

## Programmation Impérative en Langage C – Examen (durée 2 heures)

*Troisième session "spéciale" du 24 mai 2012*

NOTES : Seul le document "**Correction TP1/TP2**" (5 pages) fourni lors de la session 1 est autorisé. Ne pas le dégrafer pour éviter que des feuilles volantes ne soient répandues sur les tables. Sont interdits tout autre document, les calculatrices, les téléphones, ainsi que tout autre ustensile de calcul et/ou de communication.

REMARQUE 1 : Dans la suite, les indications concernant les nombres de lignes sont données en comptant toute ligne de code non-vide (`#include`, prototypes, accolades...).

REMARQUE 2 : Les petits oublis de point-virgule, parenthèse, accolades, virgules, doubles quotes, *etc.* ne seront pas pénalisant tant qu'ils restent **punctuels**.

**Exercice 1 : Questions de cours**

- 1) Quel est le "point de départ" de l'exécution d'un programme écrit en langage C ?
- 2) Quels affichages produiront les 3 instructions suivantes ?

```
int nombre = 12 ;
nombre++ ;
printf ("La valeur est %d\n", nombre) ;
```

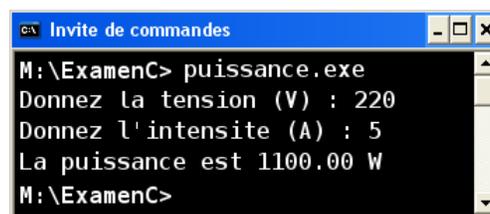
- 3) Le compilateur a détecté 3 erreurs dans le code suivant. Quelles sont-elles ?

```
int nombre
print ("Donnez un nombre entier : ") ;
scanf ("%d", nombre) ;
```

**Exercice 2 : Calcul de puissance**

La puissance électrique  $P$  en Watt se calcule à partir de la tension  $U$  en Volt et de l'intensité  $I$  en Ampère selon la formule suivante :  $P = U.I$ .

Écrire **UN PROGRAMME** dans lequel l'utilisateur donne au clavier la tension (en V) et l'intensité (en A), et qui affiche la puissance correspondante (en W).



```

C:\> Invite de commandes
M:\ExamenC> puissance.exe
Donnez la tension (V) : 220
Donnez l'intensite (A) : 5
La puissance est 1100.00 W
M:\ExamenC>

```

Remarque : 10 lignes (environ)

**Exercice 3 : Égalité**

Un pixel est gris quand ses trois composantes sont égales.

Écrire **UNE FONCTION** qui prend en paramètre les trois composantes RVB d'un pixel et qui retourne vrai si le pixel est gris, faux le cas échéant. Le prototype de la fonction sera le suivant :

```
int pixel_est_gris (unsigned char R, unsigned char V, unsigned char B) ;
```

Remarque : entre 3 et 6 lignes (environ)

**Exercice 4 : Chaînes de caractères et boucles**

Nous souhaitons écrire une fonction qui affiche une chaîne de caractères en remplaçant les voyelles par des points. Par exemple, si la chaîne donnée en paramètre est :

« Si vous comprenez cette phrase vous pouvez devenir developpeur »

alors l’affichage produit sera le suivant :

« S. v..s c.mpr.n.z c.tt. phr.s. v..s p..v.z d.v.n.r d.v.l.pp.r »

1) Commencer par écrire **UNE FONCTION** qui retourne vrai si le caractère passé en paramètre est une voyelle, faux sinon. Le prototype de la fonction sera le suivant :

```
int voyelle (char c) ;
```

AIDE : la fonction `char tolower (char c)` retourne le caractère `c` donné en paramètre après l’avoir passé en minuscule.

*Remarque : entre 3 et 10 lignes (environ)*

2) Écrire **UNE FONCTION** qui prend en paramètre une chaîne de caractères et qui affiche cette chaîne, en la parcourant caractère par caractère, en remplaçant chaque voyelle rencontrée par un point. Le prototype de la fonction sera le suivant :

```
void afficher_sans_voyelles (const char *s) ;
```

AIDE : la fonction `int strlen(const char *s)` retourne la longueur de la chaîne passée en paramètre.

*Remarque : entre 6 et 10 lignes (environ)*