



Taki Academy
www.takiacademy.com

Algo & Prog

Classe : Bac Sciences de l'informatique

Série : Bac Pratique 2009 et 2011

Nom du Prof : Mr. Anis ELBEHI

📍 Sousse (Khezama - Sahloul) Nabeul / Sfax / Bardo / Menzah El Aouina /
Ezzahra / CUN / Bizerte / Gafsa / Kairouan / Medenine / Kébili / Monastir /
Gabes / Djerba / Jendouba / Sidi Bouzid / Siliana / Béja / Zaghouan



www.takiacademy.com



73.832.000



- Séance n° : 8
- Type : Pratique
- Objectif(s) :
 - Décomposer un nombre en facteurs premiers
 - Corriger bac pratique (21/05/2009 – 14h30)
 - Corriger Bac pratique 2011 (26/05/2011- 10h.30)

Rappel

Remarque

- Un fichier = une structure de données pouvant garder les données de façon **permanente**. On distingue deux types de fichiers : les fichiers texte et les fichiers de données (binaires)
- Un fichier possède deux noms : un nom logique et un nom physique
- Chaque fichier doit être **enregistré** sur un support de stockage physique (disque dur, flash, ...) et peut contenir une grande quantité d'informations.

Remarque

un fichier est comme un livre . il faut **l'ouvrir** pour **lire** , on peut prendre des notes (**écrire**) et il faut le **fermer** à la fin



Les fichiers textes

Algorithmique	Rôle
Ouvrir ("Chemin\Nom_logique", "Mode")	Ouverture d'un fichier <ul style="list-style-type: none"> • Mode d'ouverture : <ul style="list-style-type: none"> ○ "r" : Lecture ○ "w" : Ecriture (création) ○ "a" : Ajout à la fin du fichier
Lire (Nom_logique, ch)	Lecture de la totalité d'un fichier
Lire_ligne (Nom_logique, ch)	Lecture d'une ligne depuis un fichier texte
Ecrire (Nom_logique, ch)	Écriture de la chaîne ch dans un fichier texte
Ecrire_nl (Nom_logique, ch)	Écriture de la chaîne ch dans un fichier texte et retour à une nouvelle ligne
Fin_fichier (Nom_logique)	Retourne Vrai si le pointeur est à la fin du fichier sinon elle retourne Faux
Fermer (Nom_logique)	Fermeture d'un fichier

Les fichiers textes

En algorithmique	En Python
Ouvrir ("Chemin\Nom_physique" , Nom_logique , "Mode") Avec mode d'ouverture égal à : <ul style="list-style-type: none"> ○ "r" : Lecture ○ "w" : Ecriture (création) ○ "a" : Ajout à la fin du fichier 	Nom_logique = open ('Chemin\Nom_physique' , 'Mode')
Lire (Nom_logique , ch)	ch = Nom_logique. read()
Lire_ligne (Nom_logique , ch)	ch = Nom_logique. readline()
Ecrire (Nom_logique , ch)	Nom_logique. write (ch)
Ecrire_nl (Nom_logique , ch)	Nom_logique. write (ch + "\n")
Fin_fichier (Nom_logique)	ch= Nom_logique. readline() While ch != "" : Traitement ch = Nom_logique. readline() N.B. : La fin d'un fichier texte est la chaine vide
Fermer (Nom_logique)	Nom_logique. close ()

Exercice 1

⌚ 50 min



REPUBLIQUE TUNISIENNE MINISTERE DE L'EDUCATION ET DE LA FORMATION *** Examen du baccalauréat SESSION 2009	Section : Sciences de l'informatique
	Epreuve pratique Matière : Algorithmique et programmation
	Le Jeudi , 21 mai 2009 à 14h 30 Durée : 1H 30 Coef. : 0.75

Soit p et q deux entiers naturels tels que $10 < p < q < 100000$. On se propose d'enregistrer dans un fichier texte **decomp.txt** placé sur la racine du lecteur C la décomposition en facteurs premiers de tous les entiers compris entre p et q . La première ligne du fichier **decomp.txt** contiendra le naturel p suivi d'une espace, suivie du naturel q . Chacune des autres lignes contiendra l'entier à décomposer suivi du signe égal suivi de sa décomposition.

exemple

Si $p=34$ et $q=36$ le fichier **decomp.txt** contiendra les informations suivantes :

34 36

34=2 .17

35=5.7

36=2 .2.3.3

Ecrire un programme **Decomp.py** qui permet d'afficher à l'écran la décomposition en facteurs premiers de tous les entiers compris entre p et q et de remplir le fichier **decomp.txt** comme décrit précédemment.

```

1  def saisir():
2      global p,q
3      p=int(input('p='))
4      q=int(input('q='))
5      while not(10<p<q<100000):
6          p=int(input('p='))
7          q=int(input('q='))
8
9  def decomp(x):
10     ch=str(x)+"="
11     i=2
12     while x!=1:
13         if x%i==0:
14             ch=ch+str(i)+"."
15             x=x//i
16         else:
17             i=i+1
18     return ch[0:len(ch)-1]
19
20 def remplir():
21     F=open("decomp.txt","w")
22     F.write(str(p)+" "+str(q)+"\n")
23     for i in range(p,q+1):
24         F.write(decomp(i)+"\n")
25         print(decomp(i))
26     F.close()
27
28 #pp_Anis.ELBEHI
29 saisir()
30 remplir()

