

Aucun document n'est autorisé / durée 2 heures /
La présentation et le soin sont pris en compte pour la notation.

```
/* Auteur : Eric Bachard
 * Licence de ce document : Creative Commons pas d'utilisation commerciale
 * pas de modification. Voir http://creativecommons.org/licenses/by-nc-nd/2.5/
 */
/* Author : Eric Bachard
 * This document is licensed under Creative Commons License : no commercial
 * use, no modification. See : http://creativecommons.org/licenses/by-nc-nd/2.5/
 */
```

Cours (sur 9 points)

Types et langage C

C.1) Donner l'occupation en octets en mémoire des types : short int , int , long double et long int **[1 pt]**

Type	short int	int	long double	long int
Occupation mémoire	2 octets	2 ou 4 octets	10 octets	4 octets

C.2) Comment définit-on un tableau de 5 valeurs, initialisées à 0 en langage C ? **[1 pt]**

Comme le type n'est pas précisé, on suppose que c'est un tableau d'entiers

```
int tableau[5] = {0,0,0,0,0};
```

C.3) comment définit, initialise et utilise-t-on en langage C une structure de type fiche ayant pour membres

- un tableau de 20 caractères
- un nombre réel
- un nombre entier

[1 pt]

```
/* definition */
struct fiche {
    char tab[20] ;
    float prix ;
    int stock ;
};
```

```
/* initialisation */
```

```
struct fiche velo;
```

```
/* utilisation */
velo.prix = 120.35;
velo.stock = 12;
```

```
...etc
```

LO20 / A06 Initiation à la programmation en C et aux outils de programmation Eric Bachard

Éléments de correction

C.4) Expliquer la différence essentielle existant entre les deux lignes :

```
valeur = 'a';  
valeur = "a";
```

[0,5 pt]

'a' désigne une variable de type char, qui représente le caractère a, et qui occupe un octet en mémoire.

"a" désigne une chaîne de caractères, constituée de deux caractères : a suivi de \0 (fin de chaîne)

La commande " gcc -o math log.c -Wall ", donne l'erreur de compilation suivante :

```
log.c: In function 'main':  
log.c:10: warning: control reaches end of non-void function
```

C.5) Quel est le problème ? [1 pt]

Il manque un " return 0; " (ou encore "" return EXIT_SUCCESS ") en fin de main()

Utilisation de gcc / lignes de commande

On souhaite compiler le fichier prog.c en un exécutable test1 avec les options de compilations suivantes :

- le code doit être conforme à la norme ANSI
- tous les warnings seront visibles
- les symboles seront inclus pour l'utilisation avec le débogueur GNU gdb

C.6) Donner la ligne de commande nécessaire [0,5 pt]

```
gcc -Wall -ansi -ggdb -o test1 prog.c
```

que font les commandes :

C.7) gcc -S prog2.c [0,5 pt]

gcc transforme le code C en un code traduit en assembleur, après avoir vérifié la syntaxe et inclus tous les symboles et prototypes de fonctions nécessaires

C.8) gcc -Wall -gdwarf-2 -o prg_1 prog.c [0,5 pt]

gcc crée le binaire prg1 à partir du fichier source prog.c, en affichant tous les warnings, et en incluant la table de symboles étendus pour le débogage en respectant la norme dwarf-2.

On souhaite maintenant compiler le programme matrices.c avec les options -pedantic, -Wall, obtenir l'exécutable projet_maths en linkant ce programme avec la bibliothèque de fonctions mathématiques libm.so.

C.9) Donner la ligne de commande nécessaire à la compilation de ce binaire. [0,5]

```
gcc -pedantic -Wall -o projet_maths matrices.c -lm
```

Utilisation de diff et patch

On souhaite créer un patch entre les fichiers (respectivement) salfile.c et salfile_new.c , et le patch s'appellera salfile.c_new_impl.diff

C.10) Donner la ligne de commande correspondante [0,5 pt]

```
diff -u salfile.c salfile_new.c > salfile.c_new_impl.diff
```

LO20 / A06 Initiation à la programmation en C et aux outils de programmation Eric Bachard

Éléments de correction

Note : l'option -u (format unifié est toujours conseillé)

C.11) Comment applique-t-on ce patch (la commande complète est demandée) ? [0,5 pt]

On place le patch dans le répertoire contenant salfile.c et on tape la commande :

```
patch -p0<salfile.c_new_impl.diff
```

C.12) Donner la commande permettant de créer un patch (récursivement) entre les répertoires tree et tree_new [0,5 pt]

```
diff -Naur tree tree_new > patch_recuratif.diff
```

C.13) Quelle est la différence entre les fins de lignes DOS et les fins de lignes Unix dans les fichiers textes ? [1 pt]

Fin de lignes DOS contient retour chariot suivi de new line

Fin de lignes Unix : new line seulement

Exercice 1 (sur 3 pts)

