

## EPREUVE OPTIONNELLE d'INFORMATIQUE

### A) QCM

Les réponses au QCM doivent être portées directement sur la feuille de sujet de QCM.

Ne pas omettre de faire figurer votre numéro de candidat sur cette feuille et de la joindre à vos autres copies d'examen.

Remarque : - A une question correspond au moins 1 réponse juste  
- Cocher la ou les bonnes réponses

Barème : - Une bonne réponse = +1  
- Pas de réponse = 0  
- Une mauvaise réponse = -1

---

Question 1 : Un paquet ou trame comprend :

- A - uniquement des données utilisateur
- B - un en-tête et des données
- C - des unités de transmission de taille fixe
- D - uniquement des informations d'adressage et de contrôle

Question 2 : Les principales ressources partagées au sein d'un groupe de travail comprennent les fichiers, les imprimantes et :

- A - l'information sur la conception du réseau
- B - les outils de maintenance
- C - les systèmes d'exploitation clients
- D - les services de communication

Question 3 : Un câble de catégorie 5 fait référence à :

- A - un câble coaxial
- B - la meilleure paire torsadée non blindée
- C - la fibre optique monomode
- D - la paire téléphonique ordinaire

Question 4 : IPX de Novell diffère de IP en ce sens que IPX ne supporte pas :

- A - des adresses de la forme <réseau> <hôte>
- B - la fragmentation de paquets
- C - le routage de paquets
- D - le service en mode non connecté

Question 5 : Les systèmes Clients et Serveurs de Windows NT supportent :

- A - uniquement les protocoles TCP/IP
- B - les protocoles TCP/IP, Novell et LLC
- C - uniquement les protocoles Novell
- D - les protocoles TCP/IP, Novell, Decnet, SNA

Question 6 : L'émulation d'un équipement peut se faire en sorte qu'un PC :

- A - ressemble et se comporte comme l'équipement émulé
- B - communique avec un grand système en utilisant NETBIOS
- C - transporte de l'information sur des liaisons de grande distance de manière plus performante
- D - se comporte comme un grand système

Question 7 : La représentation suivante « 193. 16. 8. 2 » correspond à :

- A - une adresse MAC
- B - un numéro de port
- C - une adresse IP
- D - un numéro de SAP

Question 8 : Le système Unix est un système :

- A - multi-tâches
- B - mono-tâches
- C - multi-utilisateurs
- D - mono-utilisateurs

Question 9 : Les instructions suivantes sont rédigées en langage :

```
main {  
  char c1 [1000], c2 [1000]  
  int i, c ;  
  i = 0  
  while ((c =getchar() ) != | \ n)  
    {      c[i] = c ;  
      if ( - - -  
    } ;  
  printf ( ^ % f \ n - - - - - ) ;  
}
```

- A - Pascal
- B - Cobol
- C - Scheme
- D - C

Question 10 : L'informatisation consistant à répartir les traitements entre un poste de travail et un serveur est une architecture de type :

- A - systèmes répartis
- B - client / serveur
- C - systèmes distribués
- D - interrogation de bases de données relationnelles

Question 11 : Le contrôle de parité est une technique:

- A - de détection d'erreur
- B - de compression de données
- C - de cryptage
- D - de multiplexage

Question 12 : Les différents éléments d'un ordinateur (mémoire, processeurs, périphériques,...) sont reliés par :

- A - des registres
- B - des pointeurs
- C - le système d'exploitation
- D - des bus

Question 13 : Parmi les mémoires suivantes quelles sont celles qui sont volatiles ? :

- A - RAM
- B - ROM
- C - EPROM
- D - mémoire cache

Question 14 : Quel est le langage de manipulation des bases de données ? :

- A - C
- B - HTML
- C - SQL
- D - JAVA

Question 15 : Dans les équipements informatiques les données sont représentées par un signal électrique de la forme :

