



	1	2	5	
--	---	---	---	--

BLOQUET  
RomainPL1  
2013**Contrôle**

Mars 2014

Durée : 1 heure

Sans calculatrice, sans documents

Avant de débiter le CE :

- Prenez le temps de bien lire les énoncés des exercices. Vous répondrez dans les espaces laissés libres à cet effet dans le sujet. Aucune autre copie ne sera prise en compte.
- N'oubliez pas de reporter vos nom, prénom, groupe et école dans les espace ci-dessous

NOM

BLOQUE

PRENOM

Romain

GROUPE

PL1

ECOLE

Efrei



ESIGETEL



## Thème 1 : le passage des paramètres et les valeurs de retour, un peu de Langage C

QCM : 1 point pour chaque question bien traitée (en cas de fausse réponse : 0 point).

Première partie : chaque question a **une et une seule** bonne réponse

1) Un argument est :

- ☒ Fourni lors de la définition de fonction
- ☐ Une variable
- ☐ Traité comme étant une expression
- ☐ Calculé lors de la phase de compilation

2) Une fonction peut avoir :

- ☐ Plusieurs types de sortie
- ☒ Pas d'entrées, pas de sortie
- ☐ Plusieurs définitions
- ☐ Pas de corps si elle n'a pas de sortie

3) Une variable locale

- ☐ Doit avoir un nom unique dans tout le programme
- ☐ Peut changer de type à l'appel d'une fonction
- ☐ Est accessible depuis tout le programme
- ☒ N'est accessible que dans la fonction dans laquelle elle est définie

4) Un prototype de fonction

- ☐ Indique comment appeler la fonction
- ☒ Sert à exécuter la fonction
- ☐ N'a pas de type de sortie
- ☐ Peut avoir un corps





Deuxième partie : chaque question a **une ou plusieurs** bonne(s) réponse(s) - il faut fournir toutes les bonnes réponses pour avoir le point associé à la question

5) Soit le prototype suivant : `char *getNextEventText (t_evt *)` ;

Indiquez les affirmations vraies le concernant :

- ☐ Ce prototype se trouve dans un fichier .cpp
- ☒ La fonction `getNextEventText` retourne un tableau de caractères
- ☒ Cette fonction a un seul paramètre
- ☐ Cette fonction n'a pas de sortie
- ☒ Le paramètre attendu par la fonction est une adresse

6) Les directives `#include`, `#ifdef`, `#define`, `#endif`

- ☒ Peuvent se trouver dans un fichier.cpp
- ☒ Peuvent se trouver dans un fichier .h
- ☒ Peuvent servir à faire de la compilation conditionnelle
- ☒ Sont traitées à l'exécution du programme
- ☐ Sont des mots-clefs du langage C

7) Lors de la séance de cours-TP sur Code:Blocks, les points suivants ont été abordés :

- ☒ Gestion des configurations de production / développement
- ☐ exemple de code avec allocation de mémoire
- ☒ Rôle et configuration d'un test unitaire
- ☐ Inclusion de fichier .cpp dans un fichier .h
- ☐ Débogage et fuites de mémoire

8) En pseudo-langage algorithmique, lorsque l'on écrit `retourner quelquechose` ;

- ☒ Ce `quelquechose` doit avoir le type annoncé en sortie dans l'entête de la fonction
- ☒ Ce `quelquechose` doit être une variable locale
- ☐ Ce `quelquechose` doit porter le nom d'une variable de la fonction appelante
- ☐ Ce `quelquechose` est une expression
- ☐ Ce `quelquechose` ne peut pas être une adresse



## Thème 2 : modules et structures

Qu'est-ce qu'une structure ?

① Une structure est un type composé. (que l'on crée) elle est composée d'une entête et d'un corps contenant ses champs. Exemple :

	structure	nom du type
{	champ du type	(attributs);
}		

En C, que signifie le mot-clef struct ?

① En langage C le mot clef struct sert à définir une nouvelle structure il signifie donc structure.  
exemple :

struct	S-complexe
{	long re, im;
}	

En C, que signifie le mot-clef typedef ?

① le mot clef typedef en langage C sert à définir un nouveau type, il signifie donc définir un nouveau type.  
exemple :

typedef struct	S-complexe
{	long re, im;
}	T-complexe;

Soit la définition de structure suivante, en pseudo-langage algorithmique  
Quelles informations en tirez-vous ?

