

## EPREUVE OPTIONNELLE d'INFORMATIQUE

### A) QCM

Question 1 : Lequel n'est pas un système d'exploitation ?

- A – Windows XP
- B – Unix
- C – Linux
- D – Winsocks

Question 2 : Un octet est codé sur :

- A – 7bits
- B – 8 bits
- C – 16 bits
- D – 32 bits

Question 3 : ADSL est une technologie qui permet de :

- A – superviser un réseau
- B – transporter de la voix et des données
- C – transporter uniquement des données à haut débit
- D – transporter uniquement de la voix à haut débit

Question 4 : Internet est un réseau :

- A – privé
- B – spécialisé
- C – local
- D – d'interconnexion

Question 5 : 195.100.10.20 est une adresse réseau :

- A – IP (Internet)
- B – X25 (Transpac)
- C – MAC (carte réseau)
- D – Téléphonique

Question 6 : DHCP est un serveur de :

- A – noms
- B – fichiers
- C – adresses
- D – messagerie

Question 7 : Dans le codage EBCDIC, les symboles sont codés sur :

- A – 8 bits
- B – 16 bits

- C – 32 bits
- D – 64 bits

Question 8 : DNS est un serveur de :

- A – fichier
- B – noms
- C – adresses
- D – messagerie

Question 9 : ASP est :

- A – est une base de données
- B – un serveur d'application
- C – un langage de programmation
- D – un système d'exploitation

Question 10 : La fonction principale d'un modem est :

- A – transformer le codage ASCII en EBCDIC
- B – protéger l'ordinateur d'une chute de tension
- C – adapter le signal de la ligne à celui de l'ordinateur
- D – adapter le signal de l'ordinateur à celui de la ligne

Question 11 : Dans les équipements informatiques les données sont représentées sous forme :

- A – décimale
- B – hexadécimale
- C – binaire
- D – alphabétique

Question 12 : Le bit de parité permet de :

- A – filtrer les données
- B – compresser les données
- C – détecter les erreurs
- D – corriger les erreurs

Question 13 : En quel langage est codée l'instruction suivante :

let foo a = if a = 0 then 1 else a\*5

- A – MAPPLE
- B – CAML
- C – C
- D – C++

Question 14 : Sur combien d'octets faut-il coder la valeur 300 :

- A – un
- B – deux
- C – trois
- D – quatre

Question 15 : Une variable booléenne peut prendre comme valeur :

- A – supérieure
- B – inférieure
- C – vrai
- D – égale

Question 16 : Quelle est la valeur de X ?

```
    Pour i = 1 à 5
        X = 5
        X = X + i
    Fin
```

- A – 1
- B – 5
- C – 10
- D – 25

Question 17 : Dans l'unité centrale d'un ordinateur, les composants communiquent à l'aide de :

- A – registres
- B – pointeurs
- C – bus
- D – tubes

Question 18 : Que fait cette instruction ?

```
int *pointeur
```

- A – indique une remarque
- B – signale un commentaire
- C – calcule une puissance
- D – signale un pointeur sur un entier

Question 19 : A quoi correspond la technologie WIFI ?

- A – transmission par satellite
- B – transmission sans fil
- C – vidéo haute définition
- D – ordinateur nouvelle génération

Question 20 : Parmi les instructions suivantes, laquelle est incorrecte ?

B : booléen      E : entier      R : entier

- A –  $R = E \times 2$
- B –  $E = R + 10$
- C – si B est vrai alors  $R = 0$
- D – sinon  $B = B + 10$

Question 21 : TCP/IP est une architecture réseau de :

- A – union internationale des télécommunications (UIT-T)
- B – BULL

- C – Microsoft
- D – Département de la Défense Américaine (DOD)

Question 22 : let rec x n = match n with

```

| 0 -> 1
| n -> x(n - 1) + y(n - 1)
and y n = match n with
| 0 -> 2
| 0 -> x(n - 1) * y(n - 1);;

```

"x4;;"

Que renvoie cette fonction ?

- A – 40
- B – 41
- C – 42
- D – 43

Question 23 : Laquelle de ces définitions n'est pas correcte ?