- A - analogique
- B - numérique
- C - alphanumérique
- D - alphabétique

Question 16 : Cette suite d'instructions est décrite en langage :

```
Var res : real ;  
begin  
    readln(res) ;  
    writeln(res/2) ;  
end
```

- A - Pascal
- B - C
- C - COBOL
- D - Scheme

Question 17 : Que fait le programme précédent (question 16) ? :

- A - affiche « res »
- B - lit « ln »
- C - lit un nombre au clavier et affiche un autre nombre
- D - calcule la factorielle

Question 18 : Quel est le débit maximum autorisé sur une ligne téléphonique à l'aide d'un modem ? :

- A - 28800 b/s
- B - 64 Kb/s
- C - 9600 b/s
- D - 56 Kb/s

Question 19 : En langage « C » un « long » mesure :

- A - 2 octets
- B - 4 octets
- C - 8 octets
- D - variable

Question 20 : Int \*pointeur :

- A - décrit un pointeur sur un entier
- B - signale une remarque sur « pointeur »
- C - déclare un tableau
- D - rappelle une commande

Question 21 : « IP » (Internet Protocol) décrit :

- A - une messagerie électronique
- B - un transfert de fichier
- C - un protocole de communication
- D - le réseau Ethernet

Question 22 : Quelle est la spécialisation qui n'est pas assurée par l'EPITA :

- A - système et réseau
- B - télécommunication
- C - multimédia
- D - hardware

Question 23 : Le bus micro EISA (Extended Industry Standard Architecture) utilise une longueur de données de :

- A - 8 bits
- B - 16 bits
- C - 32 bits
- D - 64 bits

Question 24 : Un bus d'adresses sur 16 bits représente une capacité d'adressage maximale de :

- A -  $16^2$
- B -  $2^{16}$
- C -  $8^2$
- C -  $16^8$

\* \* \* \*

### **Préliminaire au B) et C) :**

Les algorithmes peuvent être fournis en français à condition d'offrir une excellente lisibilité, ou bien fournis selon une syntaxe que vous préciserez.

Les Programmes sont à écrire soit en PASCAL, soit en C. 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.

## **B) ECHELLES ET TOBOGGANS**

Il existe un jeu appelé « Echelles et toboggans ». Il se joue sur un plateau constitué de cases numérotées de 1 à 100, les joueurs démarrant d'une case que l'on pourrait noter 0. A tour de rôle, ils jettent un dé à six faces et avancent du nombre de cases correspondant au jet de dé (la case à atteindre est obtenue par addition de la valeur du dé et de la case sur laquelle ils se trouvent). Le premier à atteindre la case 100 à gagnée.

L'intérêt du jeu réside dans l'existence d'échelles et de toboggans reliant des cases entre elles et permettant de progresser ou de régresser très vite. Si, par exemple, le jet d'un joueur l'amène dans une case contenant la base d'une échelle, il se retrouve automatiquement dans la case correspondant au sommet de celle-ci. Il en va de même pour un toboggan. Toutefois, il est à noter que le fait de tomber sur une case représentant le sommet d'une échelle ou l'arrivée d'un toboggan n'a aucun effet.

D'autre-part, il existe des cases qui vous font passer le tour suivant (« passe-le-tour ») ou vous permettent de rejouer immédiatement (« rejoue-immédiatement »). Ce type de cases, ne peut jamais se trouver à l'extrémité d'une échelle ou d'un toboggan.

Enfin, si un joueur se trouve sur une case 95 ou supérieure, un jet de dé le propulsant au delà de la case 100 sera purement ignoré ( un joueur situé sur la case 99 ignorera tous les jets autres que le 1).

