Innlevering 1

Høst 2026

Om innleveringen

- Innleveringen skal leveres i innleverings- og tilbakemeldingssystemet Devilry innen **18. september klokken 23:59**.
- Dere skal helst *levere i grupper* på inntil tre personer, under forutsetning av at alle i gruppen både står for det som leveres og alene kan gjøre rede for alle delene av arbeidet.
- Dere skal selv finne og opprette grupper i Devilry, *senest én uke før fristen for innleveringen*.
- Det er mulig å få fritak fra gruppelevering ved å fylle ut følgende nettskjema: `https://nettskjema.no/a/645741`.
- Du skal bruke programmet `in2010-testrunner` som du finner her: `https://github.uio.no/IN2010/in2010-testrunner/`
- `in2010-testrunner` hjelper deg med å:
  - generere kodeskjelett,
  - kjøre koden mot testfiler, og
  - lage en zip-fil som skal leveres i Devilry.
- Med mindre noe annet er spesifisert kan dere bruke hva dere vil fra Java eller Python sine standardbibliotek, men ingen andre biblioteker.
- Kommenter **«Klar for retting»** i Devilry hvis du leverer i god tid før fristen, og innleveringen er klar for retting.

Krav til innleveringen

- Alle oppgaver skal være besvart.
- Programmene må kompilere og kunne kjøres med kommandoen dere har oppgitt.
- Det som leveres skal være *lett forståelig, entydig og presist*.

Teque

> Oppgaven er hentet fra Kattis[1]. Vi følger samme format på input- og output, slik at oppgaven deres kan lastes opp på Kattis, men dette er *ikke* et krav. Det er heller ikke nødvendig å oppfylle tidskravet som Kattis stiller.

Deque, eller *double-ended queue*, er en abstrakt datatype som støtter effektiv innsetting på starten og slutten av en kø-struktur. Den kan også støtte effektivt oppslag på indekser med en array-basert implementasjon.

Dere skal utvide idéen om deque til *teque*, eller *triple-ended queue*, som i tillegg støtter effektiv innsetting i midten. Altså skal *teque* støtte følgende operasjoner:

`push_back(`x`)` sett elementet x inn bakerst i køen.

`push_front(`x`)` sett elementet x inn fremst i køen.

`push_middle(`x`)` sett elementet x inn i midten av køen. Det nylig insatte elememtentet x blir nå det nye midtelementet av køen. Hvis k er størrelsen på køen før innsetting, blir x satt inn på posisjon $\lfloor (k+1)/2 \rfloor$.

`get(`i`)` printer det i-te elementet i køen.

Merk at vi bruker 0-baserte indekser.

Input

Første linje av input består av et heltall N, der $1 \leq N \leq 10^6$, som angir hvor mange operasjoner som skal gjøres på køen.

Hver av de neste N linjene består av en streng S, etterfulgt av et heltall. Hvis S er `push_back`, `push_front` eller `push_middle`, så er S etterfulgt av et heltall x, slik at $1 \leq x \leq 10^9$. Hvis S er `get`, så S etterfult av et heltall i, slik at $0 \leq i <$ (størrelsen på køen).

Merk at du ikke trenger å ta høyde for ugyldig input på noen som helst måte, og du kan trygt anta at ingen `get`-operasjoner vil be om en indeks som overstiger størrelsen på køen.

Output

For hver `get`-operasjon, print verdien som ligger på den i-te indeksen av køen.

Eksempel-input	Eksempel-output
9	3
push_back 9	5
push_front 3	9
push_middle 5	5
get 0	1
get 1	
get 2	
push_middle 1	
get 1	
get 2	

Oppgaver

(a) Skriv et Java eller Python-program som leser input fra `stdin` og printer output *nøyaktig* slik som beskrevet ovenfor. Lavere kjøretidskompleksitet på operasjonene er bedre.

(b) Oppgi en kort verste-tilfelle kjøretidsanalyse for hver operasjon med $\mathcal{O}$-notasjon som en kommentar over implementasjonen. I analysen fjerner vi begrensningen på N, altså kan N være vilkårlig stor.

[1] https://open.kattis.com/problems/teque

Sortering

I denne oppgaven skal du implementere minst to sorteringsalgoritmer!

Disse to sorteringsalgoritmene skal implementeres:

- Insertion sort
- Merge sort

For den ekstra ivrige: Du kan implementere flere sorteringsalgoritmer dersom du ønsker, også de som ikke ennå er gjennomgått eller utenfor pensum. Dersom du velger sorteringsalgoritmer som er utenfor pensum må du selv sørge for at implementasjonen er forståelig for retteren, og kan ikke forvente hjelp fra gruppelærer i like stor grad.

Input

Input består av n heltall. For hvert tall i er det slik at $0 \leq i \leq 2^{31} - 1$. Input skal leses fra `stdin`.

Eksempelinput:

```
80
91
7
33
50
70
13
321
12
```

Output

Output består av de samme n heltallene ordnet fra minst til størst.

Gitt eksempelinputet ovenfor bør vi få følgende output.

Eksempeloutput:

```
7
12
13
33
50
70
80
91
321
```