

EPREUVE OPTIONNELLE D'INFORMATIQUE

CORRIGE

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 = fonction
 - | (false,false) -> false
 - | (false,true) -> true
 - | (true,false) -> true
 - | (true,true) -> false;;
- C – let est_chiffre = fonction
 - | `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

REMARQUE :

Les programmes en Pascal, C sont des exemples et peuvent être discutés en terme d'implémentation et de construction. Le choix qui a été fait, est celui d'une découpe fonctionnelle et procédurale importante.

Cela permet de mieux faire ressortir un algorithme principal.

Ensuite chaque tâche est détaillée dans la procédure ou fonction correspondante.

D'autre part, il n'y a pas forcément d'optimisation entre les programmes Pascal et C qui peuvent être de simple traduction les uns des autres.

A) SOUS-ENSEMBLE

Le principe est le suivant :

Les entiers sont lus un par un. A chaque lecture le modulo de l'entier est calculé et sa valeur est additionnée à un ensemble de reste. A la fin, le modulo de ce reste est à son tour calculé et son résultat permet de déterminer si le sous-ensemble le plus grand est de n, (n-1) ou (n-2) éléments. Cette décision dépendant du fait que l'on ait rencontré ou non des entiers dont le modulo était 1 ou 2.

Programme Pascal (Delphi mode console)

```
program Ensemble;
{$APPTYPE CONSOLE}

uses
  SysUtils;

var
  FichierEntree, (* Fichier TEXT pour l'entrée *)
  FichierSortie : Text; (* Fichier TEXT pour la sortie *)

procedure OuvertureFichier;
begin
  Assign(FichierEntree, 'ENSEMBLE.IN');
  Reset(FichierEntree); (* ENSEMBLE.IN ouvert en lecture *)
  Assign(FichierSortie, 'ENSEMBLE.OUT');
  Rewrite(FichierSortie); (* ENSEMBLE.OUT ouvert en écriture *)
end;
```

```

procedure LectureEtTraitement;
var
    n      : integer;           (* nombre d'éléments *)
    r      : longint;          (* somme de tous les éléments modulo 3 *)
    S      : array [0..2] of boolean; (* existence d'éléments modulo 3 *)
    i, x   : integer;
begin
    r := 0;                    (* initialialisation de r et de s *)
    for i := 0 to 2 do
        S[i] := false;
    readln(FichierEntree,n);   (* lecture de n *)
    for i := 1 to n do
        begin
            readln (FichierEntree, x);
            x := x mod 3;
            S[x] := true;
            r := r + x
        end;
    r := r mod 3;
    writeln(FichierSortie,'n = ',n);
    if r = 0
        then writeln (FichierSortie, n)
        else if S[r]
            then writeln (FichierSortie, n - 1)
            else writeln (FichierSortie, n - 2);
end;

procedure FermetureFichier;
begin
    Close(FichierEntree);
    Close(FichierSortie);
end;

begin
    OuvertureFichier;
    LectureEtTraitement;
    FermetureFichier;
end.

```

Programme C (DevC++ 4)

```

#include <stdio.h>
#include <stdlib.h>

#define NBMAXCADEAU 24

FILE *FichierEntree; /* Fichier TEXT pour l'entrée */
FILE *FichierSortie; /* Fichier TEXT pour la sortie */

void OuvertureFichier()
{
    if ((FichierEntree = fopen("ENSEMBLE.IN", "r")) == NULL)
    {
        perror("ENSEMBLE.IN");
        exit(1);
    }
    rewind(FichierEntree);
    /* Ouverture de ENSEMBLE.IN en lecture */
    FichierSortie = fopen("ENSEMBLE.OUT", "w");
    if (FichierSortie != NULL)
        rewind(FichierSortie);
    else
        FichierSortie = tmpfile();
    if (FichierSortie == NULL)
    {
        perror("FichierSortie");
        exit(1);
    }
    /* Ouverture de ENSEMBLE.OUT en écriture */
}

```

```

}

void FermetureFichier()
{
    if (FichierEntree != NULL)
        fclose(FichierEntree);
    FichierEntree = NULL;
    if (FichierSortie != NULL)
        fclose(FichierSortie);
    FichierSortie = NULL;
}

void LectureEtTraitement()
{
    int n,r=0;
    int s[3]={0,0,0};
    int i, x;

    fscanf(FichierEntree,"%D",&n);

    for(i=1;i<=n;i++)
    {
        fscanf(FichierEntree,"%D",&x);
        x=x%3;
        s[x]=1;
        r=r+x;
    }
    r=r%3;
    fprintf(FichierSortie,"n = %d\n",n);
    if (r=0) fprintf(FichierSortie,"%d\n",n);
        else if (!s[r]) fprintf(FichierSortie,"%d\n",n-1);
            else fprintf(FichierSortie,"%d\n",n-2);
}

int main(int argc, char *argv[])
{
    OuvertureFichier();
    LectureEtTraitement();
    FermetureFichier();

    return 0;
}

```

B) MAGIQUE

Le principe est le suivant :

On commence avec une d-valeur la plus proche de l'entier à tester (exemple : si d=3 et n=423, on essaie 333). On soustrait alors ce nombre de n autant de fois que possible, puis on passe à la puissance de 10 inférieure et ainsi de suite jusqu'à l'unité. Si n atteint pas soustraction successive la valeur 0, il est donc possible de l'écrire sous la forme d'une d-somme.

