23) IMPLEMENTING QUEUES USING TWO STACKS

This is the standard sum of implementing the stacks.

CODE OF THE PROGRAM:

```
void StackQueue :: push(int x)
{
   if(s1.size()==0 && s2.size()==0){
      s1.push(x);
   }
   else if(s1.size()==0){
      s2.push(x);
   }
   else if(s2.size()==0){
      s1.push(x);
   }
}

//Function to pop an element from queue by using 2 stacks.
int StackQueue :: pop()
{
   if(s1.size()==0 && s2.size()==0){
      return -1;
   }
   else if(s1.size()==0){
      int temp;
      while(s2.size()!=0){
         temp=s2.top();
         s2.pop();
         if(s2.size()!=0){
            s1.push(temp);
         }
      }
      int t;
      while(s1.size()!=0){
         t=s1.top();
         s1.pop();
         s2.push(t);
      }
      return temp;
   }
   else if(s2.size()==0){
      int temp;
      while(s1.size()!=0){
         temp=s1.top();
         s1.pop();
         if(s1.size()!=0){
            s2.push(temp);
            }
        }
        int t;
        while(s2.size()!=0){
            t=s2.top();
            s2.pop();
            s1.push(t);
        }
        return temp;
    }
}
```