- A – let rec fact a = if a = 0 then 1 else a \* fact (a-1)
- B – let xor = function 

```

| (false,false) -> false
| (false,true) -> true
| (true,false) -> true
| (true,true) -> false;;

```
- C – let est\_chiffre = function 

```

| `0` | `1` | `2` | `3` | `4` | `5` | `6` | `7` | `8` | `9` -> true
| _ -> false;;

```
- D – let somme x = if x = 0 then x+somme (x-1);;

Question 24 : Donnez le type de l'expression suivante :

```
let h x = match x with (0,_) -> false | (n,y) -> n > y;;
```

- A – h : 'a \* 'a -> bool
- B – h : int \* int -> bool
- C – h : int -> int -> int
- D – h : 'a -> bool

Question 25 : let rec p x n = if x = 0 then 1

```

else if x mod 2 = 0 then p (x/2) (n*n)
else n * p (x/2) (n*n) ;;

```

Que calcule cette fonction ?

- A – racine carrée de x puissance n
- B – x puissance n
- C – (x/2) puissance n
- D – n puissance x

## **B) ALGORITHMIQUE ET PROGRAMMATION**

### **REMARQUES :**

*Les Programmes sont à écrire en PASCAL, en C ou en CAML. Pour l'accès aux fichiers en entrée et en sortie, un rappel syntaxique des instructions sur les fichiers « texte » est fourni en annexe.*

*Le principe de l'algorithme retenu doit être exprimé avant chaque programme.*

### **PARTIE I : SOUS-ENSEMBLE**

Le but est de trouver parmi un ensemble de valeurs entières le plus grand sous-ensemble dont la somme des éléments est divisible par 3. Ecrire un programme qui détermine la taille de ce sous-ensemble (le nombre d'entiers dont il est constitué).

Votre programme acceptera en entrée dans un fichier texte (ENSEMBLE.IN) une séquence de nombres entiers (un par ligne) tels que le premier représente le nombre d'entiers à traiter (entre 1 et 10000) et qu'ensuite sur chaque ligne il y ait un nombre (entre 0 et 20000) représentant la valeur de chaque entier.

Il restituera en sortie dans un fichier texte (ENSEMBLE.OUT), un nombre entier représentant la taille du plus grand sous-ensemble de celui d'origine (donné par ENSEMBLE.IN) dont la somme des éléments est divisible par 3.

### **EXEMPLE :**

#### **ENSEMBLE.IN :**

```
7
10
6
7
12
4
7
22
```

#### **ENSEMBLE.OUT :**

```
5
```

### **PARTIE II : MAGIQUE**

Supposons  $d$  un chiffre et  $n$  un nombre entier positif. Une suite de nombres  $t_1, t_2, \dots, t_m$  est une  $d$ -somme si :

- $t_1 + t_2 + \dots + t_m = n$ ,

et

- $t_1, t_2, \dots, t_m$  sont des nombres uniquement composés du chiffre  $d$ .

Par exemple, si  $d=2$  alors  $t_i$  peut être égal à 2, 22, 222, etc.

Ecrire un programme qui pour un chiffre  $d$  et un nombre  $n$  donnés, détermine si le nombre  $n$  est une  $d$ -somme du chiffre  $d$ . Votre programme acceptera en entrée dans un fichier texte (MAGIQUE.IN) une séquence de nombres entiers (deux par ligne) tels que le premier représente le chiffre  $d$  et le deuxième le nombre entier positif  $n$ . Pour chaque ligne du fichier d'entrée (MAGIQUE.IN), il restituera en sortie dans un fichier texte (MAGIQUE.OUT), trois lignes, la première rappellera  $d$ , la deuxième rappellera  $n$  ainsi que son éventuelle  $d$ -somme et enfin la troisième précisera, si la  $d$ -somme existe, le nombre  $m$  de termes qui la constituent et sinon le fait qu'il n'y a pas de sommation possible.

