UNIVERSITÉ PARIS DIDEROT - PARIS 7
Année 2012-2013, Licence 2, MI4
Groupes et arithmétique

Examen partiel du 23/03/2013 (durée : 2 heures)

Document autorisé : résumé de cours avec annotations personnelles. La calculatrice n'est pas autorisée.

I

1. (a) Rappeler la définition de la fonction indicatrice d'Euler ϕ.
   (b) Calculer $\phi(16)$.
2. Donner la liste des éléments du groupe multiplicatif $(\mathbb{Z}/16\mathbb{Z})^*$ et déterminer l'ordre de chacun d'eux.
3. Le groupe $(\mathbb{Z}/16\mathbb{Z})^*$ est-il cyclique ?

II

Soit G un groupe noté multiplicativement.

1. Si a est un élément de G d'ordre pair : $k = 2l$, quel est l'ordre de $b = a^2$?
2. Si c est un élément de G d'ordre impair : $k = 2l + 1$, quel est l'ordre de $d = c^2$?

III

1. Résoudre le système :
$$\begin{cases} x \equiv 6 \mod 7 \\ x \equiv 2 \mod 6 \end{cases}$$
2. Résoudre le système :
$$\begin{cases} x \equiv 20 \mod 42 \\ x \equiv 16 \mod 20 \end{cases}$$
3. Sept étudiants motivés essaient de se répartir en parts égales un butin composé d'un grand nombre de livres de mathématiques volés. Malheureusement il reste six livres. Il en résulte une bagarre et l'un d'eux est éjecté. Les six étudiants reprennent la répartition et il reste deux livres provoquant une nouvelle dispute et une nouvelle éviction. Les cinq étudiants reprennent la répartition et il reste un livre. Après une dernière éviction, le partage est possible. Quel était le nombre minimal de livres ?

IV

Cet exercice s'appuie sur les annexes A et B. L'annexe A définit les fonctions, et l'annexe B donne quelques résultats.

1. Expliquer brièvement ce que retournent les fonctions Puissance(a,n,n), sqrt(n) et factors(n).
2. Écrire l'algorithme qui calcule la racine carrée entière (fonction sqrt(n)) et justifier le

Annexe A

```
def Puissance(a,m,n): # puissance modulaire rapide
    R=1
    while (m>0):
        if (m%2)==1:
            R=(R*a)%n
        a=(a*a)%n
        m=m/2
    return R

def sqrt(n): # racine carrée entière arrondie par excès
    a,b=0,n
    while b-a>1:
        c=(a+b)/2
        if (c*c)>=n:
            b=c
        else:
            a=c
    return b

def factors(n): # décomposition en facteurs premiers
    d=2
    r=sqrt(n)
    while d<=r:
        if n%d==0:
            return [d]+factors(n/d)
        d=d+1
    return [n]

def lehmer(n,a): # tester n avec le témoin a
    e=n-1
    if Puissance(a,e,n)!=1:
        return 'A'
    l=factors(e)
    for p in l:
        if Puissance(a,e/p,n)==1:
            return 'B'
    return 'C'
```

Annexe B

```
>>> lehmer(561,2)
'B'
>>> lehmer(561,3)
'A'
>>> factors(561)
[3, 11, 17]
>>> lehmer(2147483647,3)
'B'
>>> lehmer(2147483647,7)
'C'
```