

4a Hemligt tal

I början av varje spelomgång ska ett hemligt tal slumpas fram. Detta ska lagras i en vektor. Vektorns typ ska deklarerars i header-filen. Först skapar man en konstant, tex `TAL_STORLEK`, som anger storleken på vektorn (4 siffror i grundversionen). Sedan definierar man en egen typ som man ger ett eget namn, tex `tal`, med raden
`typedef int tal[TAL_STORLEK]`
Det här betyder att man har en ny typ som heter `tal` och som kan användas överallt i programmet. Typen är i det här fallet en vektor med fyra positioner. Ändrar man på `TAL_STORLEK` ändras även storleken på vektorn.

För att skapa det hemliga talet måste man nu skapa en variabel som är av typen `tal` och fylla den med siffror. Skriv denna del som ett eget program. Kompilera och testa programmet några gånger. Om man vill vara säker på att alla siffror är olika ges här lite tips om hur man kan göra.

Tips 1)

Skapa en array med 4 positioner som ska lagra det slumpade talet.

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| | | | |

Skapa en array med 10 positioner. Fyll den med siffrorna 0, 1, 2 osv.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Slumpa fram ett tal mellan 0-9, tex 4. Tag talet på position 4 i den stora arrayen och flytta till första positionen i den lilla.

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 4 | | | |

Flytta nu alla tal i den stora arrayen efter position 4 ett steg åt vänster och minska storleken på arrayen.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 1 | 2 | 3 | 5 | 6 | 7 | 8 | 9 |

Slumpa fram ett nytt tal. Skulle det bli 6 tar vi talet på den positionen, vilket nu är 7, och lägger i den lilla arrayen.

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 4 | 7 | | |

Flytta fram och minska storleken i den stora.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 1 | 2 | 3 | 5 | 6 | 8 | 9 |

På det här sättet kan vi aldrig få några dubletter i det hemliga talet.

Tips 2)

Skapa en array med 4 positioner som ska lagra det slumpade talet.

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| | | | |

Skapa en array med 10 positioner som kan lagra booleska värden. Fyll den med true.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| T | T | T | T | T | T | T | T | T | T |

Slumpa fram ett tal mellan 0-9, tex 3. Skriv 3 på första positionen i den lilla arrayen.

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 3 | | | |

Sätt nu false på position 3 i den stora arrayen för att visa att det talet ”redan är taget”.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| T | T | T | F | T | T | T | T | T | T |

Slumpa fram ett nytt tal tex 5. Om talet inte ”redan är taget” skriv det i den lilla arrayen. Om talet redan finns slumpa fram ett nytt.

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 3 | 5 | | |

Fördelen med tips 1 är att man inte behöver slumpa fram nya tal om det redan är taget. Nackdelen med tips 1 är att man måste flytta siffror i den stora arrayen varje omgång. Det går givetvis också att kontrollera det hemliga talet för att se om man redan har det tal som man slumpat fram.