**EXEMPLE :**

**MAGIQUE.IN :**

```
3 402
5 64
1 100
9 9
```

**MAGIQUE.OUT :**

```
d = 3
n = 402 = 333 + 33 + 33 + 3
La plus petite 3-Somme est en 4 termes.
```

```
d = 5
n = 64
Il n'y a pas de 5-Somme possible pour 64.
```

```
d = 1
n = 100 = 11 + 11 + 11 + 11 + 11 + 11 + 11 + 11 + 11 + 1
La plus petite 1-Somme est en 10 termes.
```

```
d = 9
n = 9 = 9
La plus petite 9-Somme est en 1 terme.
```

# ANNEXE

## LANGAGE PASCAL :

- ◇ **Procedure Assign(var F ; Nom\_fic : string) ;**  
Affecte le nom d'un fichier disque à une variable-fichier.
- ◇ **Procedure Reset(var f [ :FILE ;TailleEnr : Word]) ;**  
Ouvre un fichier disque existant.
- ◇ **Procedure Rewrite(var f [ :FILE ; TailleEnr : Word]) ;**  
Crée et ouvre un fichier.
- ◇ **Procedure Read( var f : TEXT ; v1 [, v2, .. , vN]) ;**  
Lit une ou plusieurs valeurs dans une ou plusieurs variables.
- ◇ **Procedure Readln( var f : TEXT ; v1 [, v2, .. , vN]) ;**  
Lit une ou plusieurs valeurs dans une ou plusieurs variables jusqu'à la fin de ligne.
- ◇ **Procedure Write( var f : TEXT ; v1 [, v2, .. , vN]) ;**  
Ecrit une ou plusieurs valeurs dans le fichier.
- ◇ **Procedure Writeln( var f : TEXT ; v1 [, v2, .. , vN]) ;**  
Ecrit une ou plusieurs valeurs dans le fichier plus un retour ligne.
- ◇ **Procedure Close( var f) ;**  
Ferme un fichier ouvert.

## LANGAGE C :

- ◇ **FILE \*fopen( char \*path, char \*mode) ;**  
Ouvre le fichier dont le nom est contenu dans la chaîne pointée par « path » et lui associe un flux. L'argument « mode » pointe vers une chaîne commençant par l'une des séquences suivantes :
  - r fichier existant ouvert en lecture
  - r+ fichier existant ouvert en lecture/écriture
  - w crée un fichier et l'ouvre en écriture
  - w+ crée un fichier et l'ouvre en lecture/écriture
- ◇ **int fscanf( FILE \*stream, const char \*format, ... ) ;**  
Lit les données depuis un flux pointé par « stream », convertit ces données selon le « format » décrit et stocke le résultat des conversions dans des arguments pointeurs.
- ◇ **int fprintf( FILE \*stream, const char \*format, ... ) ;**  
Ecrit les données sur le flux « stream » indiqué, ces données étant préalablement converties selon le « format » décrit.
- ◇ **int fclose( FILE \*stream) ;**  
Dissocie le flux nommé « stream » du fichier sous-jacent.

## LANGAGE CAML :

- ◇ **val openfile : string -> open\_flag list -> file\_perm -> file\_descr**  
Ouvre un fichier dont le premier argument est le nom, le second la liste des modes d'ouverture, le troisième les permissions, et renvoie un descripteur de fichier.

Le type "open\_flag" est défini ainsi :

```
type open_flag =  
  | O_RDONLY  
  (* Ouverture pour lecture seulement *)  
  | O_WRONLY  
  (* Ouverture pour écriture seulement *)  
  | O_RDWR  
  (* Ouverture pour lecture et écriture *)
```

*Remarque : il existe d'autres modes, mais ils n'ont pas d'intérêt dans le cas présent.*

- ◇ **val close : file\_descr -> unit**  
Ferme un fichier à partir d'un descripteur.
- ◇ **val read : file\_descr -> string -> int -> int -> int**  
"read fd buff ofs len" lit len caractères du fichier dont le descripteur est fd, stocke ces caractères dans buff à partir de la position ofs et renvoie le nombre d'octets lus.
- ◇ **val write : file\_descr -> string -> int -> int -> int**  
"write fd buff ofs len" extrait len caractères contenus dans buff à partir de la position ofs et les écrit dans le fichier dont le descripteur est fd.