

Nom et prénom : Saraiva Maria Leandir

Travail écrit no. 1

(Durée : 75 minutes)

Consignes :

- **ECRIEZ VOS REPONSES DIRECTEMENT SUR CE DOCUMENT**
- Seules les réponses figurant sur ce document seront corrigées
- Une écriture lisible est exigée
- Vous pouvez écrire au crayon papier
- Ne pas dégrafer ce document
- Aucune documentation n'est autorisée, hormis le résumé C figurant à la fin de ce document

4.25

Question 1 (8 points)

3

a) (3 pts)

Pour chacune des suites d'instructions ci-dessous, indiquer dans la colonne de droite du tableau soit la valeur affichée par le printf, soit "Ne compile pas" si vous pensez que la suite d'instructions considérée provoque une erreur à la compilation.

Suite d'instructions	Valeur affichée
<pre>#define double(d) d + d int n = -3; printf("%d\n", double(-3 == n));</pre>	0 ✓
<pre>#define CARRE(x) x * x int n = 3; printf("%d\n", CARRE(n-1));</pre>	-1 ✓
<pre>#define N 2 #define STR(X) #X #define F(N) STR(Fichier_##N) printf("%s\n", F(n));</pre>	Fichier_n ✓

- 1.25 b) (2.5 pts)
Que va afficher le programme suivant ?

```
#include <stdint.h>
#include <stdio.h>

uint16_t f(uint16_t a, uint8_t b) {
    const uint16_t MASK = 0x8000;
    while (b--) { // 5 fois
        a = (uint16_t) (a << 1 | ((a & MASK) != MASK));
    }
    return a;
}

int main(void) {
    printf("%#x", f(0x1234, 5));
}
```

00010010 0011 0100

!MSB(a)

Votre réponse :

0x469f 0x469d

- c) (2.5 pts) (x minuscule)
Que va afficher le programme suivant si &m[0][0] vaut 0xc3d5ffbd0 ?

```
#include <inttypes.h>
#include <stdio.h>

#define PRINT_ADDRESS(ADR) printf("0x%" PRIxPTR "\n", (intptr_t) (ADR))

int main(void) {
    int64_t m[4] = {{1}, {0, 1}, {0, 0, 1}, {0, 0, 0, 1}};
    PRINT_ADDRESS(*(m+2) + (&m[2][-1] - 3[m]));
}
```

avant d'entrer dans la boucle: 0001 0010 0011 0100

b-- = 5: 0010 0100 0110 1001

b-- = 4: 0100 1000 1101 0010

b-- = 3: 0001 0001 1010 0110

b-- = 2: 0010 0011 0100 1100

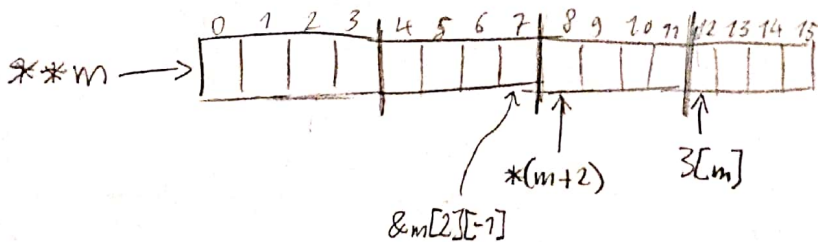
b-- = 1: 0100 0110 1001 1101

= 0x469d

Votre réponse :

Ne compile pas

3[m] équivalent à m[3]



PRINT_ADDRESS(8 + (7 - 12));

L → +3 · sizeof(int64_t) = +24 = +0x18

0xc3d5ffbd0
+ 0x000000018

0xc3d5fffb28

Saraiva Maria Leandro

4.5

Question 2 (7.5 points)

Que va afficher le programme suivant ?

IMPORTANT

- Dans vos réponses, désignez par la lettre **b** la présence d'un espace blanc.
- Seules les réponses 100% correctes seront comptabilisées.

```
#include <stdio.h>

int main(void) {

    printf("1) |%.5d|\n", 230);
    printf("2) |%-#5X|\n", 230);
    printf("3) |%f|\n", 12.345);
    printf("4) |%#.f|\n", 12.345);
    printf("5) |%2g|\n", 12.345);
    printf("6) |%+5.1f|\n", 0.157);
    printf("7) |%*. *E|\n", 8, 1, 0.01);
    printf("8) |%G|\n", 1e-4);
    printf("9) |%06.4g|\n", 123.456);

    // On suppose que l'utilisateur entre la chaîne " (089) 123-45-67 : Paul"
    #define TAILLE_MAX 50
    char chaine[TAILLE_MAX + 1];
    printf("Entrez une chaîne de caracteres (%u caract max) > ", TAILLE_MAX);
    scanf("%*[ ]%[(0123456789)]", chaine);
    printf("10) |%s|\n", chaine);
}
```

Vos réponses :

- 1) |00230| ✓
- 2) |0XE6b| ✓
- 3) |12.345000| ✓
- 4) ~~|12.345000|~~ |12| |12.1| ✓
- 5) ~~|12|~~ |12.345|
- 6) ~~|+00.2|~~ |b+0.2|
- 7) ~~|bbb7E-02|~~ |b70E-02|
- 8) |0.0001| ✓
- 9) ~~|123.456|~~ |123.4560| ~~|~~ |0123.5|
- 10) |(089)| ✓