Ecrire un programme qui simule ce jeu. Celui-ci doit lire les données dans un fichier texte (ECHELLE.IN) structuré de la façon suivante :

- ❑ D'abord, une séquence de jets de dés (1000 maximum) utilisée par tous les joueurs. Le premier utilise le premier jet, le second utilise le second jet, et ainsi de suite. Cette séquence est constituée de chiffres de 1 à 6 séparés par un simple espace et formant des lignes de 80 caractères maximum. Elle est terminée par un 0.
- ❑ Ensuite, une ou plusieurs descriptions de parties. Chacune d'elles étant découpée selon les deux phases suivantes :
  - ◆ La première est une ligne contenant un simple chiffre déterminant le nombre de joueurs (de 2 à 5).
  - ◆ La deuxième, séparée en deux, décrit le plateau de jeu.
    - Une description des échelles et des toboggans. Chacun d'eux est déclaré, sur une seule ligne, par deux entiers ( de 1 à 99 ) séparés l'un de l'autre par un espace. Le premier nombre donne la case de départ et le deuxième donne la case d'arrivée. Le premier nombre est inférieur au deuxième pour une échelle et inversement pour un toboggan. La déclaration de ces éléments est terminée par une ligne contenant deux zéros séparés d'un espace.
    - Une description des cases spéciales (passe-le-tour et rejoue-immédiatement) s'il y en a. Dans ce cas, il y a un nombre par ligne désignant une case. S'il est négatif, c'est une case « passe-le-tour », s'il est positif, c'est une case « rejoue-immédiatement ». Par exemple : -32 indique que la case 32 est une case « passe-le-tour » ; a l'inverse, 42 indique que la case 42 est une case « rejoue-immédiatement ».

La fin de la description d'une partie est déterminée par une ligne contenant un zéro.  
La fin de toute description de partie est donnée par un nombre de joueurs égal à zéro.

Les résultats doivent être écrits dans un fichier texte (ECHELLE.OUT) selon le format suivant : Une ligne par partie, celle-ci donnant le numéro du joueur gagnant. Chaque partie donnée en entrée permettant de déterminer, à coup sûr, un gagnant, il n'est donc pas nécessaire d'envisager une gestion de partie nulle (sans vainqueur).

**EXEMPLE :**

**ECHELLE.IN :**

```
3 6 3 2 5 1 3 4 2 3 1 2 0
2
6 95
99 1
0 0
-3
98
0
2
3 99
6 90
0 0
0
0
```

**ECHELLE.OUT :**

```
2
2
```

## C) IMMEDIATEMENT DECODABLE

Un ensemble de symboles codés est dit « immédiatement décodable » si aucun code des symboles ne sert de préfixe à un autre code du même ensemble.

Nous admettrons pour ce problème que tous les codes sont en binaire, qu'à l'intérieur d'un même ensemble de codes, il ne peut pas y en avoir deux identiques, que chaque code à au moins un bit et au plus dix, et que chaque ensemble de codes possède au moins deux codes et au plus huit.

Par exemple : Si l'on considère un alphabet constitué des symboles {A,B,C,D} ;

Le code suivant est « immédiatement décodable »

A:01 B:10 C:0010 D:0000

Mais celui-ci ne l'est pas :

A:01 B:10 C:010 D:0000 (A sert de préfixe à C)

Ecrire un programme qui accepte en entrée une série d'ensembles à partir d'un fichier (CODES.IN), qui détermine si ces ensembles sont « immédiatement décodable » et l'écrit dans un fichier de sortie (CODES.OUT).

Chaque ensemble est défini par une suite de zéros et de uns représentant le code d'un symbole de l'ensemble (un code par ligne). Chaque ensemble est séparé des autres par une ligne contenant un 9 (neuf). Il est à noter que le séparateur 9 ne fait pas partie de l'ensemble, que chaque ensemble est indépendant des autres, et que les codes d'un ensemble n'ont aucun rapport avec ceux des autres ensembles. Ce qui sous-entend que chaque ensemble de codes doit être traité séparément).

### EXEMPLE :

#### CODES.IN :

01  
10  
0010  
0000  
9  
01  
10  
010  
0000  
9

#### CODES.OUT :

L'ensemble 1 est immédiatement décodable  
L'ensemble 2 n'est pas immédiatement décodable

## 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.