Opgave i C++

I denne opgave skal der primært arbejdes med filer, sekundært med klasser, objekter og `vector`.
Lav en tekstfil ved navn `data.txt` med følgende indhold:

```
1;A;3
1;B;6
1;C;4
2;D;5
3;E;4
3;F;1
4;G;2
4;H;7
4;I;1
5;J;4
```

Den type af filer kaldes *kommaseparerede* filer og bruges ofte til logning af måledata og andre strukturerede data.

Her er der tale om data til en aktivitetsplan for et udviklingsprojekt med tre informationer pr aktivitet.

Den første information kalder vi *event,* og den angiver rækkefølgen af aktiviteterne; den anden kalder vi *task,* og den angiver navnet på aktiviteten; den sidste hedder *duration* og indeholder aktivitetens varighed i dage.

Opgaven går ud på følgende:

- Lav en klasse ved navn `Aktivitet,` som indeholder attributterne `event` (`int`), `task` (`string`) og `duration` (`int`) med de sædvanlige metoder (constructors, get-metoder).
- Indlæs aktiviteterne fra filen og isoler attributterne.
- Hver aktivitet transformeres til et `Aktivitet`-objekt, og objekterne placeres i en `vector` ved navn `tabel`.
- Herefter udskrives antallet af aktiviteter (10) og den gennemsnitlige varighed som decimaltal (3,7 dage).

- Udførelsen af de enkelte aktiviteter er afhængig af attributten event. Fx kan task D (event 2) ikke udføres, før A, B og C (event 1) alle er udført, og task J (event 5) kan ikke udføres før G, H og I (event 4) er udført. Aktiviteter med samme event-nummer kan altså udføres parallelt/samtidigt.
- Der er således en række aktiviteter, der kan kaldes *den kritiske vej*, hvorom det gælder, at ingen af de indgående aktiviteter kan blive forsinkede, uden at det forsinker hele projektet. Fx kan task A godt blive forsinket, fordi task D alligevel ikke kan starte, før task B er færdig. Men en forsinkelse af B, forsinker hele projektet.
- Du skal udskrive varigheden af den kritiske vej, dvs. den tid det minimum tager at udføre projektet (26 dage) samt rækkefølgen af tasks (B, D, E, H, J). Opgaven skal løses ved at gennemløbe `vector tabel`, og ikke ved at læse inputfilen igen.
- Precondition: du kan antage, at events er i stigende orden startende med 1 og uden 'huller'. Tasks, derimod, kommer ikke altid nødvendigvis i stigende alfabetisk orden som i dette tilfælde.
- Endelig skal du lave en helt ny tekstfil (eller flere) med andre data, som du kan teste dit program yderligere med.

I forhold til hvordan programmeringsopgaver løses i industrien, er det af afgørende betydning, at du følger ovenstående specifikation til punkt og prikke. Fx ikke noget med at bruge andre navne på attributterne end de angivne eller at bruge et `array` i stedet for en `vector`.

Supplerende opgave 1

Opgaven går ud på at finde den aktivitet, som har det største slæk (slack) og tillige angive varigheden af slækket.

Slæk kan defineres som det antal tidsenheder, der er til rådighed til at udføre aktiviteten minus aktivitetens eget tidsforbrug. I dataeksemplet (se ovenfor) har aktivitet A 3 tidsenheders slæk – forskellen mellem event 1's længste aktivitet (B), som er 6 og A's tidsforbrug, som er 3.

Den korrekte aktivitet og slækkets varighed skal udskrives, og det rigtige svar er aktivitet **I** med et slæk på **6** tidsenheder.

Supplerende opgave 2

Opgaven går ud på at beregne den samlede mængde af slæk (slack) i hele projektet.

Slæk kan defineres som det antal tidsenheder, der er til rådighed til at udføre aktiviteten minus aktivitetens eget tidsforbrug. I det dataeksemplet (se ovenfor) tager event 1 6 (seks) tidsenheder, nemlig den længstvarende aktivitets tidsforbrug, som er 6 tidsenheder. Aktivitet A har 3 tidsenheders slæk – forskellen mellem event 1's længste aktivitet (B), som er 6 og A's tidsforbrug, som er 3. Aktivitet C's slæk er 2 tidsenheder (forskellen mellem 6 og 4).

Det samlede slæk udskrives, og det korrekte svar i ovenstående eksempel er 19 tidsenheder.

Tip: løs opgaven i en separat metode med aktivitetstabellen som parameter og genbrug af dele af koden fra main-metoden. Opgaven bliver nemmere, hvis man sorterer aktiviteterne.

Ole Dolriis, april 2025