```


Exercice 2

⌚ 40 min



REPUBLIQUE TUNISIENNE
MINISTRE DE L'EDUCATION

EXAMEN DU BACCALAUREAT --- SESSION DE JUIN 2011

SECTION : Sciences de l'Informatique

EPREUVE PRATIQUE : Algorithmique & Programmation

DUREE : 1h 30' COEFFICIENT : 0.75

Anis ELBAHI
Informatique



Important :

- 1) Dans le dossier Bac2022 de votre poste, créez un dossier de travail en lui donnant comme nom votre numéro d'inscription (6 chiffres) et dans lequel vous devez enregistrer au fur et à mesure tous les fichiers solution au problème posé.
- 2) Vérifiez à la fin de l'épreuve que tous les fichiers que vous avez créés sont dans votre dossier de travail.
- 3) Une solution modulaire au problème posé est exigée.

Un nombre Nb et dit **rigolo** si la somme de ses chiffres est égale à la somme de tous les chiffres de ses facteurs premiers.

Exemple 1 :

Si Nb = 690, alors Nb est un nombre **rigolo**. En effet :

- La somme des chiffres de 690 est : $6 + 9 + 0 = 15$
- Les facteurs premiers de 690 sont : 2, 3, 5 et 23
- La somme des chiffres des facteurs premiers de 690 est : $2 + 3 + 5 + 2 + 3 = 15$



Exemple 2 :

Si Nb = 120, alors Nb n'est pas un nombre **rigolo**. En effet :

- La somme des chiffres de 120 est : $1 + 2 + 0 = 3$
- Les facteurs premiers de 120 sont : 2, 2, 2, 3 et 5
- La somme des chiffres des facteurs premiers de 120 est : $2 + 2 + 2 + 3 + 5 = 10$

Travail demandé :

Ecrire un programme Python qui permet de :

- Chercher tous les nombres rigolos dans l'intervalle [M, N] avec M et N, 2 entiers saisis tels que : $100 < M < N < 1000$.
- Stocker les résultats trouvés dans un fichier texte nommé « **resultat.txt** » tel que : Chaque ligne du fichier contient un nombre rigolo suivi du symbole « = », suivi de ses facteurs premiers séparés par une étoile (« * », symbolisant la multiplication). Pour l'exemple précédent, le nombre 690 sera stocké de la manière suivante : $690=2*3*5*23$
- Afficher le contenu du fichier « **resultat.txt** ». Si ce fichier est vide, le programme affichera le message « Il n'y a aucun nombre rigolo ».

Grille d'évaluation

Questions	Barème
- Décomposition du problème en modules utiles à la solution	2
- Appels des modules	2
Si exécution et tests réussis avec respect des contraintes	16
Sinon	
- Compilation d'une solution en adéquation avec le problème posé	3
- Structures de données adéquates au problème posé	3
- Saisie de M et N avec respect des contraintes	2
- Fichier « resultat.txt »	
• Création	1
• Remplissage	
▪ Calcul de la somme des chiffres du nombre	1
▪ Décomposition en facteurs premiers	2
▪ Calcul de la somme des chiffres des facteurs premiers	1
▪ Stockage du nombre	1
• Affichage	2

bp2011.py ×

```

1  def saisir():
2      global m,n
3      m=int(input('m='))
4      n=int(input('n='))
5      while not(100<m<n<1000):
6          m=int(input('m='))
7          n=int(input('n='))
8
9  def decomp(x):
10     ch=""
11     i=2
12     while x!=1:
13         if x%i==0:
14             ch=ch+str(i)+"*"
15             x=x//i
16         else:
17             i=i+1
18     return ch[0:len(ch)-1]
19
20 def somme(x):
21     s=0
22     ch=str(x)
23     for i in range(len(ch)):
24         if "0"<=ch[i]<="9":
25             s=s+int(ch[i])
26     return s
27
28 def rigolo(x):
29     s1=somme(x)
30     s2=somme(decomp(x))
31     return s1==s2
32
33 def remplir(nph):
34     F=open(nph,"w"):
35     for i in range(m,n+1):
36         if rigolo(i):
37             F.write(str(i)+"="+decomp(i)+"\n")
38     F.close()

```

```

39
40 def afficher(nph):
41     F=open(nph,"r")
42     ch=F.readline()
43     ch=ch[0:len(ch)-1]
44     if(ch==""):
45         print("il n'ya aucun nombre rigolo")
46     else:
47         while ch!="":
48             print(ch)
49             ch=F.readline()
50             ch=ch[0:len(ch)-1]
51     F.close()
52
53 #prog principal
54 nph="resultat.txt"
55 saisir()
56 remplir(nph)
57 afficher(nph)

```