

SKRIV NAVN - PÅ ALLE SIDER:

PROCESSING:

1. Skriv processing-kode, der opretter en variabel "navn", der indeholder dit navn.
2. Skriv processing-kode, der opretter et array med 100.000 forskellige heltal.

JAVASCRIPT OG HTML:

3. Byg en knap med HTML og JavaScript, der ved tryk giver en visuel effekt, f.eks. en tekst, der ændrer sig eller lignende.

OOP PRINCIPPER:

4. **2 point**

Skriv kode til følgende klasser: "Fugle", "And", "Måge" og "Hjerne", der anvender nedarvning og komposition til at "forbinde" klasserne på en hensigtsmæssig måde.

ALGORITMER:

5. **2 point**

Bestem worst-case og best-case kørselstids-funktionen af array-sammenligninger for denne (ikke så smarte) algoritme:

```
boolean toEns = false;
for ( int i = 0 ; i < list.length ; i++ ) {
    for( int k = 0 ; k < list.length ; k++ ) {
        if( k != i && list[i] == list[k] ) {
            toEns = true;
            break;
        }
    }
}
```

6. **2 point**

Kan du lave en hurtigere algoritme, end den ovenfor, der finder to ens tal i et array?

7. **2 point**

Anvend binær søgning til at søge efter tallet 4 i denne liste af tal. Hvor mange forsøg koster det før tallet er fundet? Og hvilke tal finder algoritmen i sin søgning?

[1, 2, 4, 5, 10, 23, 33, 45, 90, 91, 100, 120, 130]

8. **2 point**

Tegn et klassisk binært søgetræ ud fra følgende sekvens af tal:

[10, 6, 16, 3, 9, 13, 19]

9. Lav sekvens af samme tal som i opgave 8, der giver det "dårligst" mulige søgetræ.

10. Vurder hvor hurtigt man kan søge i træerne fra opgave 8 og 9

I alt 15 point - opgaverne 4,5,6,7 og 8 giver 2 point