

EPREUVE OPTIONNELLE d'INFORMATIQUE

CORRIGE

QCM

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 : Dans l'ordinateur, le programme BIOS (Basic Input Output System) est chargé dans la mémoire :

- A - de masse
- B - la RAM (Random Access Memory)
- C - la ROM (Read Only Memory)

Question 2 : La mémoire cache d'un ordinateur permet :

- A - une plus grande sécurité des données
- B - un accès rapide aux données
- C - une compression des données

Question 3 : La RS232C correspond à :

- A - une interface de programmation
- B - une interface de communication
- C - une application industrielle

Question 4 : Le système d'exploitation d'un ordinateur fait partie :

- A - du hardware
- B - du software
- C - des applications de bases de données

Question 5 : La taille d'un octet de données correspond à :

- A - 64 bits
- B - 1024 bits
- C - 8 bits

Question 6 : Dans un octet, le bit de parité permet :

- A - de synchroniser l'horloge du récepteur
- B - de détecter les erreurs de transmission
- C - de gérer le tour de parole

Question 7 : Le rôle d'un modem est :

- A - d'amplifier le signal
- B - de protéger l'ordinateur des chutes de tension
- C - de transformer le signal numérique en signal analogique

Question 8 : Un caractère ASCII est codé sur :

- A - 8 bits
- B - 7 bits

C - 16 bits

Question 9 : Un protocole de communication est :

- A - une autoroute de l'information
- B - une technologie de traitement de l'information
- C - un ensemble de règles et de structures

Question 10 : Un processus est :

- A - un processeur
- B - une entité exécutable
- C - un programme de gestion de files d'attentes

Question 11 : Dans l'ordinateur les données sont représentées sous forme :

- A - de signal électrique analogique
- B - de signal électrique numérique
- C - de signal lumineux

Question 12 : Dans un langage de programmation les pointeurs permettent :

- A - de partager une zone de mémoire
- B - de chaîner les blocs de mémoire
- C - de transférer des données sans les déplacer

Question 13 : La communication entre processus est :

- A - possible entre plusieurs ordinateurs reliés
- B - possible dans un seul ordinateur
- C - impossible

Question 14 : Dans une architecture Client/Serveur, le client doit :

- A - répondre aux requêtes du serveur
- B - envoyer des requêtes au serveur
- C - mettre des ressources en réseau

Question 15 : Dans une architecture Client/Serveur, le serveur doit :

- A - répondre aux requêtes des clients
- B - gérer le tour de parole
- C - mettre des ressources en réseau

Question 16 : La technique de compression de données permet de :

- A - crypter les données
- B - réduire la taille des données
- C - diminuer les coûts de communication

Question 17 : Un programme source devient exécutable après avoir été traité par un module appelé :

- A - un convertisseur
- B - un compilateur
- C - un traducteur

Question 18 : La technologie multimédia permet :

- A - de créer une réalité virtuelle
- B - de stocker plus d'informations
- C - de traiter tout type de données

Question 19 : DOS signifie :

- A - Data Output System

- B - Disk Operating System
- C - Device Open System

Question 20 : Dans l'UC de l'ordinateur, les données circulent :

- A - en mode parallèle
- B - en mode série
- C - en mode asynchrone

Question 21 : Quel est le langage le plus proche de la machine ? :

- A - le Pascal
- B - le C
- C - l'Assembleur

Question 22 : Le 68000 Motorola est :

- A - un microprocesseur
- B - un lecteur optique
- C - un réseau industriel

Question 23 : En algorithmique on parle de 2 types d'algorithmes. Lesquels ? :

- A - algorithme récursif
- B - algorithme itératif
- C - algorithme binaire

Question 24 : En algèbre de boole, le ou exclusif \oplus donne les résultats suivants :

- A - $1 \oplus 1 = 0$
- B - $1 \oplus 1 = 1$
- C - $0 \oplus 1 = 0$

Question 25 : La télématique est :

- A - le traitement de l'information par ordinateur
- B - la télévision numérique
- C - la fusion de l'informatique et des télécommunications

REMARQUE :

Les programmes en Pascal et C sont des exemples et peuvent être discutés en terme de construction. Le choix qui a été fait, est celui d'une découpe 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, le programme en C n'est pas optimisé, et se présente comme une Traduction quasi-directe du programme Pascal.

DECOUPAGE

Le principe est le suivant :

Nous récupérons dans un fichier (FEUILLE.IN) les tailles (longueur et largeur) des deux feuilles. Pour calculer la taille du plus grand carré découppable dans les deux feuilles sans qu'il y est de perte, il suffit de déterminer le PGCD des quatre nombres. Ensuite, à l'aide de ce dernier, on détermine le nombre de carrés obtenus pour chaque feuille et on les somme, ce qui nous donne le nombre total de carrés.

