METEO FRANCE

CONCOURS INTERNE POUR LE RECRUTEMENT

D'ÉLÈVES INGÉNIEURS DES TRAVAUX DE LA MÉTÉOROLOGIE

SESSION 2017

****

ÉPREUVE FACULTATIVE A OPTION :

INFORMATIQUE

Durée : 3 heures Coefficient : 2 (pour les points au-dessus de 10)

La rigueur, le soin et la clarté apportés à la rédaction des réponses seront pris en compte dans la notation. L'utilisation de toute documentation (dictionnaire, support papier, traducteur, téléphone portable, assistant électronique, etc) est strictement interdite.

Cette épreuve comporte six parties indépendantes. Elles peuvent être abordées dans l'ordre de choix des candidats.

Barème envisagé :

- Partie 1 - Structure des calculateurs : 2,5 points
- Partie 2 - Représentation de l'information et structures de données : 2 points
- Partie 3 - Programmation, langages et systèmes d'exploitation : 6,5 points
- Partie 4 - Réseaux de communication : 3 points
- Partie 5 - Calcul parallèle et systèmes distribués : 3 points
- Partie 6 - Systèmes temps réel : 3 points

Cette épreuve comporte 11 pages (celle-ci incluse).

Météo-France, établissement public administratif
Direction des ressources humaines
73, avenue de Paris 94165 Saint-Mandé Cedex

Première partie :
structure des calculateurs

Question 1 (0,5 point) : qu'est-ce que l'algèbre de Boole ?

Question 2 (0,5 point) : combien de lignes a une table de vérité d'une fonction de *n* variables ?

Question 3 (0,5 point) : combien peut-on avoir de fonctions logiques pour *n* variables ? Donnez ce nombre lorsque $n = 3$.

Question 4 (1 point) : quelle est la différence majeure entre une architecture dite de Von Neumann et une architecture dite Harvard ?

Deuxième partie :
représentation de l'information et structures de données

Question 1 (1 point) : donnez les représentations hexadécimale et décimale de la valeur binaire `1010101110111010`.

Question 2 (1 point) : la mise en œuvre d'une structure de données sous la forme d'une liste chaînée dans le cadre d'un traitement distribué sur plusieurs calculateurs différents vous semble-t-elle une bonne option *a priori* (justifiez votre réponse) ?

Troisième partie :
programmation, langages et systèmes d'exploitation

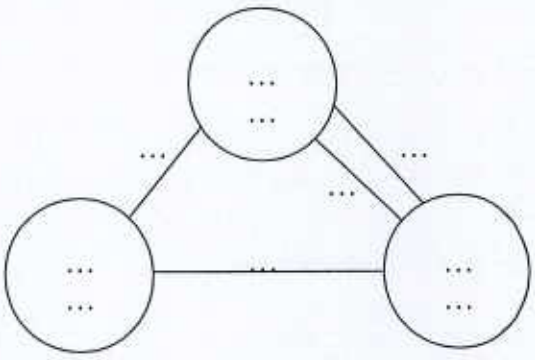
Question 1 (2 points) : citez les grandes étapes du cycle de vie d'un logiciel, pour chacune d'entre-elles décrivez ses caractéristiques principales et finalement montrez à l'aide d'un schéma les relations entre ces étapes.

Question 2 (0,5 point) : présentez (citez et décrivez brièvement) l'une des « métriques » utilisées dans le cadre des revues de code en ingénierie logicielle.

Question 3 (0,5 point) : citez trois objectifs qu'un système de fichiers doit remplir du point de vue de l'utilisateur d'un système d'exploitation.

Question 4 (0,5 point) : de manière interne au système d'exploitation, citez trois caractéristiques que le système de fichiers doit avoir pour remplir les objectifs attendus par un utilisateur.

Question 5 (0,5 point) : après avoir recopié sur votre copie le schéma ci-dessous représentant les trois principaux états dans lesquels un processus peut se trouver lors de son exécution par un système d'exploitation, complétez-le en indiquant le nom habituel de chacun de ces états, et combien de processus peuvent se trouver dans chacun d'eux à un instant donné. Vous noterez également par des flèches le sens de passage d'un état à un autre et ce qui peut provoquer ce passage.



