OPPGAVER
Del 1

Grunnleggende begreper

• Algoritmer • Hverdagsalgoritme • Halveringsmetoden • Følg en pseudokode til Algoritme til program	Å gå fra en programmeringskode for mennesker, sette opp stegene(Algoritmen) og over til en programmeringskode er en viktig del av forståelsen. Her gjelder en ting- øvelse gjør mester. Gode Pseudokoder og Algoritmer er halve programmeringen.

Pseudokode, Algoritmisk oppsett for passord

1. importerer biblioteket time for å kunne legge inn en forsinkelse ved for mange feil forsøk.
2. personlig informasjon som navn, adresse, mail og mobil, skal skrives til skjerm dersom bruker taster rett passord.
3. passordet lagres i variabel kaldt key og er en string.
4. antall tillate forsøk settes til 3.
5. bruk en while-løkke så lenge antall forsøk som er igjen er mer enn 0.
6. brukeren blir bedt om å taste inn passord. Passordet lagres i variabelen inn. (Dette er en tekst-string, bruk derfor str).
7. dersom passord fra bruker er likt som key- > informasjonen skrives ut til skjerm. Dersom den ikke er lik så reduseres antall forsøk igjen med 1, og det skrives ut beskjed om feil passord og antall forsøk igjen.
8. dersom det ikke er flere forsøk igjen så må brukeren vente i 10 sekunder før det er mulig å prøve igjen.

Lag Python programmet og testkjør dette for å se at funksjonen er riktig lagt opp.

Pseudokode, Algoritmisk oppsett for gjett på et tall

1. importerer biblioteket random for å kunne bruke tilfeldig valgt tall.
2. print informasjon om programmet til skjerm og hvordan bruker skal forholde seg til følgende:
   Dette er et program som lagrer et tall mellom 0 og det tallet du velger.
   Så skal du gjette på hvilket tall programmet har lagret(random/tilfeldig).
   Programmet vil fortelle om tallet er for høyt, for lavt eller riktig.

3. Skriv inn øvre grense for tallet som skal gjettes, lagre dette i en variabel som heter topp, hent dette fra bruker, input.
4. Skriv ut fra verdi til topp verdi i gjettingen
5. legg inn variablene gjett, tall og gjettinger og skriv ut
6. bruk en while-løkke for å definere sant, ikke sant(True, False).
7. Hvis bruker gjetter for lavt, print tallet du skrev er for lavt, og samme hvis for høyt.(if og elif)
8. else, og print da , du gjettet riktig !
9. til slutt skriv ut antall gjettinger i en funksjonsutskrift (f" x {gjettinger}xx.")

Lag Python programmet og testkjør dette for å se at funksjonen er riktig lagt opp.