5.2

Nom et prénom : Francheti Thibaud

Travail écrit no. 2

(Durée : 2 périodes)

Directives :

- **ECRIVEZ VOS REPONSES DIRECTEMENT SUR LA DONNEE**
- Ne pas dégrafer le document
- Vous pouvez écrire au crayon
- L'usage d'une calculette n'est pas autorisé
- Seule documentation autorisée : **la Quick Reference Card C (non annotée !)**

6.5

Question 1 (6.5 points)

2

a) (2 pts)
Qu'affiche le code suivant ?

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

uint16_t f(uint16_t a, uint8_t b) {
   for (uint8_t c = 0; c < b; ++c)
      a = (uint16_t) ((a << c) | (a & 0x8000 ? 1 : 0));
   return a;
}

int main(void) {
   printf("%#x\n", f(0xabcd, 3));
   return EXIT_SUCCESS;
}
```

Votre réponse :

~~0x5EGC~~

0x5e6c ✓

2.5 b) (2.5 pts)
Qu'affiche le code suivant ?

1) si ce dernier est supposé compilé en **32 bits** et que *m* est supposé contenir l'adresse 0x60fd50 ?
2) si ce dernier est supposé compilé en **64 bits** et que *m* est supposé contenir l'adresse 0x60fd50 ?

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

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

typedef struct {
   uint16_t a;
   size_t b;
} Element;

int main(void) {
   Element m[3][4] = {{{0, 0}}};
   PRINT_ADDRESS(&m[2][3]);
   return EXIT_SUCCESS;
}
```

Vos réponses :

1) 0x60fdc8 ✓

2) 0x60fe00 ✓

2 c) (2 pts)
Qu'affiche le code suivant ?

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

int main(void) {

   int t1[] = {0, 1, 0, 1, 0};
   int t2[] = {1, 0, 1, 0, 1};

   int *p1 = t1, *p2 = t2;

   while (!*p1++ || !*++p2);
   printf("%" PRIdPTR " %" PRIdPTR "\n",
          (ptrdiff_t) (p1 - t1), (ptrdiff_t) (p2 - t2));

   return EXIT_SUCCESS;
}
```

Votre réponse : 4 2 ✓

5.625

Question 2 (6.25 points)

Que va afficher le programme suivant, après que l'utilisateur ait saisi la suite de caractères " (024) 1234567 - RRH " ?

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>
#include <stdlib.h>

#define TAILLE_MAX 50

int main(void) {

   int n = 216;
   printf("1) |%5d|\n", n);
   printf("2) |%.5d|\n", n);
   printf("3) |%-#5x|\n", n);

   double x = 0.13579;
   printf("4) |%f|\n", x);
   printf("5) |%g|\n", x);
   printf("6) |%+5.1f|\n", x);
   printf("7) |%10.2e|\n", x);

   printf("8) |%03.2g|\n", 13.579);
   printf("9) |%#G|\n", .1E-3);

   // test : " (024) 1234567 - RRH "
   char chaine[TAILLE_MAX + 1] = "";
   printf("Entrez une chaine de caracteres (%u caract max) > ", TAILLE_MAX);
   scanf("%*[ ]%[(0123456789)]", chaine);
   printf("10) |%s|\n", chaine);

   return EXIT_SUCCESS;
}
```

Vos réponses :

1) 1)b|bb216| ✓
2) 2)b|00216| ✓
3) 3)b|0xd8b| ✓
4) 4)b|0.135790| ✓
5) 5)b|0.13579| ✓
6) 6)b|b+0.1| ✓
7) 7)b|b1.36e-001| ✓
8) 8)b|b141
9) 9)b|0.000100000| ✓
10) 10)b|(024)| ✓

9

4.6875

Question 3 (6.25 points)

2.5

a) Soient les déclarations suivantes :

```
int a[] = {1, 5, 13, 7, 8, 2};
int* b[] = {a, a+1, a+2, a+3, a+4, a+5};
int** c = b + 2;
int*** d = &c;
```

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

1) `--**c`
2) `c[1][-1]`
3) `**(c+3)`
4) `*++c-*b+1`
5) `-1[(**d)]`

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) 12 ✓
2) 12 ✓
3) 2 ✓
4) 4 ✓
5) ~~-7~~ 4

2.1875

b) Soient les déclarations suivantes :

```
const char* a[] = {"Vivement", "les", "vacances"};
const char** b[] = {a, &a[1], a+2};
char** c = (char**) a;
```

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

1) `++a[0]`
2) `**(b[1]-1)`
3) `*++*++c`
4) `**b[2]+1`
5) `c[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.

