

Examen IN102, 28 novembre 2023

Cet examen dure 2 heures.

Autorisations

- Les documents (polys, transparents, TDs, livres...) sont autorisés.
- Sont absolument interdits : le web, le courrier électronique, les messageries, les répertoires des camarades, le téléphone (même pour avoir l'heure puisque vous l'avez sur votre ordinateur).

Soumission

A la fin de l'examen, vous devrez créer une archive contenant tous les fichiers sources que vous avez écrits (.c, .h). Le nom de cette archive devra avoir la structure `nom_prenom.zip` (ou .tgz selon l'outil d'archivage que vous utilisez). Par exemple, Laura Croft nommera son archive `laura_croft.zip`. Avec l'outil zip, la commande serait

```
zip -r laura_croft.zip nom_du_dossier_a_zipper
```

Cette archive est à soumettre deux fois, comme suivant :

1. Vous le téléchargez en utilisant le formulaire à l'adresse
<https://forms.office.com/e/658GHp8HvS> (<https://forms.office.com/e/658GHp8HvS>).
2. Vous l'envoyez par mail à votre chargé de TD (comme pour les copies des TDs).

Partie 1 : Questionnaire à choix multiples

Cochez, de façon visible et non ambiguë, toutes les réponses correctes.

Q1.1

```
double x = 36 ;
double y = 5/10*(x-32) ;
printf("%.2f F = %.2f C\n", x, y) ;
```

Que va afficher le code ci-dessus ?

- ☐ 36 F = 2 C
- ☐ 36 F = 0 C

- ☐ 36.00 F = 2.00 C
- ☒ 36.00 F = 0.00 C

Q1.2

Un programme nommé `prog.c` contient entre autres les lignes de codes suivantes :

```
#include <stdio.h>
#include "paint.h"
```

Sachant que les fichiers `paint.h` et `paint.c` sont dans le même dossier que le programme ci dessus, quelle commande permet de compiler correctement ce programme:

- ☒ `gcc -o prog prog.c paint.c`
- ☐ `gcc -o prog prog.c paint.h`
- ☐ `gcc -o prog prog.c`

Q1.3

Un programme nommé `prog.c` contient entre autres les lignes de code suivantes :

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

Quelle commande permet de compiler correctement ce programme:

- ☐ `gcc -o prog prog.c math.c`
- ☐ `gcc -o prog prog.c`
- ☐ `gcc -o prog prog.c math.h`
- ☒ `gcc -o prog prog.c -lm`

Q1.4

Quel est le type de la variable `M` dans le code suivant :

```
char M[] = "d"
```

- ☐ `char`
- ☐ `int`
- ☒ `char*`
- ☐ `int*`

Q1.5

Combien d'octets de mémoire occupe la chaîne de caractère `M` dans le code suivant :

```
char M[] = "OK"
```

- ☐ 1
☐ 2
☒ 3
☐ 4

Q1.6

Considérons la fonction suivante. Cochez les réponses correctes.

```
int *create_array(int len)
{
    int array[len];
    for (int i = 0; i < len; i++)
        array[i] = 0;
    return array;
}
```

- ☒ La fonction initialise les cases du tableau `array` à 0.
☒ La fonction crée un tableau `array`.
☒ Le tableau n'est plus accessible une fois la fonction `create_array` terminée.

Q1.7

On souhaite allouer une zone mémoire dont la taille permet de stocker précisément un tableau de 10 `char`. Cochez les réponses correctes.

- ☒ `char *tab = malloc(sizeof(char) * 10);`
☒ `char *tab = malloc(10);`
☐ `char *tab = malloc(256 * 10);`

Q1.8

Qu'affiche le code suivant ?

```
int main()
{
    int i = 5;
    int j = 5;
    printf("i = %d, j = %d\n", i++, ++j);
    return 0;
}
```

- ☐ `i = 5, j = 5`
☐ `i = 6, j = 5`
☒ `i = 5, j = 6`
☐ `i = 6, j = 6`

Q1.9

Qu'affichera l'exécution du code suivant?

```
#include <stdio.h>

void foo(int n){
    for(int i = 0, i < n; i += 2){
        n++;
    }
    return;
}

int main(){
    int n = 10;
    foo(n);
    printf("%d\n", n);
    return;
}
```

- ☒ 10
☐ 15
☐ 20
☐ 25

Q1.10

On considère le programme suivant :

```

struct Eleve{
    char nom[20];
    char prenom[20];
    int age;
}

int main(){
    struct Eleve John;
    struct Eleve *J = &John;
    // ... à compléter ici ...
    return 0;
}

```

Parmi les propositions suivantes, lesquelles permettent de définir l'âge de John à 20:

- ☐ John->age = 20
- ☒ J->age = 20
- ☒ (*J).age = 20
- ☒ (&John)->age = 20

Q1.11

Considérez le code suivant :

```

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

void empile(int x, int* L) {
    int temp = L[0];
    L[0] = L[1];
    L[1] = x;
    x = temp;
}

int main() {
    int L[2] = {1, 2};
    int x = 3;
    empile(x, L);

    printf("%d, %d, %d", x, L[0], L[1]);
}

```

Quelle sera la sortie du programme après l'exécution du code ci-dessus?

- ☐ 1, 2, 3
- ☐ 3, 1, 2
- ☒ 3, 2, 3

- ☐ 2, 3, 1

Q1.12

Quelle sera la sortie du programme après l'exécution du code suivant :

```

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

int main(void)
{
    int i;
    int *ptr = (int *) malloc(4 * sizeof(int));

    for (i=0; i<4; i++)
        *(ptr + i) = i * 2;

    printf("%d ", *ptr++);
    printf("%d ", *ptr + 1);
    printf("%d ", (*ptr)--);
    printf("%d ", *(++ptr) + 2);
}

```

- ☐ 0 3 3 6
- ☒ 0 3 2 6
- ☐ 0 2 3 8
- ☐ 0 4 2 8