

# ALGORÍSMICA I – CURS 2009/2010

## PROVA 20/10: PROGRAMACIÓ BÀSICA EN PYTHON

### Criteris d'avaluació (per ordre d'importància):

- 1.- El programa ha de donar un resultat correcte : 50% de la nota.
- 2.- Ús adequat del llenguatge (fer servir if/while correctament, fer servir la col·lecció adequada, etc.): 30% de la nota.
- 3.- Bon estil de programació (fer una interfase d'usuari adequada, comentaris, etc.): 20% de la nota.

Creeu un únic mòdul, **prova.py**, amb les següents funcions:

1. Escriu una funció (**nombres**) que, donats 5 nombres enters entrats per l'usuari, determini quin dels quatre últims nombres és el més proper al primer (Per exemple, si l'usuari introdueix 2, 6, 4, 1 y 10, el programa ha de respondre que el nombre més proper al 2 és l'1 ). (Punts: 20/100)
2. Escriu una funció (**consonant**) que (1) demani a l'usuari una frase F, (2) demani a l'usuari una consonant C, i (3) demani a l'usuari un nombre enter N, i llavors substitueixi totes les vocals de la frase per la consonant entrada C, sempre i quan hi hagi a la frase F un nombre de vocals > N. (Punts: 30/100)
3. Feu una funció (**seq**) que trobi quina és la seqüència de 5 dígits consecutius del nombre N amb el producte més gran. (Punts: 50/100)

N=731671765313306249192251196744265747423553491949349698352031277450632623957831801  
69848018694788518438586156078911294949545950173795833195285320880551112540698747158  
52386305071569329096329522744304355766896648950445244523161731856403098711121722383  
11362229893423380308135336276614282806444486645238749303589072962904915604407723907  
13810515859307960867017242712188399879790879227492190169972088809377665727333001053  
36788122023542180975125454059475224352584907711670556013604839586446706324415722155  
39753697817977846174064955149290862569321978468622482839722413756570560574902614079  
72968652414535100474821663704844031998900088952434506585412275886668811642717147992  
44429282308634656748139191231628245861786645835912456652947654568284891288314260769  
00422421902267105562632111110937054421750694165896040807198403850962455444362981230  
98787992724428490918884580156166097919133875499200524063689912560717606058861164671  
09405077541002256983155200055935729725716362695618826704282524836008232575304207529

## Prova 1 - Solució 1

---

### def nombres():

```
import string
t = []
for n in range (5):
    x = input ("Introdueix un nombre enter: ")
    t.append(x)
dist = abs(t[0]-t[1])
res = [t[1]]
for n in range (2,5):
    if abs(t[0]-t[n]) < dist:
        res = [t[n]]
        dist = abs(t[0]-t[n])
    elif abs(t[0]-t[n]) == dist:
        res.append(t[n])
print
```

### def consonant():

```
import string
frase = ""
f = raw_input ("Entra una frase: ")
c = raw_input ("Entra una consonant: ")
n = input ("Entra un nombre enter: ")
num = n
llarg = len(f)
for x in range (llarg):
    if f[x]=="a" or f[x]=="e" or f[x]=="i" or f[x]=="o" or f[x]=="u" or f[x]=="A" or f[x]=="E" or
f[x]=="I" or f[x]=="O" or f[x]=="U" :
        num = num-1
if num<=0:
    for x in range (llarg):
        if f[x]=="a" or f[x]=="e" or f[x]=="i" or f[x]=="o" or f[x]=="u" or f[x]=="A" or f[x]=="E" or
f[x]=="I" or f[x]=="O" or f[x]=="U" :
            frase = frase + c
        else:
            frase = frase + f[x]
    print "Si hi havia mes de ", n, " vocals, les he canbiat per ", c,": ", frase
else:
    print "Com no hi havia mes de ", n, " vocals, no les he canbiat per ", c,": ", f
```

**def seq():**

import string

i = 0

file = open("N.txt","r")

text = file.readlines()

file.close()

x = text[0]

llarg = len (x)

num = int(x[0])\*int(x[1])\*int(x[2])\*int(x[3])\*int(x[4])

for n in range (1,llarg -7):

    aux = int(x[n])\*int(x[n+1])\*int(x[n+2])\*int(x[n+3])\*int(x[n+4])

    if aux > num:

        num = aux

        i = n

print "La sequencia amb el producte mes alt es: ", x[i],x[i+1],x[i+2],x[i+3],x[i+4]

## Prova 1 - Solució 2

---

### def nombres():

```
n = []
ref = input("Introdueix un nombre\n")
for i in range(4):
    tmp = input("Introdueix un nombre\n")
    if ref > tmp:
        dif = ref-tmp
    else:
        dif = tmp-ref
    n.append([tmp,dif])
print ref
print n
min = n[0][1]
pos = 0
for i in range(4):
    if n[i][1] < min:
        pos = i
        min = n[i][1]
print n[pos][0]
```

### def consonant():

```
f = raw_input("Introdueix una frase\n")
c = raw_input("Introdueix una consonant \n")
n = input("Introdueix un nombre enter\n")
cVocals = 0
pVocals = []
fResult = []
vocals = ["a", "e", "i", "o", "u"]
f = f.lower()
c = c.lower()
print f
for i in range(len(f)):
    for j in range(len(vocals)):
        if f[i] == vocals[j]:
            cVocals +=1
            pVocals.append(i)
if cVocals > n:
    for h in range(len(f)):
        canvi = 0
        for k in range(len(pVocals)):
            if h == pVocals[k]:
                canvi = 1
        if canvi != 0:
            fResult.append(c)
        else:
            fResult.append(f[h])
print "".join(fResult)
```

