**Am ales sa analizez invariantul pentru cazul relatiei de recurenta cand x > 1**

Termenii din sir se cumuleaza pana la o valoare “k” care este strict mai mica decat eroarea epsilon. Putem lua o variabla ajutatoare “s” care sa exprime aceasta suma, iar invariantul ar putea fi reprezentat de faptul ca “s” stocheaza suma termenilor pana la valoarea “k”.

I = {s = }

Voi transforma relatia de recurenta intr-un ciclu “while”

Function serie\_2(x,eps,n):

curent = (((x-1)/x)\*\*n)/n

sum = curent

if abs(curent < eps):

returneaza n

else:

while curent >= eps do

n += 1

curent = (((x-1)/x)\*\*n)/n

sum += current

returneaza n

**(i)**Invariantul este corect inainte de intrarea in ciclu deoarece cand n=1, variabila “sum” trebuie sa stocheze deja primul termen calculat, iar acest lucru se intampla.

**(ii)**Invariantul este corect pe parcursul ciclului, deoarece la fiecare pas se calculeaza noul termen care este adaugat la variabila “sum”.

(iii)La iesirea din ciclu, invariantul implica postconditia, s-a ajuns la al k-lea termen care este strict mai mic decat eroarea epsilon, iar ciclul se termina.

**(iv)**Finitudine

Finitudinea depinde de precizia cu care vrem rezultatul, cand aceasta este mare, vor aparea limitarile calculatorului.

Finitudinea ar putea depinde de valoarea variabilei “curent”, acesta este descrescatore, iar functia care depinde de ea va fi descrescatoare, cu cat termenul va fi mai mic, cu atat diferenta pana la valoarea epsilon va fi mai mica.

**F(p) = current –**  (numarul de zecimale prin care trebuie sa trecem pentru a ajunge strict mai mic ca )