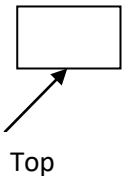
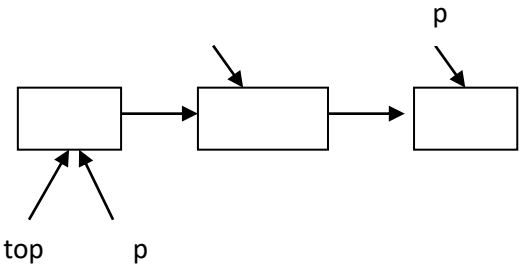
Erzeugen einer Liste durch das Einfügen der Knoten am Ende:

Schritt1: Man erzeugt den ersten Knoten der Liste

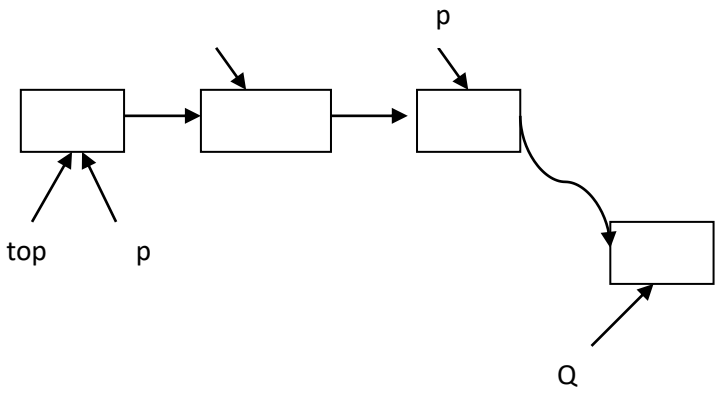
New(top);
Readln(Top^.info);
Top^.link:=nil;



Schritt2: Angenommen, die Liste hat eine bestimmte Anzahl von Knoten. Diese müßen von dem ersten bis zum letzten Knoten durchquert werden, damit am Ende der Liste eine neuer Knoten eingeführt wird.



P:=top;
While p^.link<>nil do p:=p^.link;
Schritt3: Wenn der sich Zeiger P auf dem letzten Knoten befindet, kann ein neuer Knoten, Q, am Ende der Liste eingefugt werden.



New(q);
Readln(q^.info);
P^.link:=q;
Q^.link:=nil;

Erzeuge eine Liste mit n Knoten, welche ganze Zahlen enthalten, indem die Knoten am Ende der Liste eingefugt werden.

```
Type pointer=^knoten;
     Knoten=record
              Info:integer;
              Link:pointer ;
     End;
Var p,q,top:pointer;
     I,n:integer;
Begin
     Write('Ánzahl der Knoten=');
     Readln(n);
     Top:=nil;
     New(top);
    Readln(Top^.info);
   Top^.link:=nil;
   For i:=1 to n do
      Begin
           P:=top;
           While p^.link<>nil do p:=p^.link;
           New(q);
           Readln(q^.info);
           P^.link:=q;
           Q^.link:=nil;
     End;
Drucken der Liste;
End.
```

Hausaufgabe:

Gegeben ist eine natürliche Zahl mit höchstens 9 Ziffern. Erzeuge und drucke mit Hilfe einer Liste das Spiegelbild dieser Zahl, wobei die Knoten:

1. Am Anfang der Liste eingefügt werden
2. Am Ende der Liste eingefügt werden(die Liste wird rekursiv gedruckt).

Entfernen des Knotens der die Information x enthält:(Hausaufgabe von letzter Stunde)

```
Procedure Entfernen;
Var x:integer;
Begin
    Write(‚x=‘);
    Readln(x);
    If top<>nil then {zuerst versichert man sich dass die Liste existiert, top<>nil}
        If top^.info=x then {wenn der erste Knoten die Inform. X enthalt, wird er
        Begin                          entfernt}
            P:=top;
            Top:=top^.link;
            Dispose(p);
        End
    Else
    Begin  {damit ein Knoten im Inneren der Liste entfernt wird, arbeitet man mit 2
        P:=top;  Pointer, um die Arbeit zu erleichtern}
        Q:=top^.link;
        While (q<>nil) and (q^.info<>x) do  {der erste Pointer der sich auf dem
        Begin                  Knoten mit der Information x befindet, ist Q}
            P:=p^.link;
            Q:=q^.link;
        End;
        If q=nil then write(‘Knoten mit der Information x existiert nicht‘)
            Else
            Begin
                P^.link:=q^.link;
                Dispose(q);
            End;
    End;
End;
```

Erklärung: um einen Knoten im Inneren der Liste zu entfernen, arbeitet man mit 2 Pointer, p und q. Am Anfang befinden sich diese Pointer auf den ersten 2 Knoten. Der Pointer der zuerst auf den Knoten mit der Information x (wenn dieser existiert) gelangt, ist q. Solange das Ende der Liste nicht erreicht wird bzw. der Knoten mit der information x nicht erreicht wurde, verlagert man beide Pointer(sowohl P als auch Q: P:=p^.link; Q:=q^.link;). Wenn q=nil (das Ende der Liste wurde erreicht), wird kein Knoten entfernt, da es keine Knoten mit der Information, x, gibt. Wenn q<>nil, dann wurde der Knoten mit der Information, x, gefunden und dieser wird entfernt.

Entfernen der gesamten Liste

```
Procedure entf;
Begin
        While top<>nil do
        Begin
                P:=top;
                Top:=top^.link;
                Dispose(p);
        End;
End;
```