```python
#Oppgave 2a.

def binomialCoefficient(n, i): # funksjon for utregning av binomialCoefficient
    binCoe = 1
    for j in range(1, i + 1):
        binCoe *= ((n - j + 1) / j)
    return binCoe

print(binomialCoefficient(5000,4))
print(binomialCoefficient(1000,500))
print(binomialCoefficient(100000,99940))
"""
Kjøreeksempel
kvale@Aleksanders-MBP MAT-INF 1100 % python3 binomialcoefficient.py
26010428123750.0
2.702882409454359e+299
inf
"""

#Oppgave 2b
"""
Det er veldig lurt å bruke flyttall da resultatet blir upresist dersom programmet begynner å runde opp/ned til hele flyttall
man kan også risikere at ett av leddene blir lik 0.x, og programmet runder ned til 0. Uansett risikerer man å ende opp med
upresist og/eller feil tall.
Ved 88 iterasjoner av den siste binomial koeffisienten når binCoe en verdi på 5.188 * 10^305, og spytter deretter ut infinity
Da har koden nådd det største flyttallet som kan representeres på maskinen
"""

#Oppgave 2c
"""
Denne metoden er relativt effektiv for alle verdier opp til i > n / 2.
Dersom dette inntreffer vil det være mer effektivt for programmet å kjøre utregningen for binomialkoeffisienten
på motsatt side av pascals trekant. Dvs. gjøre om [i] i formelen til n - i.
Feks. (5 over 4) er det samme som (5 over 1), det blir færre iterasjoner for programmet dersom vi setter
i = n - i når vi skal regne ut (5 over 4). Istendenfor å kjøre gjennom for løkken 4 ganger,
trenger det bare å kjøre igjennom en gang.
Denne metoden ville gjort at vi faktisk kan løse (100000 over 99940) som vi fikk OverFlow error på over.
"""

```