

## Instuderingsfrågor, del B

### Uppgift 1.

Vad skrivs ut i nedanstående programsekvenser?

- a) `int minutes = 10;`  
`while ( minutes <= 60) {`  
     `System.out.println( minutes + " minutes has passed.");`  
     `minutes = minutes + 20;`  
`}`
- b) `int i = 1;`  
`int formula = 0;`  
`while (i < 10 && formula < 25) {`  
     `formula = formula + i*i;`  
     `i = i + 1;`  
`}`  
`System.out.println(formula);`
- c) `int a = 1, antal = 3;`  
`while (a < 6) {`  
     `System.out.println("Resultat: " + a % antal);`  
     `a = a + 1;`  
`}`
- d) `boolean cont = true;`  
`int a = 1, b = 2;`  
`while (cont) {`  
     `if (a > 3) {`  
         `b = b + 2;`  
         `if ((a >= 6) || (b >= 8))`  
             `cont = false;`  
     `}`  
     `System.out.println("a = " + a + " b = " + b);`  
     `a = a + 1;`  
`}`
- e) `final int LIMIT = 16;`  
`int count = 1;`  
`int sum = 0;`  
`int nextVal = 2;`  
`while (sum < LIMIT) {`  
     `sum = sum + nextVal;`  
     `nextVal = nextVal + 2;`  
     `count = count + 1;`  
`}`  
`System.out.println("Sum is " + sum + "\nCount is " + count);`

### Uppgift 2.

Använd en **while**-sats

- a) för att skriva ut serien 0.25, 0.5, 0.75, . . . , 4.75, 5.0
- b) för att skriva ut serien 1 3 6 10 15 21 28 36
- c) som löper så länge som den boolska variabeln `finished` är **false**
- d) så länge som variabeln `counter` är större än konstanten `LOWER` och mindre än konstanten `UPPER`

### Uppgift 3.

Betrakta nedanstående programsekvens:

```
int i = 11;
double sum = 0;
while (i > 0) {
    i = i - 1;
    sum = sum + 1.0 / i;
}
```

Avsikten är att beräkna serien  $1/10 + 1/9 + \dots + 1/2 + 1/1$ . Något är dock gale och det uppkommer ett exekveringsfel i stil med

Exception in thread "main" java.lang.ArithmeticException: / by zero

Förklara vad som är fel? Rätta felet!

### Uppgift 4.

Vilka av nedanstående **while**-satser resulterar i en evighetsloop?

- a) 

```
int i = 1;
while (i != 25)
    i = i + 2;
```
- b) 

```
double x = 1.0;
while (x != 0.1)
    x = x - 0.1;
```
- c) 

```
int x = 0;
while (x != 55)
    x = x + 2 ;
```
- d) 

```
int i = 1, sum = 0;
while(i < 10 || sum != 15) {
    sum = sum + i;
    i = i + 1;
}
```
- e) 

```
int k = 1;
while (k != 10); {
    System.out.println(k);
    k = k + 1;
}
```

### Uppgift 5.

Vad blir utskrifterna av nedanstående programsegment?

- a) 

```
for (int k = 1; k <= 5; k = k + 1)
    System.out.println(k % 4);
```
- b) 

```
for (int k = 1; k <= 20; k = k + )
    if ((k % 4) == 0)
        System.out.println(k);
```
- c) 

```
for (int k = -8; k <= 10; k = k + 4)
    System.out.println(k);
```
- d) 

```
for (int k = 1; k <= 30; k = k * 2)
    System.out.println(k);
```
- e) 

```
for (int k = 5; k < 15; k = k + 2)
    System.out.println(k);
```
- f) 

```
for(int i = 15; i > 5; i = i - 3) {
    System.out.println(i);
}
```

### Uppgift 6.

**for**-satsen är egentligen en generell villkorsloop. Trots detta bör man enbart använda **for**-satsen som en räkneloop. Varför?

### Uppgift 7.

Nedanstående programsegment är ett exempel på missbruk av **for**-satsen. Varför? Hur borde koden se ut?

```
for (int i = 1; i <= 10; i = i + 1) {  
    System.out.println(i);  
    i = i+1;  
}
```

Tips: Vad skrivs ut?

