

## DS1 – Algorithmique et structures de données – Octobre 2021

### 1. Pile - 6 pts

$$\begin{cases} C(n,p) = C(n-1,p) + C(n-1,p-1) \text{ pour } n>1, p>0 \text{ et } p<n \\ C(n,0) = 1 \\ C(n,n) = 1 \end{cases}$$

1. Écrire en pseudo-langage une fonction récursive qui calcule  $C(n, p)$ .

```

Fonction C(n,p : entier) : entier
Var res : entier
Début
  Si (p=0) ou (p=n)
    Alors res ← 1
    Sinon res ← C(n-1,p){@1} + C(n-1,p-1){@2}
  FinSi
retourner(res)
Fin
  
```

2. Simuler la pile sur l'appel de l'instruction écrire( $C(7,6)\{@0\}$ ) dans le programme principal.  
13 appels et  $C(7,6)=7$

```

@2, n=1 ; p=0 res=1
@1, n=1 ; p=1 res=1
@2, n=2 ; p=1 res=2
@1, n=2 ; p=2 res=1
@2, n=3 ; p=2 res=3
@1, n=3 ; p=3 res=1
@2, n=4 ; p=3 res=4
@1, n=4 ; p=4 res=1
@2, n=5 ; p=4 res=5
@1, n=5 ; p=5 res=1
@2, n=6 ; p=5 res=6
@1, n=6 ; p=6 res=1
@0, n=7 ; p=6 res=7
  
```

### 2. Crible d'Ératosthène - 7 pts

Le crible d'Ératosthène permet de calculer tous les nombres premiers inférieurs ou égaux à un certain entier donné  $N$ . L'algorithme est très simple et procède par élimination : on prend un tableau de 2 à  $N$  booléens qu'on initialise à vrai. Ensuite, il suffit, pour chaque indice du tableau, de mettre à Faux tous ses multiples. Type tab : tableau[2..N] de booléen

2.1.a. Écrire en pseudo-code une procédure itérative qui met à Faux les éléments d'un tableau  $t$  pour chaque indice  $i$  multiple d'un entier donné  $nb$

```

Procédure supprimer_it (E nb : entier ; E/S t : tab)
Var i : entier
Début
Pour i ← nb*2 à N inc +1 faire
  Si i mod nb = 0
    Alors t[i] ← faux
  FinSi
FinPour
Fin
  
```

2.1.b. Ecrire en pseudo-code une procédure itérative qui réalise le crible d'Eratosthène (le tableau  $t$  est initialisé à vrai en dehors de cette procédure).

```

Procédure crible_it (E/S  $t$  : tab)
Var  $i$  : entier
Début
  Pour  $i \leftarrow 2$  à  $N$  inc +1 Faire
    supprimer_it( $i, t$ )
  FinPour
Fin

```

2.2.a. Ecrire en pseudo-code la version récursive de la procédure `supprimer_it` écrite en 2.1.a

```

Appel de supprimer_rec(nb, 2*nb, t)
Procédure supprimer_rec(E  $nb, i$  : entier ; E/S  $t$  : tab)
Début
  Si ( $i \leq N$ )
    Alors Si ( $i \bmod nb = 0$ )
      Alors  $t[i] \leftarrow$  faux
    FinSi
    supprimer_rec( $nb, i+1, t$ )
  FinSi
Fin

```

2.2.b. Ecrire en pseudo-code la version récursive de la procédure `crible_it` écrite en 2.1.b

```

Appel de crible_rec(t, 2)
Procédure crible_rec(E/S  $t$  : tab ; E  $i$  : entier)
Début
  Si ( $i \leq N$ )
    Alors supprimer_rec( $i, i*2, t$ )
    crible_rec( $t, i+1$ )
  FinSi
Fin

```

### 3. Fichier texte - 7 pts

Le fichier intitulé 'queneau.txt' (au verso) contient 14 strophes de 4 vers. Nous souhaitons créer un nouveau poème de 14 vers en ayant choisi aléatoirement un vers dans chaque strophe.

3.1. Expliquer en français la méthode utilisée. Le fichier n'est lu qu'une seule fois et on pourra utiliser une structure de données pour mémoriser les vers.

Il faut lire le fichier en entrée ligne par ligne et mémoriser chaque ligne dans un tableau de chaînes. Ensuite, on utilise ce tableau de chaînes pour créer le nouveau fichier. On tire une valeur entre 0 et 4 qu'on ajoute aux multiples successifs de 5 (en commençant à 1). Ce nombre nous donne l'indice du vers choisit qu'on écrit dans le fichier de sortie. On itère ce processus jusqu'à la fin du tableau (on s'arrête à taille effective du tableau -3 pour la dernière strophe).

3.2. Ecrire en pseudo-code une procédure qui crée un fichier texte contenant un nouveau poème. Cette procédure prendra en entrée le fichier 'queneau.txt' et créera un nouveau fichier texte. Elle pourra appeler d'autres fonctions ou procédures.

Procédure créer\_poème(E nomfich : Chaîne)

Var fe,fs : FT

    i,n,j : entier

    t : tableau[1..200] de Chaîne

Début

fe←ouvrirEnLecture(nomfich)

{On charge le fichier dans un tableau de Chaines}

i←0

TantQue non finFichier(fe) Faire

    i←i+1

    t[i] ← lireChaîne(fe)

FinSi

FinTantQue

fermer(fe)

{On crée un nouveau fichier qu'on remplit grâce au tableau en tirant aléatoirement un vers par strophe}

    fs ← créerFichier('Poème.txt')

    fs ← ouvrirEnEcriture('Poème.txt')

Pour j←1 à i-3 inc +5 Faire

        n←random(4)

        fs←écrireChaîne(t[j+n])

FinPour

fermer(fs)

Fin