

Liste, Stack

```
1  OpenConsole()
2  NewList Numbers.w()
3  AddElement(Numbers())
4  Numbers() = 3
5  AddElement(Numbers())
6  Numbers() = 5
7  AddElement(Numbers())
8  Numbers() = 7

9  LastElement(Numbers())
10 a = Numbers()
11 PrintN(Str(a))
12 DeleteElement(Numbers())

13 LastElement(Numbers())
14 a = Numbers()
15 PrintN(Str(a))
16 DeleteElement(Numbers())

17 LastElement(Numbers())
18 a = Numbers()
19 PrintN(Str(a))
20 DeleteElement(Numbers())

21 Input()
```

Stack

```
1  Structure Knoten
2    x.w
3    y.w
4    n.s
5  EndStructure

6  NewList Knotenliste.Knoten( )

7  AddElement(Knotenliste( ))
8    Knotenliste( )\x=100
9    Knotenliste( )\y=200
10   Knotenliste( )\n="K1"

11  AddElement(Knotenliste( ))
12    Knotenliste( )\x=200
13    Knotenliste( )\y=300
14    Knotenliste( )\n="K2"
15
16  LastElement(Knotenliste( ))
17  PrintN(Str(Knotenliste( )\x))
18  PrintN(Str(Knotenliste( )\y))
19  PrintN(Knotenliste( )\n)
20
21  DeleteElement(Knotenliste( ))
22
23  LastElement(Knotenliste( ))
24  PrintN(Str(Knotenliste( )\x))
25  PrintN(Str(Knotenliste( )\y))
26  PrintN(Knotenliste( )\n)
```

Stack mit Pointern

```
1  OpenConsole()

2  Structure Knoten
3    a.i
4    b.i
5  EndStructure

6  Global Dim Knotenliste.Knoten(20)
7  Global *Pointer.Knoten = @Knotenliste()
8  Global *Pointer.Knoten = @Anfang

9  Procedure push(*k.Knoten)
10   *Pointer + SizeOf(Knoten)
11   *Pointer\a = *k\a
12   *Pointer\b = *k\b
13 EndProcedure

14 Procedure pop()
15   *p.Knoten = *Pointer
16   *Pointer - SizeOf(Knoten)
17   ProcedureReturn *p
18 EndProcedure

19 Eingabe.Knoten

20 Eingabe\a = 5
21 Eingabe\b = 7
22 push(@Eingabe)

23 Eingabe\a = 2
24 Eingabe\b = 9
25 push(@Eingabe)
26 PrintN(" ")

27 *adresse.Knoten=pop()
28 PrintN(Str(*adresse\a))
29 PrintN(Str(*adresse\b))

30 *adresse=pop()
31 PrintN(Str(*adresse\a))
32 PrintN(Str(*adresse\b))

33 Input( )
```