Programme Pascal (TP7)

```
program feuille;
uses crt;
var
Fichier      : Text;          (* Fichier TEXT pour l'entrée/Sortie *)

    Long, larg,
    Long2, Larg2 : Byte;      (* Taille inférieure à 100 *)
    TailleMax,
    NombreTotal : Byte;

procedure LectureFichier;      (* Lecture des données *)
begin
    Assign(Fichier, 'FEUILLE.IN');
    Reset(Fichier);           (* Ouverture de FEUILLE.IN en lecture *)
    Readln(Fichier, Long, Larg); (* Récupération de longueur et largeur *)
    Readln(Fichier, Long2, Larg2); (* pour les deux feuilles *)
    Close(Fichier);           (* Fermeture de FEUILLE.IN *)
end;

function Pgcd(i, j:byte):byte; (* Calcul du PGCD de deux octets *)
var
    reste : byte;            (* utilisation du modulo *)
begin
    while j>0 do              (* si i>j un tour pour rien, cela *)
    begin                      (* évite le test et la permutation *)
        reste := i mod j;
        i := j;
        j := reste;
    end;
    Pgcd := i;
end;

procedure EcritureFichier;     (* Ecriture du résultat *)
begin
    Assign(Fichier, 'FEUILLE.OUT');
    Rewrite(Fichier);          (* Création de FEUILLE.OUT *)
```

```

    writeln(Fichier, 'TAILLE MAXIMUM=', TailleMax); (* Ecriture du Résultat *)
    writeln(Fichier, 'NOMBRE TOTAL=', NombreTotal);
    Close(Fichier);                               (* Fermeture de FEUILLE.OUT *)
end;

```

```

procedure CalculeTailleEtNombre; (* Calcul des deux valeurs - appel de Pgcd *)

```

```

begin
    TailleMax := Pgcd(Long, Pgcd(Larg, Pgcd(Long2, Larg2)));
    NombreTotal := (Long DIV TailleMax) * (Larg DIV TailleMax);
    NombreTotal := NombreTotal + (Long2 DIV TailleMax) * (Larg2 DIV
TailleMax);
end;

```

```

begin
    LectureFichier;
    CalculeTailleEtNombre;
    EcritureFichier;
end.

```

Programme C (GCC)

```

#include <stdio.h>
#define byte unsigned char

FILE *Fichier;                /* Fichier TEXT pour l'entrée/Sortie */
byte Long, Larg;
byte Long2, Larg2;           /* Taille inférieur a 100 */
byte TailleMax, NombreTotal;

void LectureFichier()         /* Lecture des données */
{
    Fichier=fopen("FEUILLE.IN", "r"); /* Ouverture de FEUILLE.IN en lecture */
    fscanf(Fichier, "%d %d", &Long, &Larg); /* Récupération de longueur et */
    fscanf(Fichier, "%d %d", &Long2, &Larg2); /* largeur pour les 2 feuilles */
    fclose(Fichier);
}

byte Pgcd(byte i, byte j)     /* Calcul du PGCD de deux octets */
{
    byte reste;              /* utilisation du modulo */

    while (j>0)              /* Si i>j un tour pour rien, cela */
    {                         /* évite la permutation */
        reste = i % j;
        i = j;
        j = reste;
    }
    return (i);
}

void EcritureFichier()       /* écriture du resultat */
{
    Fichier = fopen("FEUILLE.OUT", "w"); /* Création de FEUILLE.OUT */
    fprintf(Fichier, "TAILLE MAXIMUM=%d\n", TailleMax); /* Ecriture résultat */
    fprintf(Fichier, "NOMBRE TOTAL=%d\n", NombreTotal);
    fclose(Fichier);
}

```

```

void CalculeTailleEtNombre() /* Cacul des deux valeurs - appel de Pgcd */
{
    TailleMax = Pgcd(Long, Pgcd(Larg, Pgcd(Long2, Larg2)));
    NombreTotal = (Long / TailleMax) * (Larg / TailleMax);
    NombreTotal += (Long2 / TailleMax) * (Larg2 / TailleMax);
}

int main()
{
    LectureFichier();
    CalculeTailleEtNombre();
    EcritureFichier();
    return (0);
}

```

HISTOGRAMME

Le principe est le suivant :

Nous commençons par lire dans un fichier (HISTO.IN) le nombre d'entrée. Puis, nous récupérons celles-ci. A chaque fois, que l'on trouve un chiffre (0 à 9), nous augmentons de 1 le nombre d'apparitions de ce chiffre dans un tableau (tableHisto) précédemment initialisé à zéro. Dans le même temps, nous déterminons quelle est la fréquence d'apparition la plus importante. Pour finir, une ligne est créée pour chaque niveau (de cette fréquence maximum jusqu'à un). Pour cela, on pose une étoile si le nombre d'apparition est égale à celui de la fréquence traitée et un espace si non. Cette ligne est écrite dans un fichier (HISTO.OUT). Et enfin on écrit la ligne de référence.