1.1) Expliquer ce que fait le programme suivant, écrit en langage C : [2 pts]

```
/* inclusion bibliotheque standard */
#include <stdio.h>

/* inclusion stdlib.h (EXIT_SUCCESS) */
#include <stdlib.h>

/* MACROS constantes, qui remplaceront FOO(x) par -1./x a chaque occurrence ,
de meme qu'on remplacera BAR(x) par -1./x aussi a chaque occurrence
on pourra vérifier cela en arretant la compilation apres le preprocesseur : gcc -E ...etc */
#define FOO(x) -1. / (x)
#define BAR(x) -1. / (x)

/* fonction principale */
int
main (void)
{
/* on definit deux entiers x et y */

int x = 2;
int y = 10;

/* on affiche 1/x. Notez le typage en reel, grace a la macro qui utilise 1. dans le formatage */
fprintf(stdout,"Valeur obtenue %f\n", FOO(x));

/* on affiche cette fois 1/y mais avec 3 chiffres + 2 decimales */
fprintf(stdout,"Valeur obtenue %3.2f\n", BAR(y));
```

```
/* si aucun probleme , on quitte main et le programme se termine */  
  
return EXIT_SUCCESS;  
}
```

1.2) Dans ce même programme, une erreur qui va faire échouer la compilation. Quelle est cette erreur, et comment la corrige-t-on ? [1 pt]

Noter l'espace entre FOO et (x) -> la compilation va échouer, pour cause de variable indéfinie

```
#define FOO (x) -1. / (x)
```

Exercice 2 Algorithme du Jeu de la fourchette (sur 8 pts)

Le jeu de la fourchette consiste à découvrir un nombre choisi par l'adversaire grâce aux indications de celui-ci qui indique après chaque coup dans quelle fourchette se trouve le nombre qu'il a choisi

Exemple de déroulement attendu

45
démarrer le programme
proposer une partie ou quitter
le nombre est choisi par l'ordinateur (ici : 52);
afficher " choisir x entier compris entre 1 et 100 "
la proposition de l'utilisateur est 50
[... cut ...]
afficher : " choisir x entier compris entre 50 et 54"
la proposition de l'utilisateur est 52
afficher: " nombre découvert en 6 essais"
proposer une autre partie ou quitter

2.1) Ecrire l'algorithme correspondant au déroulement attendu

[4 pts]

```
Variables: entier:      continuer  
                  essai  
                  nombre_cache  
                  nombre_saisi  
                  min  
                  max  
  
Début:  
  Écrire ("Saisir 1 pour jouer, 0 pour quitter")  
  Lire ( continuer)  
  essai <-- 1  
  min <-- 1  
  max <-- 100  
  nombre_cache <-- 52  
  Tant que (continuer == 1 ou essais < max )  
    Écrire: ("Choisir x entier compris entre 'mini' et 'maxi' ")  
    Lire: (nombre_saisi)  
    Si:  
      Alors :  
        (nombre_saisi < nombre_cache)
```

LO20 / A06 Initiation à la programmation en C et aux outils de programmation Eric Bachard
Éléments de correction

```
        min <-- nombre_saisi
        incrémenter essais
    Sinon (nombre_saisi > nombre_cache)

        Faire: max <-- nombre_saisi
        incrémenter essais
    Sinon
        Ecrire( "Nombre decouverts en 'essais' essais" )
        essais <-- max + 1;
        min <-- VAL_MIN;
        max <-- VAL_MAX;
        Écrire ("Saisir 1 pour jouer, 0 pour quitter")
        Lire ( continuer)
    Fin Si
    Si ( essais > max )
        Continuer <-- 0
    Fin si
    Fin Tant que
Fin
```

2.2) Écrire le programme en langage C qui permet de mettre en oeuvre cet algorithme [4 pts]

```
/* Programme de la fourchette Exame A06 LO20 Eric Bachard */

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

/* constantes */

#define NOMBRE_CACHE 52
#define VAL_MAX 100
#define VAL_MIN 1

int main(void)
{
    int continuer;
    int essais = 1 ;
    int nombre_cache;
    int nombre_saisi;
    int min = VAL_MIN;
    int max = VAL_MAX;

    fprintf(stdout, "Saisir 1 pour jouer, sinon 0 pour quitter\n");
    fscanf(stdin, "%d", &continuer);

    while ( (continuer != 0) )
    {
        nombre_cache = NOMBRE_CACHE;
        fprintf(stdout, "Choisir x entier compris entre %d et %d\n", min, max);
        fscanf(stdin, "%d", &nombre_saisi);

        if (nombre_saisi < nombre_cache)
        {
            min = nombre_saisi;
            essais += 1;
        }
        else if (nombre_saisi > nombre_cache )
        {

```

```
    max = nombre_saisi;
    essais += 1;
}
else /* nombre_saisi == nombre_cache */
{
    fprintf(stdout, "Nombre cache decouvert en %d essais\n", essais);
    essais = 1;
    min = VAL_MIN;
    max = VAL_MAX;
    fprintf(stdout, "Saisir 1 pour jouer, sinon 0 pour quitter\n");
    fscanf(stdin, "%d", &continuer);
}
if ( essais > max )
    continuer = 0;
}
return EXIT_SUCCESS ;
}
```