

# HUGO

## PROBLEM

Hugo, som er en lille trolld i et computerspil, har til opgave at gribe så mange æbler som muligt. Hugo bevæger sig på en sti nederst på skærmen. Stien er inddelt i  $N$  lige store felter, hvor  $N$  er ulige. Felterne er nummereret fra 1 (feltet længst til venstre) til  $N$  (feltet længst til højre). Fra tid til anden falder der et æble ned på de enkelte felter. Hvis Hugo på det tidspunkt befinder sig på feltet, hvor æblet falder, kan han gribe det.

Ved spillets begyndelse ( $T=1$ ) er Hugo på midterfeltet (der er et ulige antal felter). Hugo kan bevæges et felt til højre eller et felt til venstre eller blive stående. Hvis Hugo således til tidspunktet  $T$  befinder sig på feltet  $P$ , kan han til næste tidspunkt,  $T+1$  flyttes til  $P+1$  eller  $P-1$  (men altså ikke udenfor 1 til  $N$ ) eller blive på  $P$ . Hugo ved på forhånd, hvor og hvornår æblerne falder.

Det er nu din opgave at finde ud af, hvor mange æbler Hugo maksimalt kan gribe.

## OPGAVE

Skriv et program der

- fra standard-input indlæser antallet af felter, som Hugo kan bevæge sig på, samt hvordan æblerne falder.
- beregner hvor mange æbler, Hugo maksimalt kan gribe.
- skriver resultatet til standard-output.

## INPUT

Input læses fra standard-input og består af et antal linjer:

- Første linje indeholder et ulige heltal,  $N$ , der angiver antallet af felter ( $1 \leq N \leq 999$ ).
- De næste  $N$  linjer indeholder information om de faldende æbler, således at der på linje  $i+1$  er data om æblerne på felt nummer  $i$ . På hver af disse linjer er der først et heltal  $K$  ( $1 \leq K \leq 3000$ ), som angiver antallet af æbler, der falder på felt  $i$ . Dernæst følger  $K$  heltal (mellem 1 og 100 000) i voksende orden. Disse heltal angiver de tidspunkter, som æblerne falder på. De enkelte heltal på en linje er adskilt af et mellemrum.

## OUTPUT

Output skal skrives til standard-output og består af et enkelt heltal, antallet af æbler, Hugo maksimalt kan gribe.

## EKSEMPLER

input

```
5
2 2 6
2 1 5
1 4
1 3
1 2
```

output

```
4
```

```
7
2 1 2
1 8
2 2 3
2 2 5
2 1 2
2 3 5
2 3 7
```

```
4
```

```
7
2 6 7
3 5 7 8
2 2 4
3 2 3 5
3 5 6 7
2 3 6
1 5
```

```
7
```