

Algoritmisk oppgave/problem – Case 4:

(Teksten er hentet fra oppg.nr.4 på eksamen 19/12-2017 i IMT2021-Algoritmiske metoder)

Vi skal finne tall mellom 1 og 100.000 som er likt når de blir rotert 180° (mao: opp-ned).

Dette er aktuelt når tallet *kun* består av sifrene: 0, 1, 6, 8 og 9

(2, 3, 4, 5 og 7 er *ikke* andre tall når de er snudd opp-ned).

En del eksempler på slike tall er: 1, 8, 96, 916, 1001, 1691, 1881, 6009, 6969, 8008, 8968, 10001, 11811, 19061, 19861, 66099, 68089, 80108, 86098, 89168 og 96096

Lag funksjonen `int snuOppNed(const int n)`

Funksjonen forsøker å snu tallet `n` opp-ned. Lykkes dette (dvs. tallet inneholder *kun* relevante sifre), så returnerer den opp-ned-tallet (altså tallet lest baklengs etter at det er snudd opp-ned. Dvs. at man leser/ser tallet på en vanlig måte, fra venstre mot høyre, om arket det er skrevet på er rotert 180°). Er det ikke mulig å snu det opp-ned, returneres 0 (null).

Lag også et fullstendig program som går gjennom alle aktuelle tall, bruker funksjonen ovenfor, skriver ut alle tall som er «snubare» og totalantallet av dem.

NB: F.eks. 19881 er det mulig å snu opp-ned, men det er *ikke* det samme lest opp-ned.