Vos réponses :

1) "ivement" ✓
2) ';' ✓
3) 'e' ✓
4) les (code ascii de 'w') 0.5
5) "vacances" 3.5

6

Question 4 (6 points)

1) Traduire en français les déclarations C suivantes :

a) `const double *(*a)(void *(*)[5]);`
b) `double* const (*b(void))[5];`

2) Ecrire les déclarations C correspondant aux énoncés suivants :
Attention! Des points seront déduits en cas de surparenthésage

a) f est une fonction prenant en paramètre un pointeur constant sur char et livrant un pointeur sur un tableau de 5 pointeurs sur int

b) t est un tableau de 5 pointeurs pointant chacun sur une fonction prenant en paramètre un int et livrant un pointeur sur un tableau de 10 int

Vos réponses :

3

1)

a) a est un pointeur sur une fonction prenant en paramètre
1.5 on ~~fait~~ pointeur sur un tableau de 5 pointeurs sur void
et retournant un pointeur sur double constant. ✓

b) b est une fonction sans paramètre qui renvoit un
1.5 pointeur sur un tableau de 5 pointeurs constants
sur double, ✓

3

2)

a) `int* (*f(char* const))[5]` ✓
1.5

b) `int (*(*+[5])(int))[10]` ✓
1.5

2.125

Question 5 (8 points)

Compléter les 4 parties notées ***<à compléter xxx>*** du code ci-dessous de telle sorte que celui-ci affiche à l'exécution :

```
cos(0°) = 1
sin(0°) = 0
tan(0°) = 0
cos(5°) = 0.996
sin(5°) = 0.0872
tan(5°) = 0.0875
cos(10°) = 0.985
…
tan(30°) = 0.577
```

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

<à compléter 1>

#define PRINT(func, x) <à compléter 2>

<à compléter 3>

int main(void) {
   <à compléter 4>
   for (double degre = 0; degre <= 30; degre += 5) {
      for (size_t i = 0; i < TAILLE; ++i) {
         trigo[i](degre);
      }
   }
   return EXIT_SUCCESS;
}
```

Vos réponses :

<à compléter 1>

```
#include <math.h> ✓
#define TAILLE 3 ✓
```

Manque des doses

<à compléter 2>

```
void afficher_##func (double degre) {\
   printf(#func "(%g°) = %g\n", degre, func(degre));\
}
```

cf compté en double

<à compléter 3>

```
PRINT (cos, 0)
PRINT (sin, 0)
PRINT (tan, 0)
```

ok !... mais peu "intuitif"

...possible mais pas des plus "propres"

<à compléter 4>

```
void (trigo[TAILLE])(double) = { afficher_cos, afficher_sin,
                                 afficher_tan };
```

inutile

```
(*trigo[])
```

5.25

Question 6 (8 points)

Ecrire une fonction C qui prend en paramètre une matrice *n* (lignes) x *m* (colonnes) de *int* et qui livre en retour 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 l'adresse des éléments 1, 3, 7 et 9

IMPORTANT

- L'usage de VLA (*Variable Length Array*) n'est pas autorisé
- Traduire dans la fonction le fait que le(s) paramètre(s) sont supposés valides.

Votre réponse :

⚠ const!

```
int ** adresse4Coins (int* matrice, size_t nbLignes, size_t nbColones) {
    int** coins = calloc (4,

    if ( matrice == NULL || nbLignes == 0 || nbColonnes == 0 ) {
        return NULL;
    }

    int ** coins = (int**) calloc (4, sizeof(int*));
    coins[0] = & matrice [0];
    coins[1] = & matrice [nbColonnes - 1];
    coins[2] = & matrice [(nbLignes - 1) * nbColonnes+1];
    coins[3] = & matrice [(nbLignes) * nbColonnes - 1];

    return coins;
}
```

assertions!

← manque vérif alloc OK