**def seq():**

```
    file = open("N.txt", "r")
    str = file.read()
    sProds = []
    c = 0
    for i in range(len(str)):
        prod = 1
        c += 1
        if len(str) - c >= 5:
            for j in range(0,5):
                prod = prod * int(str[i+j])
            sProds.append([prod,i])
    max = [0,0]
    for i in range(len(sProds)):
        if sProds[i][0] > max[0]:
            max[0] = sProds[i][0]
            max[1] = sProds[i][1]
    print max[0]
    print str[max[1]],str[max[1]+1],str[max[1]+2],str[max[1]+3],str[max[1]+4]
```

# ALGORÍSMICA I – CURS 2009/2010

## PROVA 26/10: PROGRAMACIÓ BÀSICA EN PYTHON

### Criteris d'avaluació (per ordre d'importància):

- 1.- El programa ha de donar un resultat correcte : 50% de la nota.
- 2.- Ús adequat del llenguatge (fer servir if/while correctament, fer servir la col·lecció adequada, etc.): 30% de la nota.
- 3.- Bon estil de programació (fer una interfase d'usuari adequada, comentaris, etc.): 20% de la nota.

Creeu un únic mòdul, **prova.py**, amb les següents funcions:

4. Escriu una funció (**potencia**) que calcula i imprimeix totes les potències de 2 que no comencen per 2 fins a un nombre determinat que introdueix l'usuari (per exemple, totes les potències de 2 que són menors que 50.000 i que no comencen pel dígit 2). (Punts: 20/100)
5. El "valor" d'una paraula es pot definir com el resultat de multiplicar el valor ASCII de cada una de les seves lletres. Per exemple, el valor de "casa" és  $99 \times 97 \times 115 \times 97 = 107121465$ . Donat un text en un fitxer (text.txt), feu una funció (**text**) que llegeixi tot el text i que busqui la paraula amb més valor. (Punts: 30/100)
6. Definim la funció  $f(n,d)$  com el nombre de vegades que el dígit  $d$  apareix en els enters menors o iguals que  $n$ . Per exemple:

$n$	0	1	2	3	4	5	6	7	8	9	10	11	12	...
$f(n,1)$	0	1	1	1	1	1	1	1	1	1	2	4	5	...

Considerem ara l'equació  $f(n,1)=n$ . Aquesta equació es compleix per  $n=0$  i per  $n=1$  (veure taula anterior). Escriu una funció, **valor**, que calculi quin és el següent valor de  $n$  pel que es compleix. (Punts: 50/100)

## Prova 2 - Solució 1

---

### def potencia():

```
print
potencia=2
cadena = "Las potencias son"
numero = input("Escribe un numero: ")
while potencia *2 <= numero:
    potencia= potencia *2
    digito = potencia
    while digito >= 10:
        digito = digito / 10
    if int(digito)<>2:
        cadena = cadena + "; " + str(potencia)
print cadena
```

### def text():

```
file = open("text.txt", "r")
text = file.readlines()
file.close()
cont = 0
valor=[1]
for line in text:
    for palabra in string.split(line):
        for letra in palabra:
            if (ord(letra) >= 65 and ord(letra) <= 90) or (ord(letra) >= 97 and ord(letra) <= 122):
                valor[cont]= valor[cont] * ord(letra)
        valor.append(1)
        cont= cont +1
mayor = valor[0]
for i in range(0,cont):
    if int(valor[i]) > int(mayor):
        mayor=valor[i]
print "El mayor es:", mayor
```

**def contar1(numero):**

```
    cont=0
    mul= 1
    while numero >= 10:
        digito = numero
        while digito >= 10:
            digito = digito / 10
            mul = mul *10
        numero = numero - int(digito)* mul
        mul = 1
        if int(digito)==1:
            cont = cont +1
    if numero==1:
        cont = cont +1
    return cont
```

**def valor():**

```
    print
    print
    print "EJERCICIO 3"
    contador =1
    n = 2
    while n <> contador:
        contador = contador + contar1(n)
        n= n+1
    print "El siguiente numero que cumple la funcion F(n, 1)=0 es:", str(n)
```



## Prova2 - Solució 2

---

**def primer\_digit(n):**

```
while (n > 10):
    n /= 10
if n == 2:
    return 1
else:
    return 0
```

**def potencia():**

```
x = input("Escriu un nombre enter per calcular totes les potencies de dos menors que aquest nombre i que no comencen per 2 ")
pot = 1;
while (pot < x):
    if not primer_digit(pot):
        print pot
    pot *= 2
```

**def text():**

```
file = open("text.txt", "r")
text = file.readlines()
file.close()
maxim = 0
temp = 0
paraula = ""
paraulamax = ""
for line in text:
    for i in range(0, len(line)):
        if line[i] == " " or line[i] == "" or line[i] == "." or line[i] == ",":
            if temp > maxim:
                maxim = temp
                paraulamax = paraula
            temp = 0
            paraula = ""
        else:
            paraula += line[i]
            if temp == 0:
                temp += ord(line[i])
            else:
                temp *= ord(line[i])
    if temp > maxim:
        maxim = temp
print "La paraula amb mes valor del text es:", paraulamax
```

```
def digits(n):  
    cont = 0  
    while (n > 0):  
        if n%10 == 1:  
            cont += 1  
        n /= 10  
    return cont
```

```
def valor():  
    x = 2  
    cont = 1  
    acaba = 0  
    while (not acaba):  
        cont += digits(x)  
        if cont == x:  
            acaba = 1  
        x += 1  
    print x
```