### Uppgift 8.

Programsekvenserna i samtliga deluppgifter nedan resulterar i ett kompileringsfel! Varför? Rätta till koden!

- ```
for (int k = 1, k <= 10, k = k + 1)  
    System.out.println(k);
```
- ```
int i;  
for (int i = 1; i <= 10; i = i + 1) {  
    //här kommer en eller flera programsatser  
}
```
- ```
for (int j = 1; j <= 10; i = j + 1); {  
    System.out.println(j);  
    //här kommer eventuellt flera programsatser  
}
```
- ```
int antalVarv = 0;  
for (int i = 1; i <=10; i = i + 1)  
    for (int i = 1; i <=10; i = i + 1)  
        antalVarv = antalVarv + 1;
```

### Uppgift 9.

Vad blir utskrifterna av nedanstående programsegment?

- ```
int n = 3;  
int m = 6;  
for (int i = 1; i <= n; i = i + 1) {  
    for (int j = 1; j <= m; j = j + 1)  
        System.out.print('*');  
    System.out.println();  
}
```
- ```
int n = 3;  
int m = 6;  
for (int i = 1; i <= n; i = i + 1) {  
    for (int j = 1; j <= i; j = j + 1)  
        System.out.print('*');  
    System.out.println();  
}
```
- ```
for (int i = 0; i <= 3; i = i + 1) {  
    for (int j = 5; j <= 7; j = j + 1)  
        System.out.print(j);  
    System.out.println(i);  
}
```

### Uppgift 10.

Vad blir utskrifterna av nedanstående programsegment?

- a) 

```
for (int j = 1; j <= 5; j = j + 1) {
    for (int k = 1; k <= 5; k = k + 1) {
        System.out.print(j + " ");
    }
    System.out.println();
}
```
- b) 

```
for (int j = 1; j <= 5; j = j + 1) {
    for (int k = 1; k <= j; k = k + 1) {
        System.out.print(j + " ");
    }
    System.out.println();
}
```
- c) 

```
for (int j = 1; j <= 5; j = j + 1) {
    for (int k = j; k <= 5; k = k + 1) {
        System.out.print(j + " ");
    }
    System.out.println();
}
```

### Uppgift 11.

Hur många gånger kommer HURRA! att skrivas ut i nedanstående programavsnitt?

- a) 

```
for (int i = 1; i <= 3; i = i + 1)
    for (int j = 1; j <= 3; j = j + 1)
        for (int k = 1; k <= 3; k = k + 1)
            System.out.println("HURRA!");
```
- b) 

```
for(int i = 1; i <= 3; i = i + 1)
    for (int j = 1; j <= 3; j = j + 1)
        for (int k = 1; k <= j; k = k + 1)
            System.out.println("HURRA!");
```

### Uppgift 12.

Använd en **for**-sats för att

- a) skriva ut alla udda värden i intervallet 1-15  
b) skriva ut serien 0.25, 0.5, 0.75, . . . , 4.75, 5.  
c) skriva ut serien 10, 8, 6, 4, 2, 0, -2, -4, -6, -8, -10  
d) skriva ut serien 0, 1, 4, 9, 16, . . . , 81, 100

### Uppgift 13.

I vilka av nedanstående problem är det lämpligast att använda en **for**-sats och i vilka problem är det lämpligast att använda en **while**-sats? Motivera!

- a) skapa en iteration som löper från 15 till -15 med steget 1.  
b) skapa en iteration som håller på tills värdet av variabel **x** till absolutbelopp är mindre än 0.0005.  
c) skapa en iteration som löper från 2 till 100 med steget 2.  
d) skapa en iteration som löper så länge som variabeln **position** är mindre än 100 och den boolska broken är **false**

### Uppgift 14.

Vad skrivs ut när nedanstående kodsegment exekveras?

```
int i = 4;
do {
    System.out.println(i + " + i);
    i = i - 1;
} while(i > 4);
```

### Uppgift 15.

Betrakta nedanstående programsegment:

```
int i = 0, sum = 1;
do {
    boolean more;
    i = i + 1;
    sum = sum * i;
    if (i > 6)
        more = false;
    else
        more = true;
} while(more);
System.out.println(sum);
```

Vid kompilering får följande felutskrift:

```
cannot find symbol
symbol : variable more
```

Förklara vad som är fel och åtgärda felet.

### Uppgift 16.

Skriv en programsekvens som läser in ett heltal och upprepar inläsningen tills ett tal i intervallet 1 till 7 erhålls. Använd dialogrutor för kommunikationen med användaren.

### Uppgift 17.

SI-systemet, dvs det internationella måttenhetssystemet, antogs 1960. De traditionsbundna engelsmännen har förvisso officiellt övergått till SI-systemet, men i praktiken använder de dock fortfarande sina gamla brittiska måttenheter från imperietiden.

Skriv ett program som läser in en längdenhet angivna i meter och skriver ut motsvarande längd angiven i antal yards, feet och inches. Programmet skall utformas på så sätt att inläsningen upprepas till användaren avbryter exekveringen. Det gäller att

```
1 yard = 0,9144 meter
1 yard = 3 feet
1 foot = 12 inches
```

Detta innebär alltså att noggrannheten så erhålls blir  $0.9144 / (3 \cdot 12) = 0.0254$  centimeter.

Vill man ha högre noggrannhet måste man förutom yard, feet och inches även ange antalet line, point och twip, där

```
1 inch = 12 lines
1 lines = 6 points
1 point = 20 twips
```