RISOLUZIONE APPROSSIMATA DI UN'EQUAZIONE

L'equazione $e^x = x + 2$ ammette una sola soluzione **c** nell'intervallo [0; 2].
Scrivere un programma in Python che permetta di calcolare **c** a meno di 10^{-n}, con n intero positivo a scelta.

```
#!/usr/bin/env python
# coding=latin-1
print "Teorema degli zeri per l'equazione exp(x)=x+2 e l'intervallo [0;2]; approssimazione 10^(-n)"
import math
n=int(raw_input("Inserisci un numero intero n: "))
def f(t):
            return math.exp(t)-t-2
passo=pow(10,-n)        # oppure 10**(-n)
x=0
while f(x)*f(x+passo)>0:
            x=x+passo
print "la funzione f(x)=exp(x)-x-2 assume questi valori"
print "in x=",x," f(x)=",f(x)
print "in x=",x+passo," f(x)=",f(x+passo)
print "Quindi la soluzione è compresa fra ",x," e ",x+passo
```

Metodo di bisezione (o dicotomia)

Il programma precedente si può ottimizzare utilizzando il cosiddetto metodo di bisezione (o dicotomia).
Il metodo consiste nel dimezzamento successivo dell'intervallo di ricerca della soluzione, operando in questo modo:

1. si pone c=(a+b)/2, cioè c è il punto medio dell'intervallo [a;b]; indicata con f(x)=0 l'equazione in esame, si controlla se f(c)=0: se ciò è vero il programma termina, altrimenti:
2. si controlla se f(c) ha il segno di f(a) o di f(b): se c ha il segno di f(a), si sostituisce a con c e si ripete il punto 1, altrimenti si sotituisce b con c e si ripete il punto 1;
3. il programma prosegue fin quando la lunghezza dell'intervallo è superiore alla precisione desiderata.

Ecco un possibile codice

```
#!/usr/bin/env python
# coding=latin-1
print
print "Radice di un'equazione con il metodo di bisezione:"
print "exp(x)=x+2 nell'intervallo [0;2]"
print
import math
a=0.0
b=2.0
n=int(raw_input("Radice approssimata a meno di 10 alla -n; n="))
def f(x):
            return      math.exp(x)-x-2
def comunica(radice):
            print "Soluzione dell'equazione approssimata a meno di 10 alla -",n,":",radice
c=(a+b)/2
if f(c)==0:
            comunica(c)
else:
            while (b-a)>pow(10,-n-1):
                        c=(a+b)/2
                        if f(c)==0:
                                    comunica(c)
                        else:
                                    if f(c)*f(a)<0:
                                                b=c

                                    if f(c)*f(b)<0:
                                                a=c
            comunica(c)
print
print "Fine programma"
```

ESERCIZIO

Riscrivere i due programmi precedenti aggiungendo le istruzioni necessarie per valutare il tempo di esecuzione del programma e confrontare i due metodi.

Codice 1 (... metodo lento)

```
#!/usr/bin/env python
# coding=latin-1

print "Teorema degli zeri per l'equazione exp(x)=x+2 e l'intervallo [0;2]; approssimazione 10^(-n)"
import math,time
n=int(raw_input("Inserisci un numero intero n: "))
start=time.time()
def f(t):
            return math.exp(t)-t-2
passo=pow(10,-n)        # oppure 10**(-n)
x=0
while f(x)*f(x+passo)>0:
            x=x+passo
print "la funzione f(x)=exp(x)-x-2 assume questi valori"
print "in x=",x," f(x)=",f(x)
print "in x=",x+passo," f(x)=",f(x+passo)
print "Quindi la soluzione è compresa fra ",x," e ",x+passo

stop=time.time()-start
print "Calcolo effettuato in",stop,"secondi"
```

Codice 2 (... metodo veloce)

```
#!/usr/bin/env python
# coding=latin-1

print
print "Radice di un'equazione con il metodo di bisezione:"
print "exp(x)=x+2 nell'intervallo [0;2]"
print
import math,time
a=0.0
b=2.0
n=int(raw_input("Radice approssimata a meno di 10 alla -n; n="))
start=time.time()
def f(x):
            return      math.exp(x)-x-2
def comunica(radice):
            print "Soluzione dell'equazione approssimata a meno di 10 alla -",n,":",radice
c=(a+b)/2
if f(c)==0:
            comunica(c)
else:
            while (b-a)>pow(10,-n-1):
                        c=(a+b)/2
                        if f(c)==0:
                                    comunica(c)
                        else:
                                    if f(c)*f(a)<0:
                                                b=c

                                    if f(c)*f(b)<0:
                                                a=c
            comunica(c)

stop=time.time()-start
print "Calcolo effettuato in",stop,"secondi"
```