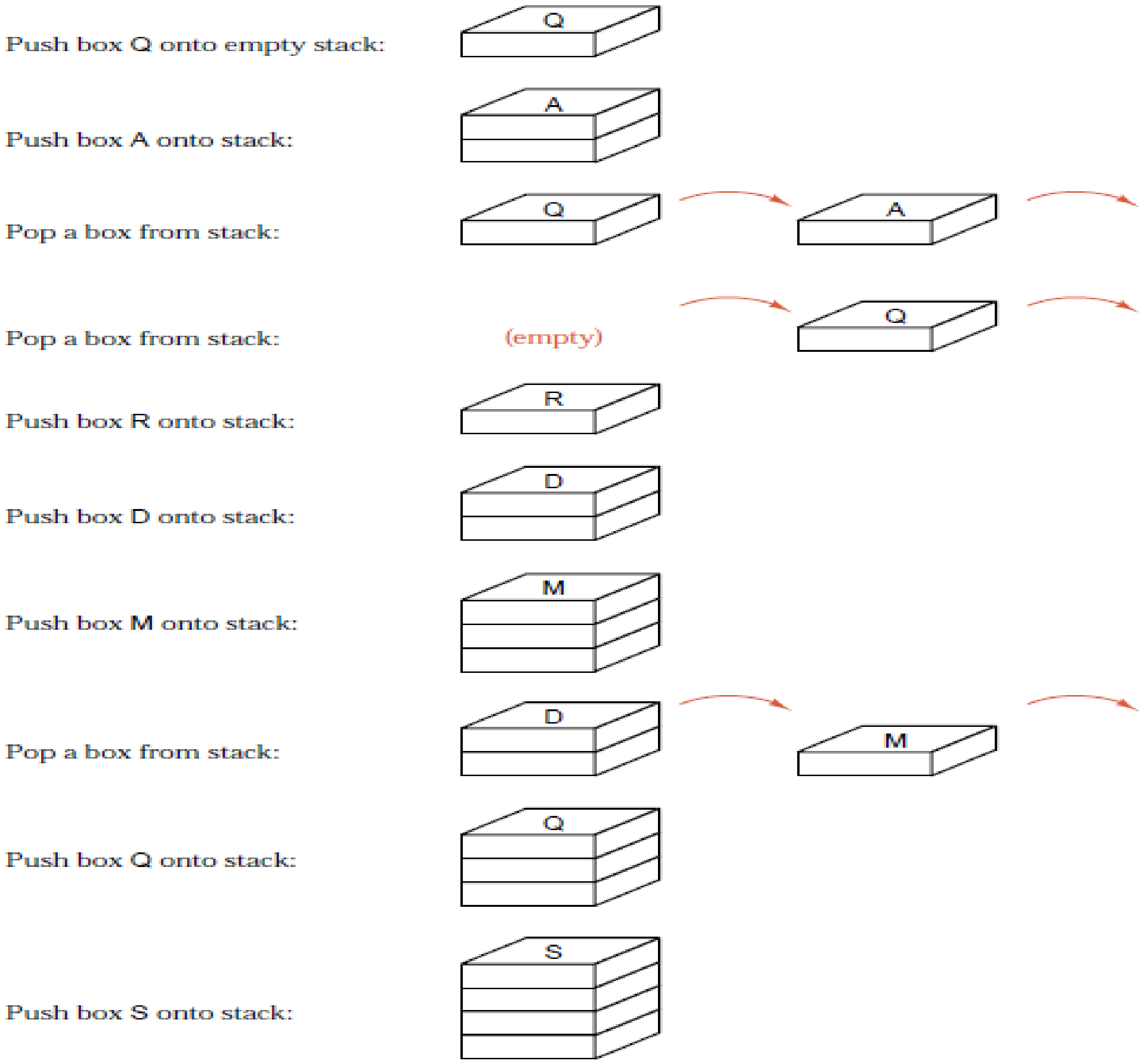

Figure 2.2. Pushing and popping a stack

Basic Operations

- Pushing an element at the top of the stack with overflow check.
- Pop an element from the top of the stack with underflow check.
- Print all elements currently in the stack.

Push Function

```
void push (int a[], int x)
{
++top;
If (top==MAX)
{
printf(“Stack Full”);
exit(1);
}
a[top]=x;
}
```

Pop Function

```
int pop(int a[])
{
int x;
if (top==-1)
{
printf(“Stack Empty”);
exit(1);
}
x=a[top];
top--;
return x;
}
```

Main Function

```
define MAX 10
int top=-1;
void main()
{
int a[MAX], n,x;
printf(“\n How many elements to push”);
scanf(“%d”,&n);
for(i=0;i<n;i++)
{
printf(“\n Enter element %d”, i+1);
scanf(“%d”,&x);
push(a,x);
}
for(i=0;i<n;i++)
{
x=pop(a);
printf(“%d”,x);
}
}
```

Assignments

- To implement the discussed program.
- To implement stack operations using switch case structure. On input 1, element will be pushed into the stack; on input 2 element will be popped out of the stack; on input 3 all stack elements will be printed . These operation are to be embedded in an infinite loop.