Programme Pascal (TP7)

```

program histo;
uses crt;
type
    T_Vect10Mot = array[0..9] of word;
var
    Fichier      : Text;          (* Fichier TEXT pour l'entrée/Sortie *)
    NombreEntree : Word;         (* Nombre d'entrées, lu en première ligne *)
    TableHisto   : T_Vect10Mot;  (* Tableau contenant les fréquences/Chiffre *)
    PlusGrand    : Word;

procedure InitTableau;          (* Initialisation des fréquences ... zéro *)
Var
    i : Byte;
begin
    For i:=0 to 9 do TableHisto[i]:=0;
end;

procedure LectureFichier;      (* Lecture des données *)
Var
    i      : Word;
    Chiffre : Byte;
begin
    PlusGrand:=0;              (* Initialisation de la plus grande fréquence ... zéro *)
    Assign(Fichier, 'HISTO.IN');
    Reset(Fichier);           (* Ouverture de HISTO.IN en lecture *)
    Readln(Fichier, NombreEntree); (* Récupération du nombre d'entrées *)
    for i:=1 to NombreEntree do
    begin
        Read(Fichier, Chiffre); (* Lecture des chiffres et mise ... jour *)
        Inc(TableHisto[Chiffre]); (* de leur fréquence d'apparition *)
        if TableHisto[Chiffre]>PlusGrand then PlusGrand:=TableHisto[Chiffre];
    end;
end;

```

```

    Close(Fichier);                (* Fermeture de HISTO.IN *)
end;

Procedure EcritureFichier;        (* Ecriture du résultat *)
Var
    i,j : Word;
    Ligne : String[20];           (* 20 caractères de longueur/Ligne *)
begin
    Assign(Fichier,'HISTO.OUT');
    Rewrite(Fichier);             (* Création de HISTO.OUT *)
    For i:=PlusGrand downto 1 do
    begin
        Ligne:='';
        for j:=0 to 9 do          (* Mise en place des '*' de l'histogramme *)
        begin                      (* et décrémentation des fréquences des chiffres *)
            if TableHisto[j]=i then (* concernés *)
            begin
                Ligne:=Ligne+'* ';
                Dec(TableHisto[j]);
            end
            else Ligne:=Ligne+' ';
            end;
            writeln(Fichier,Ligne);
        end;
        writeln(Fichier,'0 1 2 3 4 5 6 7 8 9');
    Close(Fichier);              (* Fermeture de HISTO.OUT *)
end;

begin
    InitTableau;
    LectureFichier;
    EcritureFichier;
end.

```

Programme C (GCC)

```

#include <stdio.h>

#define Word unsigned int

FILE *Fichier;                    /* Fichier TEXT pour l'entree/Sortie */
Word NombreEntree;                /* Nombre d'entrees, lu en premiere ligne */
Word TableHisto[10];              /* Tableau contenant les frequences/Chifre */
Word PlusGrand;

void InitTableau()                /* Initialisation des frequences a zero */
{
    Word i;

    for ( i=0; i<=9; i++ )
        TableHisto[i]=0;
}

void LectureFichier()             /* Lecture des donnees */
{
    Word i;
    Word Chiffre;

    PlusGran                      /* Initialisation de la plus grande frequence a zero */
    Fichier = fopen("HISTO.IN", "r"); /* Ouverture de HISTO.IN en lecture */
}

```

```

    fscanf(Fichier, "%d", &NombreEntree); /* Recuperation du nombre
d'entrees */
    for (i=0; i<NombreEntree; i++)
    {
        fscanf(Fichier, "%d", &Chiffre); /* Lecture des chiffres et mise a jour
*/
        TableHisto[Chiffre]++;          /* de leur frequence d'apparition */
        if (TableHisto[Chiffre]>PlusGrand)
            PlusGrand = TableHisto[Chiffre];
    }
    fclose(Fichier);                    /* Fermeture de HISTO.IN */
}

void EcritureFichier()                  /* Ecriture du resultat */
{
    Word i;
    Word j;
    char Ligne[21]; /* 2*10 caracteres de longueur/Ligne + 0 terminal */

    Fichier = fopen("HISTO.OUT", "w"); /* Creation de HISTO.OUT */
    for (i = PlusGrand; i>0; i--)
    {
        Ligne[0] = 0;
        for (j=0; j<=9; j++) /* Mise en place des '*' de l'histogramme */
        { /* et decrement des frequences des chiffres */
            if (TableHisto[j] == i) /* concernees */
            {
                strcat(Ligne, "* ");
                TableHisto[j]--;
            }
            else
                strcat(Ligne, " ");
        }
        fprintf(Fichier, "%s\n", Ligne);
    }
    fprintf(Fichier, "0 1 2 3 4 5 6 7 8 9\n");
    fclose(Fichier); /* Fermeture de HISTO.OUT */
}

int main()
{
    InitTableau();
    LectureFichier();
    EcritureFichier();
    return (0);
}

```