# Detecting Molecules

Petr jobber for et selskap som har laget en maskin som detekterer molekyler. Vekten av hvert molekyl er et heltall. Maskinen har et *deteksjonområde* \([l, u]\), hvor \(l\) og \(u\) er heltall. Maskinen kan detektere et sett med molekyler hvis, og bare hvis, dette settet innholder et subsett av molekyler med en total vekt som er innenfor maskinens deteksjonsområde.

Formelt, gitt \(n\) molekyler med vekter \(w_0, \ldots, w_{n-1}\). Deteksjonen er vellykket hvis det er et sett med distinkte indekser \(I = \{i_1, \ldots , i_m\}\) slik at \(l \le w_{i_1} + \ldots w_{i_m} \le u\).

På grunn av spesifikasjonen til maskinen, så er avstanden mellom \(l\) og \(u\) garantert å være større eller lik vektavstanden mellom det tyngste og det letteste molekylet. Formelt, \(u - l \ge w_{max} - w_{min}\), hvor \(w_{max}=\max(w_0, \ldots , w_{n-1})\) og \(w_{min}=\min(w_0, \ldots , w_{n-1})\).

Din oppgave er å skrive et program som enten finner hvilket som helst subsett av molekylene med totalt vekt innenfor deteksjonsområde, eller finner ut at et slikt subsett ikke eksisterer.

## Implementasjonsdetaljer

Du skal implementere en funksjon (metode):

- `int[] solve(int l, int u, int[] w)`
  - `l` og `u`: ytterpunktene i deteksjonsområdet,
  - `w`: vektene på molekylene.
  - hvis det nødvendige subsettet eksisterer, skal funksjonen returnere et array med indekser på molekylene som danner et slikt subsett. Hvis det finnes flere korrekte løsninger, returner hvilket som helst av dem.
  - hvis det nødvendige subsettet ikke eksisterer, skal funksjonen returnere et tomt array.

For språket C er funksjonen litt annerledes:

- `int solve(int l, int u, int[] w, int n, int[] result)`
  - `n`: antall elementer i w (altså antallet molekyler),
  - de andre parameterene er det samme som over.
  - istdenfor å returnere et array med \(m\) indekser (slik som over), skal funksjonen skrive indeksene til de første \(m\) cellene av arrayet `result` og så returnere \(m\).
  - hvis det ikke finnes et subsett som oppfyller kravene, så skal ikke funksjonen skrive noe til arrayet 'result`, og funksjonen skal returnere \

(0\).

Programmet ditt kan skrive indeksene til det returnerte arrayet (eller til arrayet `result` array in C) i en hvilken som helst rekkefølge.

Bruk de utleverte templatfilene for instrukser på implementeringen i ditt programmeringspråk.

## Eksempler

### Eksempel 1

`solve(15, 17, [6, 8, 8, 7])`
I dette eksempelet har vi fire molekyler med vektene 6, 8, 8 og 7. Maskinen kan detektere subsett av molekyler med totalvekt mellom 15 and 17, inklusivt. Merk at, \(17-15 \ge 8-6\). Den totale vekten av molekylene 1 og 3 er \(w_1 + w_3 = 8 + 7 = 15\), så funksjonen kan returnere `[1, 3]`. Andre mulige riktige svar er `[1, 2]` (\(w_1 + w_2 = 8 + 8 = 16\)) og `[2, 3]` (\(w_2 + w_3 = 8 + 7 = 15\)).

### Eksempel 2

`solve(14, 15, [5, 5, 6, 6])`
I dette eksempelet har vi fire molekyler med vektene 5, 5, 6 og 6, og vi ser etter et subsett av dem med en totalt vekt mellom 14 og 15, inklusivt. Igjen, merk at \(15-14 \ge 6-5\). Det er ingen subsett blant molekylene med total vekt mellom \(14\) og \(15\), så funksjonen skal returnere et tomt array.

### Eksempel 3

`solve(10, 20, [15, 17, 16, 18])`
I dette eksempelet har vi fire molekyler med vektene 15, 17, 16 og 18, og vi ser etter et subsett med den totale vekten mellom 10 og 20, inklusivt. Igjen, merk at \(20-10 \ge 18-15\). Hvilket som helst subsett med nøyaktig ett element tilfredsstiller kravene, så de korrekte svarene er: `[0]`, `[1]`, `[2]` og `[3]`.

## Subtasks

1. (9 points): \(1 \le n \le 100\), \(1 \le w_i \le 100\), \(1 \le u, l \le 1000\), alle \(w_i\) er like.
2. (10 points): \(1 \le n \le 100\), \(1 \le w_i, u, l \le 1000\) og \(\max(w_0, \ldots , w_{n-1}) - \min(w_0, \ldots , w_{n-1}) \le 1\).
3. (12 poeng): \(n \le 100\) og \(w_i,u,l \le 1000\).
4. (15 poeng): \(n \le 10\,000\) og \(w_i,u,l \le 10\,000\).
5. (23 poeng): \(n \le 10\,000\) og \(w_i,u,l \le 500\,000\)
6. (31 poeng): \(n \le 200\,000\) og \(w_i,u,l < 2^{31}\).

## Sample grader

Eksempelgradereren leser input i følgende format:

- linje 1: heltall \(n\), \(l\), \(u\).
- linje 2: \(n\) heltall: \(w_0, \ldots , w_{n-1}\).