Linear probing 10 poeng

(a) Vi starter med et tomt array på størrelse 10.

0	1	2	3	4	5	6	7	8	9

Hashfunksjonen du skal bruke er $h(k, N) = k \mod N$, som for dette eksempelet blir det samme som $h(k, 10) = k \mod 10$. Altså hasher et tall til sitt siste siffer.

Bruk *linear probing* til å sette inn disse tallene i den gitte rekkefølgen:

$$21, 54, 82, 10, 20, 44$$

Fyll ut tabellen slik den ser ut etter alle tallene er satt inn med linear probing. Skriv svaret som en kommaseparert liste, der _ kan brukes for å indikere en tom plass.

(b) Forklar kort hvordan algoritmen for innsetting ved linear probing fungerer, og skisser algoritmen med pseudokode. Du kan anta at arrayet i input ikke er fullt, og at du har en hashfunksjon h slik at $h(k, N)$ gir et tall mellom 0 og $N - 1$ for en vilkårlig nøkkel k. Ledig plass i arrayet er indikert ved **`null`**.

```
Input: Et array A av størrelse N, en nøkkel k og verdi v
Output: Et array som inneholder (k, v)
1 Procedure LinearProbingInsert(A, k, v)
  | // ...
```

(a) Det resulterende arrayet ser slik ut:

10	21	82	20	54	44				
0	1	2	3	4	5	6	7	8	9

2 poeng for riktig svar, ingen poeng for feil. Vi trekker ikke for unøyaktigheter rundt antall tomme plasser.

(b) Linear probing fungerer ved å prøve å plassere nøkkel/verdi-paret på posisjonen som svarer til hashen av nøkkelen ($h(k, N)$). Dersom posisjonen er ledig kan vi sette inn nøkkel/verdi-paret og algoritmen er ferdig.

Dersom posisjonen er opptatt må vi:

- dersom nøkkelen på posisjonen er lik k må vi overskrive verdien;
- dersom nøkkelen på posisjonen er ulik må vi prøve neste posisjon.

```
Input: Et array A av størrelse N, en nøkkel k og verdi v
Output: Et array som inneholder (k, v)
1 Procedure LinearProbingInsert(A, k, v)
2   i ← h(k, N)
3
4   while A[i] ≠ null do
5     if key at A[i] is k then
6       break                          // Overskriv med ny verdi
7     i ← i + 1 mod N
8
9   A[i] ← (k, v)
```

Det viktigste er at studenten formidler en god forståelse av hvordan linear probing fungerer.

Opp til 5 poeng kan tildeles for en god forklaring; for full uttelling må også korrekt pseudokode være på plass. Et par ting å være oppmerksom på:

- Det trekkes 1 poeng for å ikke overskrive en lik nøkkel med ny verdi.
- Det trekkes 2 poeng dersom prosedyren vil aksessere indekser utenfor området $1 \ldots N - 1$.
- Det trekkes 1 poeng for feil bruk av hash-funksjonen (for eksempel $i \leftarrow h(i + 1, N)$ og lignende).