Sensorveiledning IN1000 vår 2025

Generelt: det er fullt lovlig å trekke kandidateropp/ned basert på helhetsinntrykk. Poengene er ikke en «must-follow», og hvis noen viser forståelse, men dessverre svarer feil så kan det trekke opp. Tilsvarende kan «flaks» med riktige svar uten noen forståelse trekke ned. Sensor har mulighet til å justere.

Oppgave 1 er autorettet.

Det er ikke nødvendig å kjøre programmene. Vi er ikke interessert i om de har skrevet helt 100% syntaktisk korrekt, men heller om de viser forståelse. Ingen trekk for f.eks. «print» med stor P, manglende kolon, små skrivefeil o.l. med mindre det er gjennomgående feil.

Vedlagte løsningsforslag er *forslag* og ikke fasit. Det finnes mange like gode/bedre måter å løse oppgavene på.

Oppgave 1

Automatisk rettet, se besvarelsen i Inspera for riktige svar.

Oppgave 2

A

De må ha fått til å skrive en funksjon som tar inn en parameter som beskrevet i oppgaven. De bør ha fått til å benytte en while-løkke som tar høyde for valg utenfor intervallet, se spesielt etter casting til int for å kunne bruke indeksene.

B

Se etter om de har fått til å fjerne elementer in-place, altså redigere innholdet i eksisterende liste. Retur av ny liste gir ikke ønsket resultat.

C

Må ta høyde for at studenter kommer i ukjent rekkefølge, holder ikke bare å ta ut 6 og 6 lineært. Må ta inn parameter på korrekt vis, og returnere en dictionary som er riktig.

Oppgave 3

De har lært om lenket liste, og det er et MÅ-krav at de klarer å opprette objekt av riktig klasse (ikke Node). Må sette inn vha. metoden push. Om variabelnavnet er feil så trekker vi ikke.

Oppgave 4

A

Om de ikke klarer å opprette en enkel klasse med en __init__ så er det 0 poeng. Enkle get-metoder bør også være på plass, og de må ha med self. Dette er helt elementært, feil på denne trekkes hardt.

B

Trekk mye om liste og saldo-verdi tas som parameter, det viser manglende forståelse (vi har hatt fokus på at __init__ kan opprette variabler uten at de kommer som parameter). Ellers er dette mye samme som forrige oppgave, de bør klare å lage enkle metoder som justerer/henter instansvariabler.

C

Parametere til metodene bør absolutt være på plass. Ellers trekkes det mer eller mindre avhengig av hvor godt de har klart å implementere algoritmene.

D

Spesielt viktig her at legg_til_bruker og opprett_annonse ikke tar inn bruker- eller annonse-objekt som parameter. Vi vil ha inn nevnte parametere og la Salgsportal håndtere opprettelse og lagring av bruker-objekt, og la bruker-objektene håndtere annonse-objekter. Egentlig et slags aggregering-pattern. Gjenbruk av eksisterende lag_annonse i bruker er viktig, trekk en god del om de begynner å aksessere annonse-lista til brukere direkte fra Salgsportal. De må forstå at når de har laget en metode fra før i en klasse, så er det denne som brukes fordi funksjonaliteten alt eksisterer.

E

Gjenbruk av metoder viktig, som i forrige oppgave. Ellers trekk mer/mindre avhengig av hvor nært de er å faktisk få til algoritmen.

F

Samme som over, merk at «brukeren selger varen» innebærer at de bruker selg-metoden på riktig bruker-objekt.

G

Akkurat samme som foregående oppgaver, sjekk at de gjenbruker metoder de alt har skrevet. Husk å sjekke at de nullstiller saldoer (skjer gjennom nullstill_saldo på bruker-objektene). Trenger ikke ta høyde for feilhåndtering der ingen brukere har solgt noe som helst.

Oppgave 5

Det gjør ikke noe (dvs. Full score) om de har med duplikater i motsatt rekkefølge (typ 4+6 og 6+4). Ellers trekkes det etter hvor nært/langt unna de har en fungerende algoritme.