z.1.

1)

Procedura pierwsza przyjmuje wartość n - liczbę całkowitą, >=0, x – liczbę rzeczywistą oraz listę a[] (n+1) wartości rzeczywistych.

Przypisuje ona na początku wartość 0.0 do zmiennej s.

Następnie w pętli, której licznikiem jest i iterującej się od n do 0 powtarzamy, że do zmiennej p przypisz a[n-1], czyli drugi od końca element z naszej listy dalej uruchamiamy drugą pętlę, której licznikiem jest j, iterujemy od j=i do 0, zmniejszając o 1. Wewnątrz tej pętli mnożymy p = p\*x, potem wychodzimy z pętli i sumujemy s = s + p

Ostatecznie zwracamy wynik procedury, czyli s.

Liczy s (sumę) od i = 0 do i= n z a[n-1] \* (x^i)

2) Procedurą druga przejmuje dokładnie te same argumenty

Przypisuje ona na początku wartość od zmiennej s wartość a[0].

Zaczynamy iterować, licznikiem pętli jest zmienna i, pętla wykonuje się od n-1 do 0, zmniejszając się o 1, wewnątrz pętli przypisujemy zmiennej s wartość s = s\*x + a[n -i]

Liczy s (sumę) od i = 0 do i = n – 1 z s\*x + a[n-1]

3) W krótszym czasie wykona się procedura numer 2, ze względu na niższą złożoność czasową, występuje w niej jedna pętla zatem złożoność to O(n) – liniowa, a tej 1 ze względu na pętlę w pętli złożoność to

O(n^2) – kwadratowa.

z.2

1) Jeżeli switch jest ustawiony na False to obliczymy, wartość charakterystyki fazowej dla zadanej przez użytkownika częstotliwości. Switch ustawiony na True pozwala obliczyć wartość reaktancji pojemnościowej kondensatora użytego w obwodzie.

2)

switch = True

w = -8.65511591008504e-09 [F]

switch = False

w = -1.5707849610201994 [rad]

3) Kod

import math  
  
def procedura(R,L,C,f,switch):  
 w = 0  
 if switch == True:  
 w = 1.0/(4 \* (math.pi\*\*2) \* (f\*\*2) \* L)  
 return w  
 else:  
 w = (math.atan(2\*math.pi\*f\*L-((1.0/(2\*math.pi\*f\*C)))/R))  
 return w  
  
I=6  
N=7  
R=I\*10  
L=N\*10^(-5)  
A=8  
f=(I+N+A)\*10^3  
C=(A+1)\*10^(-6)  
switch=True  
print(procedura(R,L,C,f,switch))