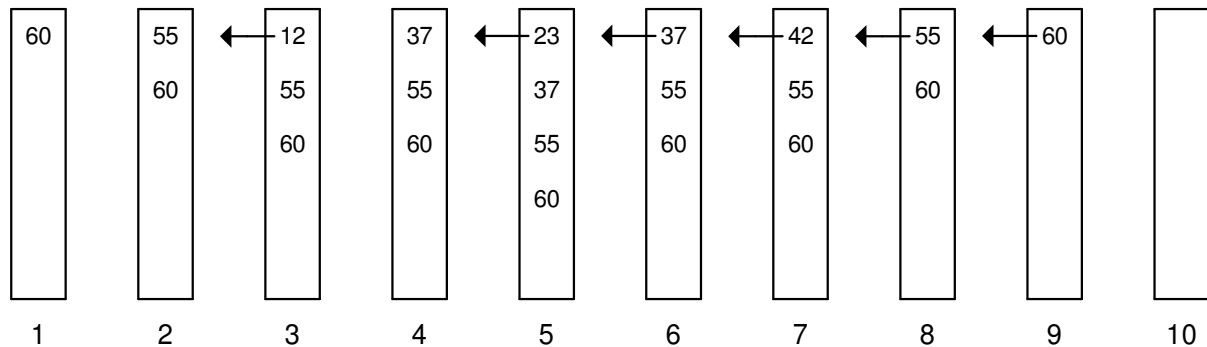


## Lösungsskizze

### Aufgabe 1

a)

Phasen der Belegung des Transportbandes C



Die Pakete „60“, „55“ und „12“ werden in den ersten drei Phasen auf C gelagert, ebenso in Phase 5 das Paket „23“. In Phase 4 wird das Paket „12“ von Band C auf das Band B gelegt, bevor das Paket „37“ von Band A auf C gebracht wird. Entsprechend werden in den Phasen 5 und 6 die Pakete „23“ und „37“ von C nach B gebracht, bevor „42“ von A nach C gelegt werden kann. Am Ende wird das Band C noch geleert.

b)

Nicht sortieren lässt sich z. B. die Folge 55, 60, 12, 37, 23, 42: Das Paket mit der Nummer 55 wird auf das Transportband C gelegt, im nächsten Schritt mit dem Paket mit der Nummer 60 auf Band A verglichen und auf das linke Transportband B gelegt. Damit liegt das Paket mit der Nummer 55 auf dem Band B an einer falschen Position.

c)

Erläuterung des Sortiervorgangs:

- 27 kommt als erste Zahl direkt in den rechten Stapel
- da 42 größer als 27 ist, wird 27 zunächst im linken Stapel zwischengespeichert, 42 in den rechten Stapel gelegt und 27 zurückgebracht
- 18 kann direkt auf den rechten Stapel gelegt werden, da 18 kleiner als die oberste Zahl im Stapel ist
- da 32 größer als 18 und 27 ist, werden diese beiden Zahlen nacheinander in den linken Stapel ausgelagert, die 32 auf den rechten Stapel gelegt und die 18 und 27 zurückgebracht

Der linke Stapel dient nur zur Zwischenlagerung, der rechte Stapel enthält die schon bearbeiteten Zahlen in sortierter Reihenfolge.

d)

Implementation des Sortierverfahrens

Implementierung in Java

```

void sortieren() {
    Stapel rechterStapel = new Stapel();
    Stapel linkerStapel = new Stapel();
    for (int i = 0; i < zahlen.length; i++) {
        while (!rechterStapel.isEmpty() && (rechterStapel.top() < zahlen[i])) {
            linkerStapel.push(rechterStapel.pop());
        }
        rechterStapel.push(zahlen[i]);
    }
}

```

```
    while (!linkerStapel.isEmpty()) {  
        rechterStapel.push(linkerStapel.pop());  
    }  
}  
}
```

### Implementierung in Delphi/Pascal

```
procedure sortieren;  
var  
    rechterStapel: tStapel;  
    linkerStapel: tStapel;  
    i: Integer;  
begin  
    rechterStapel := tStapel.init;  
    linkerStapel := tStapel.init;  
    for i := 1 to length(zahlen) do  
    begin  
        while (not rechterStapel.isEmpty) and (rechterStapel.top < zahlen[i]) do  
            linkerStapel.push(rechterStapel.pop);  
        rechterStapel.push(zahlen[i]);  
        while not linkerStapel.isEmpty do  
            rechterStapel.push(linkerStapel.pop);  
    end;  
end;
```