

1)

3 representerer en «identifiser». Kalles en literal constant.

'3' Single character literal, refererer til det som representerer tallet 3.

«3» String literal, representerer tallet 3 i en string.

2)

```
if (i < 10) {  
    printf("foo");  
} else {  
    printf("bar");  
}
```

Denne koden vil bare printe «foo» hvis i har en verdi over 10. Hvis i er 11, printer den «foo», er i 9, printer den «bar».

3)

```
if (i > 10) {  
    printf("foo");  
} else {  
    printf("bar");  
}
```

Denne koden vil printe «bar» gitt at «i» er 0xFFFFFFFF. Da regnes det også med at dette er 2's complement, og 0xFFFFFFFF vil returnere «-1», som med denne koden vil printe «bar».

4)

```
for (i = 0; i < 5; i = i+1) {  
    printf("%d\n", i);  
}  
for (i = 0; i < 5; i++) {  
    printf("%d\n", i);  
}  
for (i = 5; i >= 0; i = i-1) {  
    printf("%d\n", i);  
}  
for (i = 0; i <= 5; i *= 2) {  
    printf("%d\n", i);  
}
```

Den første loopen printer tallene [0,1,2,3,4]. Den printer «i» så lenge «i» er mindre enn 5, og adderer 1 til «i» for hver gang loopen kjøres.

Den andre loopen gjør nøyaktig det samme som den første, da «i++» og «i+1» begge adderer 1 til variablen. Tallene [0,1,2,3,4] vil bli printet av denne også.

Den tredje loopen printer tallene [5,4,3,2,1,0], da «i» har en startverdi på 5, og subtraherer 1 for hver gang loopen gjennomføres til «i» = 0.

Den siste loopen vil printe «0» evig mange ganger. Den sier at «i» skal printes så lenge den er mindre eller lik tallet 5, og for hver gang loopen gjennomkjøres, skal den gange seg selv med tallet 2. Med tanke på at «i» har en startverdi på 0, vil « $0 * 2 = 0$ » alltid gjelde, og loopen vil fortsette i all evighet.