

# Eksamen

## Emne 5 Algoritmiske metoder



TID: Start: 2025-12-1, Innlevering: 2025-12-5

Hjelpemidler: Alle

Innlevering:

- Zip-fil: eksamen\_emne5\_<navn>.zip
- README.md (kort oversikt + hvordan kjøre kode + eventuelle KI-prompter referert til)
- Videofil
- Legg ved Mappe 1 eller 2 ved din innlevering! (**Du kan forbedre koden din**)

### Oppgave 1

Studenten skal velge **Mappe 1 eller Mappe 2** som grunnlag for videoen. Selv om mappene allerede er levert tidligere i semesteret, skal studenten **laste opp valgt mappe på nytt i Wiseflow** som en del av denne eksamenen.

**Viktige presiseringer:**

- **Koden kan gjerne forbedres** før ny innlevering.
- Det er **ikke nødvendig å gjøre store endringer** – kun justeringer som gjør koden tydeligere, ryddigere eller mer korrekt.
- **Den mappen som leveres her vil være grunnlaget for vurderingen av videoen**, og videoen skal forklare:
  1. Mappens struktur og hensikt

2. Løsningen i Oppgave 2
3. De viktigste konseptene som er brukt i koden

## Oppgave 2 (30%)

Du skal velge en minioppgave basert på hvilken mappe de leverte. **Minioppgave A – Lineært søk (Mappe 1)**

**Implementer metoden:**

```
int LinearSearch(int[] array, int target)
```

**Testdata:**

- Array: [4, 8, 2, 9, 1]
- Target: 9 → **Forventet output: 3 Svar i presentasjon også på:**

1. Hva er tidskompleksiteten for lineært søk?
2. Hva er det verste tilfellet (worst case)?
3. Når er lineært søk “godt nok” som løsning?

## Minioppgave B – Bubble Sort

**(Mappe 2) Implementer metoden:**

```
void BubbleSort(int[] array)
```

**Sorter følgende liste:**

- [5, 1, 4, 2, 8]

**Svar i presentasjon også på:**

1. Hvorfor Bubble Sort regnes som ineffektiv
2. Sammenligning med Quick Sort
3. Hvorfor optimalisering av Bubble Sort kan hjelpe

## Oppgave 3 (70%) - Presentasjon og Refleksjon

Målet med denne oppgaven er å reflektere over løsningen av Oppgave 1,2 og formidle denne refleksjonen på en klar og sammenhengende måte. **Instruksjoner:** A.

Lage en Video:

- Lag en video hvor du forklarer hvordan du tenkte når du løste oppgaven og hvordan du har implementert løsningen.
- Videoen bør være mellom 8-10 minutter lang. B. Inkluder følgende i videoen: o Kodegjennomgang:
- Vis og forklar de viktigste delene av koden din. Du trenger ikke å gå gjennom hver eneste linje, men konsentrer deg om de mest sentrale delene i forhold til oppgavens funksjonalitet.
- o Problemløsning:

- Diskuter eventuelle utfordringer du støtte på mens du løste oppgaven og hvordan du overkom dem.
- Demonstrasjon:
  - Kjør programmet og demonstrer hvordan det fungerer. Fokuser på hvordan programmet reagerer på forskjellige brukerinnndata.

### C. Vurderingskriterier:

- Klarhet: Er din forklaring klar og lettfattelig?
- Fullstendighet: Har du dekket alle aspekter av oppgaven?
- Teknisk dybde: Går du i dybden på hvordan koden fungerer, ikke bare hva den gjør? o  
Presentasjon: Er videoen organisert på en måte som er lett å følge?

#### Tips:

- Planlegg hva du skal si før du begynner å filme for å sikre en strømlinjeformet og konsis presentasjon.
- Du kan benytte skjermopptaksverktøy som OBS, QuickTime (for Mac), eller andre tilgjengelige programvare for å lage videoen.
- Vær deg selv når du presenterer. Mindre feil her og der er helt akseptable, det viktigste er at du klart formidler din forståelse av problemet og løsningen

## Vedlegg

### Vurderingskriterier, Hva bør du tenke på før innlevering

#### Kjørbarhet

- Scriptene fungerer uten feil
- Delvis funksjonelle (inneholder mindre feil som påvirker output)
- Ikke kjørbare
  1. Syntax error
  2. Koden er kommentert ut

#### Fullførelse

- Alle oppgaver er fullført
- Oppgavene er delvis fullført
- Ingen av oppgavene er fullført

#### Korrekthet

- Output er i tråd med oppgaveteksten
- Mindre avvik fra oppgaveteksten
- Betydelige avvik (oppfattes som et annet arbeid)

#### Kodekvalitet (Lesbarhet og struktur)

##### Kodekvalitet (Lesbarhet og struktur) Høy kvalitet

- God kodeorganisering og ryddig struktur
- Konsistent og korrekt formattering
- Gode, beskrivende variabelnavn

- Relevante og presise kommentarer
- Koden er enkel å lese og forstå **Middels kvalitet**
- Delvis ustrukturert kode
- Manglende eller svake kommentarer
- Variabelnavn er lite presise eller blander språk
- Noe vanskelig å lese, men logikken kan følges **Dårlig kvalitet**
- Uorganisert og uoversiktlig kode
- Uklare eller misvisende variabelnavn
- Manglende kommentarer
- Koden er vanskelig å lese og forstå

### Video

- **Klarhet:** Er forklaringen klar og enkel å forstå?
- **Fullstendighet:** Dekker videoen alle aspekter av oppgaven?
- **Teknisk dybde:** Gir den en forståelse av hvordan koden fungerer, ikke bare dens funksjon?
- **Presentasjon:** Er videoen godt organisert og lett følgelig?

## Bruk av KI (Kunstig intelligens)

Bruk av kunstig intelligens (KI) er **tillatt**, men kun under forutsetning av korrekt og fullstendig dokumentasjon.

### Krav til dokumentasjon

- Du må levere alle forespørsler (prompter) og svar som er brukt i besvarelsen.
- Dokumentasjonen skal være en del av innleveringen — enten i README.md eller i en egen fil.
- Du skal kun inkludere forespørsler og svar som helt eller delvis er kopiert inn i, eller som direkte har
- påvirket, løsningen din.
- **Konseptuelle eller generelle samtaler** med KI som ikke har påvirket innleveringen, trenger ikke dokumenteres.
- Alternativt kan du oppgi en eksplisitt erklæring om at du ikke har brukt KI i den aktuelle innleveringen.

### Viktig

- **Manglende dokumentasjon av KI-bruk regnes som brudd på eksamensreglene.**
- **Ved mistanke om udokumentert KI-bruk kan kandidaten bli innkalt til en muntlig kontroll** for å verifisere egen forståelse av innleveringen.
- Dersom kandidaten ikke kan gjøre rede for arbeidet sitt, eller dersom KI-bruk viser seg å være udokumentert, kan det bli behandlet som juks i tråd med skolens retningslinjer