4

Question 3 (7.5 points)

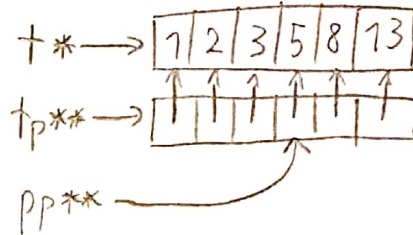
3.75 a) (3.75 pts)

Soient les déclarations suivantes :

```
int t[] = {1, 2, 3, 4, 5, 13};
int* tp[] = {t, t+1, t+2, t+3, t+4, t+5};
int** pp = tp + 3;
```

Quelle valeur fournit chacune des expressions ci-dessous ?
(Conseil : Aidez-vous d'un petit dessin)

- 1) $-(**pp)$
- 2) $pp[1][-1]$
- 3) $(*pp++)[1]$
- 4) $++(*pp--)$
- 5) $*(+pp) - (*tp) + 1$



IMPORTANT

- Les 5 expressions sont supposées être évaluées les unes après les autres, en commençant par la no 1). Le résultat d'une certaine expression peut donc dépendre de ce qui s'est produit lors de l'évaluation des expressions précédentes.

Vos réponses :

- 1) 4 ✓
- 2) 4 ✓
- 3) 8 ✓
- 4) 9 ✓
- 5) 5 ✓

0.25 b) (3.75 pts)

Soient les déclarations suivantes :

```
const char* t[] = {"Mars", "Mai""Avril", "Mai"};  
const char** tp[] = {t, t+1, t+2};  
char** pp = (char**) t+1;
```

Quelle valeur fournit chacune des expressions ci-dessous ?
(Conseil : Aidez-vous d'un petit dessin)

- 1) ++t[0]
- 2) ** (tp[1] - 1)
- 3) (*(+(t+pp)))
- 4) *(tp[2]) + 1
- 5) pp[0][1]

IMPORTANT

- Les 5 expressions sont supposées être effectuées les unes après les autres, en commençant par la no 1). Le résultat d'une certaine expression peut donc dépendre de ce qui s'est produit lors de l'évaluation des expressions précédentes.

Consigne

- Pour les caractères, donner le résultat entre simples guillemets.
Exemple : 'a'.
- Pour les chaînes de caractères, donner le résultat entre doubles guillemets.
Exemple : "ABC".

Vos réponses :

- 1) ~~"Avril"~~ "ars"
- 2) ~~'M'~~ 'a'
- 3) ~~'M'~~ 'v'
- 0.25 4) ~~78~~ 'N' 78
- 5) ~~'M'~~ 'r'

Question 4 (7 points)

1) (3 pts)

Traduire en français les déclarations C suivantes :

- a) `char* const (*a(void))[3];`
- b) `const int *(*b)(void (*)(void))[3];`

Vos réponses :

a) a est un tableau de 3 pointeurs sur ~~une~~ des fonctions prenant aucun paramètre et livrant un pointeur constant sur char

b) b est un pointeur sur un pointeur de fonction renvoyant un int constant et prenant en paramètre un ~~tableau de 3 pointeur~~ ~~sur une fonction prenant~~ pointeur sur un tableau de 3 pointeurs sur void

2) (4 pts)

Ecrire les déclarations C correspondant aux énoncés suivants :

Attention! Des points seront déduits en cas de parenthésage inutile

- a) f est une fonction prenant en paramètre un pointeur constant sur char et livrant un pointeur sur un tableau de 3 pointeurs sur double
- b) t est un tableau de 3 pointeurs pointant chacun sur une fonction prenant en paramètre un pointeur sur int constant et livrant un pointeur sur un tableau de 10 double

Vos réponses :

- a) ~~`*((double *) [3]) (*f) (char * const)`~~
- b) ~~`*((*(double [10]) (int * const) + [3])`~~

Bonne chance
pour vos
révisions!

2

Question 5 (8 points)

Ecrire, de la manière la plus simple et efficace possible, une fonction C qui prend en paramètre une matrice n (lignes) \times m (colonnes) de `int` et qui livre en retour un tableau contenant les adresses des quatre éléments constituant les quatre "coins" de la matrice.

Exemple

Si matrice = $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$, la fonction doit renvoyer un tableau contenant les adresses des éléments 1, 3, 7 et 9 (dans cet ordre)

IMPORTANT

- Traduire dans la fonction le fait que le(s) paramètre(s) sont supposés valides.
- Les adresses des éléments constituant les "coins" de la matrice sont à renvoyer dans l'ordre suivant : 1) coin supérieur gauche, 2) coin supérieur droit, 3) coin inférieur gauche, 4) coin inférieur droit (voir exemple ci-dessus)

Votre réponse :

```
#include <stdlib.h>
#include <assert.h>
```

par demande

```
int* getCoins(int* mat, size_t n, size_t m) {
    assert(n > 0);
    assert(m > 0);
```

```
    int* ptr = malloc(4 * sizeof(int));
```

```
    *ptr++ = *mat;
```

```
    *ptr++ = *(mat + n - 1);
```

```
    *ptr++ = *(mat + n * m - m);
```

```
    *ptr = *(mat + n * m - 1);
```

```
    return ptr;
```

```
}
```

possible
mais
pas des
plus judicieux

cf corrigé en classe

(j'ai oublié de recopier ds1)