

Autor: Paweł Salwa

Złożoność poniższego zadania:

Exp() – 30,

Cos() – 30,

25 razy przejście po pętli z 5 obliczeniami – 125.

Output:

dla n=27 i x=0.2

wynik=1.543669495535392

```
from math import cos, exp
#sum_0^N cos nx exp(-n)
#cos((n+1)x) = 2. * cos(x) * cos(nx) - cos((n-1)x)
```

```
N=27;
```

```
x=0.2;
```

```
e = exp(-1.)
```

```
Cosx = cos(x)
```

```
wynik = 1 + Cosx*e
```

```
En = e
```

```
Cn_1 = Cosx
```

```
Cn_2 = 1
```

```
for n in range(2 , N+1):
    Cn = 2*Cn_1 * Cosx - Cn_2
    En = En * e
    wynik = wynik + Cn * En
    Cn_2 = Cn_1;
    Cn_1 = Cn;
```

```
print "dla n=" + repr(N) + " i x=" + repr(x)
```

```
print "wynik=" + repr(wynik)
```

```
#=====tutaj inna metoda- dzialajaca, choc jest zarloczna obliczeniowo:
```

```
#
```

```
#x = 0.2
```

```
#dokladnosc = 1.e-10
```

```
#
```

```
#e = exp(1.)
```

```
#moj_cosinus = cos(x)
```

```
#
```

```
#iloscWywolanCos_nx =0
```

```
#
```

```
#def cos_nx( n ):
```

```
#   global iloscWywolanCos_nx
```

```

# iloscWywolanCos_nx = iloscWywolanCos_nx +1
# if n==0:
#     return 1.
# if n==1:
#     return moj_cosinus
# return 2.*cos_nx( 1 ) * cos_nx( n-1 ) - cos_nx( n-2 )
#
#def wynik(N):
#    suma = 0.
#    for n in range(N+1):
#        suma += cos_nx(n) * e**(-n)
#    #
#    return suma
#
#
#print 'wynik = ', wynik(27)
#
#print 'iloscWywolanCos_nx = ', iloscWywolanCos_nx

```