Övningar med Arrayer

(Skapa en klass för varje uppgift)

- **Uppgift 1**
  Skriv ett program `Reverse.java` som läser ett godtyckligt antal positiva heltal från tangentbordet och sedan skriver ut dem baklänges. Inmatningen sker fram tills det att användaren matar in ett negativt tal. En körning kan se ut enligt följande:

```
Mata in positiva heltal. Avsluta med ett negativt.
tal 1: 5
tal 2: 10
tal 3: 15
tal 4: 20
tal 5: -7

Antal positiva: 4
Baklänges: 20, 15, 10, 5
```

  Notera: Man skall inte behöva ange hur många tal man ämnar mata in.

- **Uppgift 2**
  Skriv en klass `Arrays.java` som innehåller följande **statiska** metoder:
  - Metoden `int sum(int[] arr)` som adderar ihop alla element i arrayen `arr` och returnerar summan.
  - Metoden `String toString(int[] arr)` som bygger upp en sträng som när den skrivs ut ger en snygg utskrift av arrayens innehåll. Den skall kunna användas på följande sätt med gott resultat

```
int n = {3,4,5,6,7};
String str = Arrays.toString(n);
System.out.println("n = " + str);
```

  Metoden `int[] addN(int[] arr, int n)` som bygger upp, och returnerar, en ny array där man adderat talet `n` till alla element i arrayen `arr`. Arrayen `arr` skall lämnas opåverkad.

  Metoden `int[] reverse(int[] arr)` som bygger upp, och returnerar, en ny array där alla element i arrayen `arr` finns i omvänd ordning. Arrayen `arr` skall lämnas opåverkad.

  Metoden `boolean hasN(int[] arr, int n)` som returnerar `true` om `n` finns i arrayen `arr`, annars `false`.

  Metoden `void replaceAll(int[] arr, int old, int nw)` som byter ut alla förekomster av `old` mot `nw` i `arr`.

  Metoden `int[] sort(int[] arr)` som returnerar en ny sorterad array (minsta först) med samma mängd av heltal som `arr`. Arrayen `arr` skall lämnas opåverkad.

- Metoden `boolean hasSubsequence(int[] arr, int[] sub)` som returnerar `true` om subsekvensen `sub` finns i arrayen `arr`, annars `false`. I fallet `hasSubsequence({1,2,3,4,5}, {3,4,5})` skall den alltså returnera `true` eftersom `{3,4,5}` finns som sista del i `{1,2,3,4,5}`.

Skriv också ett demonstrationsprogram `ArraysMain.java` som visar hur de olika metoderna fungerar.

- **Uppgift 3**
  Skriv ett program `Histogram.java` som läser ett godtyckligt antal heltal från en fil och sedan ritar upp ett histogram (stapeldiagram) för de av talen som är mellan 1 och 100. Notera: Talen i filen behöver ej vara inom intervallet [1,100]. En körning kan se ut enligt följande:

```
Läser heltal från filen: C:\Temp\heltal.dat
Antal i intervallet [1,100]: 46
Övriga: 16
Histogram
  1  - 10  | ******
  11 - 20  | ****
  21 - 30  | **
  31 - 40  | ***
  41 - 50  | *******
  51 - 60  | ****
  61 - 70  | ***
  71 - 80  | *********
  81 - 90  | *****
  91 - 100 | ***
```

Uppgift 4

Skriv en klass `Kort` som representerar ett kort i en vanlig kortlek med 52 olika kort. Varje kort har en *färg* (4 möjliga) och en *valör* (13 möjliga). Skriv sedan en klass `Kortlek` som från början innehåller 52 olika objekt av klassen `Kort`. Klassen `Kortlek` skall ha metoder för att blanda kortleken, dela ut ett kort och upplysa om hur många kort som finns kvar. Notera, det skall bara gå att blanda kortleken när den har 52 kort.

Skriv också ett program `SpelaKortMain` som skapar en kortlek, delar ut ett par kort, och sedan visar hur många kort det finns kvar och vilka kort som delats ut.