Question 6 (2 points) : donnez les noms des quatre principales phases permettant l'obtention d'un programme exécutable à partir d'un code source écrit dans un langage compilé en indiquant le rôle de chacune d'elles et la nature des fichiers qu'elles manipulent.

Question 7 (0,5 point) : citez deux langages compilés.

Quatrième partie :
réseaux de communication

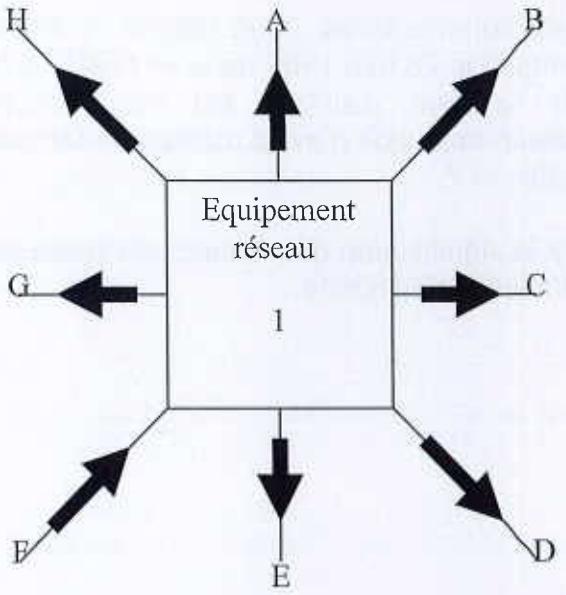
Question 1 (0,5 point) : donnez, sans explication, la liste des couches du modèle OSI (numéros et noms).

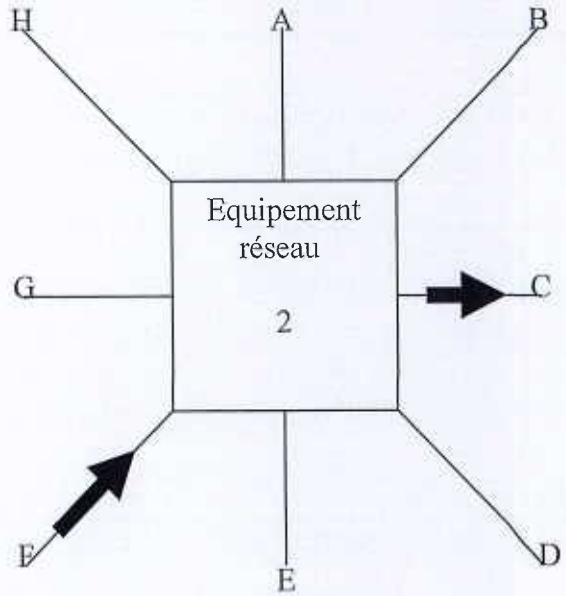
Question 2 (0,5 point) : dessinez trois topologies de réseaux informatiques.

Question 3 (0,5 point) : que sont le poids de Hamming et la distance de Hamming dans le contexte de la protection contre les erreurs de transmission ?

Question 4 (0,5 point) : considérez la mise en œuvre d'une protection contre les erreurs de transmission par un contrôle de parité à parité paire. Que signifie une parité paire ? Combien d'erreurs permet-elle de détecter pour chaque mot transmis ?

Question 5 (1 point) : ci-dessous sont représentés schématiquement deux équipements réseaux (numérotés 1 et 2) avec huit interconnexions vers d'autres équipements (nommés de A à H). Supposez qu'on souhaite faire circuler une information du dispositif estampillé F vers le dispositif estampillé C en passant par l'un et par l'autre des équipements réseaux dessinés ci-dessous. Les flèches représentées sur le dessin montrent la circulation de l'information dans ce cas, circulation due au fonctionnement respectif des deux équipements. Sur cette base, donnez le nom de l'équipement 1 et le nom de l'équipement 2.





Cinquième partie :
calcul parallèle et systèmes distribués

Question 1 (2 points) : quels commentaires vous inspire la définition d'un système distribué donnée par Leslie Lamport le 28 mai 1987 dans un mail interne lorsqu'il travaillait pour la DEC Corp. : « Un système distribué est un système pour lequel le dysfonctionnement d'un ordinateur dont vous n'aviez même pas connaissance peut rendre votre propre ordinateur inutilisable »[1] ?