Programme Pascal (Delphi mode console)

```

program Magique;
{$APPTYPE CONSOLE}

uses
    SysUtils;

var
    FichierEntree, (* Fichier TEXT pour l'entrée *)
    FichierSortie : Text; (* Fichier TEXT pour la sortie *)

procedure OuvertureFichier;

```

```

begin
  Assign(FichierEntree, 'MAGIQUE.IN');
  Reset(FichierEntree);
  Assign(FichierSortie, 'MAGIQUE.OUT');
  Rewrite(FichierSortie);
end;

procedure LectureEtTraitement;
var
  d, n : integer;
  nbtermes, somme, dval : integer;
begin
  while not eof(FichierEntree) do
  begin
    readln(FichierEntree, d, n);
    writeln(FichierSortie, 'd = ', d);
    write(FichierSortie, 'n = ', n);
    if n mod d = 0 then
    begin
      nbtermes:=0;
      somme:=n;
      dval:=d;
      while dval<=somme do
      while dval>d do
      begin
        dval := dval div 10;
        while dval<=somme do
        begin
          nbtermes:=nbtermes+1;
          somme:=somme-dval;
          if nbtermes>1 then write(FichierSortie, ' +')
            else write(FichierSortie, ' =');
          write(FichierSortie, ' ', dval);
        end;
      end;
      writeln(FichierSortie, '');
      writeln(FichierSortie, 'La plus petite ', d, '-somme est en ', nbtermes, '.');
    end
    else
    begin
      writeln(FichierSortie, '');
      writeln(FichierSortie, 'il n''y a pas de ', d, '-somme possible pour ', n, '.');
    end;
    writeln(FichierSortie, '');
  end;
end;

procedure FermetureFichier;
begin
  Close(FichierEntree);
  Close(FichierSortie);
end;

begin
  OuvertureFichier;
  LectureEtTraitement;
  FermetureFichier;
end.

```

Programme C (DevC++ 4)

```

#include <stdio.h>
#include <stdlib.h>

FILE *FichierEntree; /* Fichier TEXT pour l'entrée */
FILE *FichierSortie; /* Fichier TEXT pour la sortie */

void OuvertureFichier()

```

```

{
  if ((FichierEntree = fopen("MAGIQUE.IN", "r")) == NULL)
  {
    perror("MAGIQUE.IN");
    exit(1);
  }
  rewind(FichierEntree); /* Ouverture de MAGIQUE.IN en lecture */
  FichierSortie = fopen("MAGIQUE.OUT", "w");
  if (FichierSortie != NULL)
    rewind(FichierSortie);
  else
    FichierSortie = tmpfile();
  if (FichierSortie == NULL)
  {
    perror("FichierSortie");
    exit(1);
  }
  /* Ouverture de MAGIQUE.OUT en écriture */
}

void FermetureFichier()
{
  if (FichierEntree != NULL)
    fclose(FichierEntree);
  FichierEntree = NULL;
  if (FichierSortie != NULL)
    fclose(FichierSortie);
  FichierSortie = NULL;
}

void LectureEtTraitement()
{
  int d, n, nbtermes, somme, dval;

  while (!feof(FichierEntree))
  {
    fscanf(FichierEntree, "%D %D", &d, &n);
    fprintf(FichierSortie, "d = %d\n", d);
    fprintf(FichierSortie, "n = %d\n", n);

    if (n%d==0)
    {
      nbtermes=0;
      somme=n;
      dval=d;
      while (dval<=somme) dval=10*dval+d;
      while (dval>d)
      {
        dval = dval/10;
        while (dval<=somme)
        {
          nbtermes++;
          somme-=dval;
          if (nbtermes>1) fprintf(FichierSortie, " +");
          else fprintf(FichierSortie, " =");
          fprintf(FichierSortie, " %d", dval);
        }
      }
      fprintf(FichierSortie, "\n");
      fprintf(FichierSortie, "La plus petite %d-somme est en %d.\n", d, nbtermes);
    }
    else
    {
      fprintf(FichierSortie, "\n");
      fprintf(FichierSortie, "Il n'y a pas de %d-somme possible pour %d.\n", d, n);
    }
    fprintf(FichierSortie, "\n");
  }
}

```

```
int main(int argc, char *argv[])
{
    OuvertureFichier();
    LectureEtTraitement();
    FermetureFichier();

    return 0;
}
```