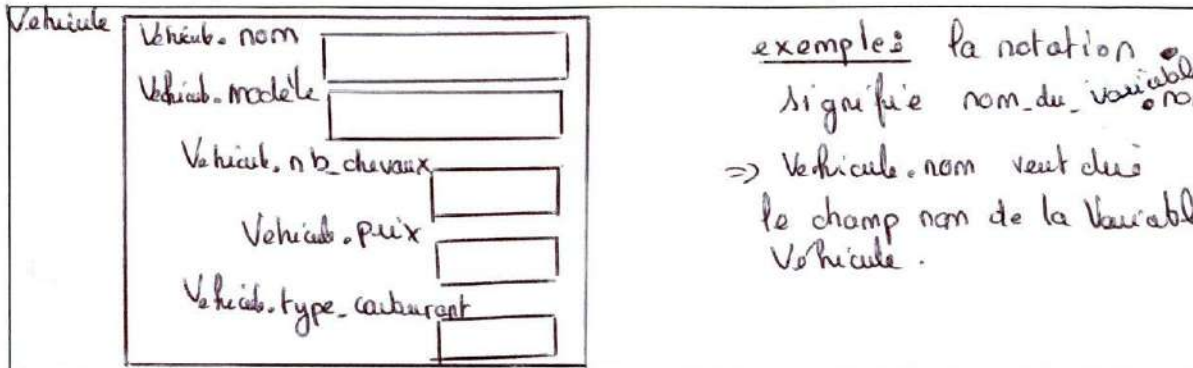
```
structure voiture
{
    caractere *nom;
    caractere *modele;
    entier    nb_chevaux;
    reel      prix;
    entier    type_carburant;
};
```

Tableaux dynamiques

①,5 le nom du type sera Voiture.  
Il aura un champ composé d'une chaîne de caractère pour son nom,  
une chaîne de caractère pour le modèle.  
Un champ de type entier pour définir le nombre de chevaux,  
un champ de type réel pour le prix  
et un dernier champ de type entier pour le type de carburant.

Soit la définition de variable suivante : faites le schéma de cette variable (schéma de type boîte utilisé depuis le début de l'année).

voiture vehicule;



Soit le fichier .h suivant (étudié en cours). Indiquez les informations portées par ce fichier

```
#ifndef DATE_H
#define DATE_H

typedef struct s_date
{
    long jj, mm, aa;
} t_date;

void    afficherDate(t_date);
void    saisirDate(t_date *);
t_date  calculerPeriode(t_date, t_date);
long    comparerDates(t_date, t_date);

#endif
```

- Il y a la définition du type date, qui est composé de trois champs entiers jj, mm, aa (jour, mois, année).
- Il y a le prototype de la fonction afficherDate avec son type de sortie (il y en a pas) et son type d'entrée (elle attend une adresse).
- Il y a le prototype de la fonction saisirDate qui n'a pas de sortie et deux entrées de type t\_date.
- Il y a le prototype de la fonction calculerPeriode qui est composé d'une sortie de type t\_date et deux entrées de type t\_date.
- Il y a le prototype de la fonction comparerDates qui est composé de deux entrées de type t\_date et d'une sortie de type entier.



Dans le fichier .cpp correspondant, on trouve la définition de la fonction **afficherDate()**, qui affiche la date sous la forme jj/mm/aaaa : voici le code correspondant

```
void afficherDate(t_date d)
{
    cout << d.jj << "/" << d.mm << "/" << d.aa;

    return;
}
```

Indiquez ce que signifient les notations **d.jj**, **d.mm**, **d.aa**

la notation d.jj signifie le champ jj de la variable d  
la notation d.mm signifie le champ mm de la variable d.  
la notation d.aa signifie le champ aa de la variable d.

la Variable d est de type t\_date.

Quels sont les fichiers .h qui doivent être inclus par **#include** dans ce fichier .cpp ? Justifiez votre réponse.

Il faut inclure :  
- `#include <iostream>` pour les chevrons  
- `using namespace std;` pour cout et cin  
- `#include "Date.h"` car lors de la compilation le type t\_date ne sera pas reconnu, il y aura donc une erreur.