Question 2 (1 point) : donnez la signification des initiales du protocole NTP et expliquez brièvement à quoi il sert et comment il fonctionne.

1Citation originale en anglais : « *A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable* ».

Sixième partie :
systèmes temps réel

Dans les questions qui suivent, on cherche à améliorer la prévisibilité temporelle de programmes ; c'est-à-dire que nous souhaitons faire en sorte que leur temps d'exécution soit prévisible et le plus constant possible, éventuellement au détriment de l'efficacité (le temps de calcul pourrait être plus long une fois le programme transformé).

Question 1 (0,5 point) : quel est le facteur essentiel, au niveau de sa construction, rendant un programme difficilement prévisible en termes de temps d'exécution ? Vous supposerez que ce programme s'exécute sur une machine dont tous les composants sont connus et caractérisés temporellement, qu'il est seul à s'exécuter (donc que la charge de la machine n'est pas en cause), et qu'elle ne présente pas de dysfonctionnement particulier tout comme le programme lui-même.

Question 2 (0,5 point) : l'extrait de programme ci-dessous écrit en langage C (mais le langage n'a pas d'importance pour la question, et l'extrait utilise suffisamment peu d'instructions, simples au demeurant, pour rester compréhensible et pour que vous puissiez considérer cette partie de programme comme étant écrite en pseudo-code) exhibe une implémentation classique d'un tri à bulle d'une liste d'entiers. Dans cet extrait, qu'est-ce qui fait que ce tri ne peut pas être temporellement prévisible (son temps d'exécution va varier si les données en entrée changent) ?

```
void tri_a_bulle( int * liste, long int n )
{
  long int     longueur, index;
       int     temporaire;

  for (longueur = n; longueur>=2; longueur--)
  {
    for (index = 0; index<longueur-1; index++)
    {
      if (liste[index]>liste[index+1])
      {
        temporaire      = liste[index  ];
        liste[index  ] = liste[index+1];
        liste[index+1] = temporaire;
      }
    }
  }
}
```

Question 3 (1 point) : quelle solution employant une transformation de programme pourriez-vous proposer afin de rendre les programmes comme celui de la question précédente indépendants des données en entrée pour leur temps d'exécution tout en conservant la même fonctionnalité ? Vous pouvez raisonner en vous appuyant sur l'extrait de programme de la question précédente. Aucune implémentation n'est demandée, seule la manière de procéder l'est.

Question 4 (1 point) : supposez que vous souhaitiez calculer la valeur absolue d'un entier signé codé sur 16 bits. Le programme (écrit en langage C) est habituellement le suivant :

```
unsigned short int  valeur_absolue( signed short int     i )
{
  unsigned short int      a = 0;

  if (i>=0)
  {
    a = i;
  }
  else
  {
    a = -i;
  }
  return( a );
}
```

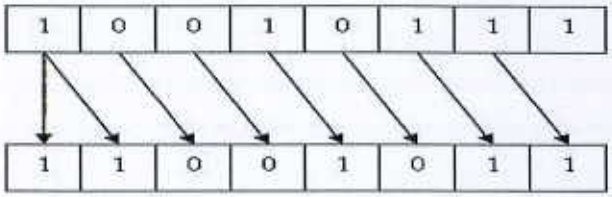
Proposez la séquence d'opérations nécessaire à ce calcul de manière à ce que le temps d'exécution du programme soit constant (non dépendant du signe de l'entier en entrée).

Vous avez à votre disposition pour cela :

- l'opérateur = qui réalise l'affectation de la valeur à sa droite à la variable désignée à sa gauche,
- l'opérateur ^ qui réalise un ou exclusif entre deux entiers,
- l'opérateur | qui réalise un ou inclusif entre deux entiers,
- l'opération – qui soustrait un entier à un autre,
- l'opération + qui additionne un entier à un autre,
- l'opérateur >> qui réalise un décalage arithmétique de bits vers la droite sur l'entier à gauche de l'opérateur du nombre de bits spécifié à droite de l'opérateur,
- l'opérateur << qui réalise un décalage de bits vers la gauche sur l'entier à gauche de l'opérateur du nombre de bits spécifié à droite de l'opérateur.

Pour mémoire :

- le décalage de bits arithmétique à droite produit le résultat suivant :



- le décalage de bits à gauche produit le résultat suivant :