

Tous documents interdits à l'exception d'une page A4 recto, manuscrite de votre main. La correction tiendra compte de la qualité de l'argumentaire et de la présentation. Le barème indiqué est approximatif.

## Exercice 1 : Debugger les problèmes mémoire (2pts)

Lorsque l'on exécute le programme ci-dessous, on obtient le message d'erreur suivant : `Erreur de segmentation`. Pourquoi ? Corrigez les (nombreux) problèmes.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 typedef struct fiche {
6     char *nom;
7     char *prenom;
8     int age;
9 } fiche_t ;
10
11 void maj_fiche(int no, fiche_t *fich){
12     if (no>1999)
13         no=no % 2000;
14     *(fich.no_fiche) = no;
15 }
```

```
void main() {
    fiche_t *fiche;
    fiche->nom = malloc(sizeof(char));
    strcpy(fiche->nom, "Bon");
    strcpy(fiche->prenom, "Jean");
    fiche->age = 3;
    printf("Nom : %s\n", fiche->nom);
    printf("Prenom : %s\n", fiche->prenom);
    printf("Age : %s\n", fiche->age);
}
```

## Exercice 2 : Manipuler des chaînes de caractères (3pts)

Écrivez un programme C ***min\_maj.c***, compilé sous le nom ***min\_maj***, qui affiche à l'écran ses arguments en convertissant toutes les lettres minuscules en lettres majuscules (les autres caractères étant inchangés).

`min_maj bOnNE nUIt LES petits ...` affiche à l'écran : BONNE NUIT LES PETITS ...

`min_maj lE c, tROp 3lL3t` affiche à l'écran : LE C, TROP 3LL3T

*Indication* : le décalage entre les majuscules et les minuscules est donné par l'expression `'A'-'a'`.

## Exercice 3 : Lire un fichier (3pts)

La commande ***strings*** permet d'étudier les parties lisibles d'un fichier binaire dont le nom est passé en argument. Elle n'affiche à l'écran que les caractères du fichier dont les codes Ascii sont dans l'intervalle [32,126] (ce sont les caractères "affichables", les autres étant des codes de contrôle).

*Exemple* : si ***fichier*** contient les caractères de codes Ascii (12, 33, 65, 69, 5, 4, 77, 89, 14, 35, 101, 102) alors `strings fichier` affichera les caractères : (33, 65, 69, 77, 89, 35, 101, 102). Soit : `!AEMY#ef`

(3.1) Écrivez cette comande en C.

Nom :

### Exercice 4 : QCM (2pts)

Répondez sur la feuille fournie (indiquez votre nom). Il peut y avoir plusieurs cases valides par question ; les réponses fausses seront pénalisées.

- Le langage C a été inventé chez ...
  - ☐ AT&T
  - ☐ bell
  - ☐ IBM
  - ☐ Sun
- En cas de succès, que doit retourner la fonction main ?
  - ☐ 0
  - ☐ 1
  - ☐ -1
  - ☐ void
- Quel caractère termine classiquement les chaînes en mémoire en C ?
  - ☐ '\n'
  - ☐ '\0'
  - ☐ ''
  - ☐ '\END'
- Que vaut  $7/9*9$  en C ?
  - ☐ 0
  - ☐ 0.08642
  - ☐ 1
  - ☐ 42
- Si p est de type short\* et de valeur 0x1000, que vaut p+1 ?
  - ☐ 0x1001
  - ☐ 0x1002
  - ☐ 0x1003
  - ☐ 0x1004
- En C, a[i] est équivalent à ...
  - ☐ &(a+i)
  - ☐ a+i
  - ☐ \*(a+i)
  - ☐ a[0] + i
- On obtient l'adresse de la variable v avec ...
  - ☐ &v
  - ☐ \*v
  - ☐ @v
  - ☐ #v
- Par quel caractère doit commencer une ligne de commande dans un Makefile ?
  - ☐ un dollar
  - ☐ une dièse
  - ☐ une espace
  - ☐ une tabulation