

Examen info4A - année 2022/2023 - session 1

Partie langage C

Vous devez écrire vos réponses sur la fiche de réponse qui vous a été fournie. Votre nom doit être écrit sur la partie cachée de la copie double. Votre numéro d'anonymat officiel doit être écrit sur la copie double et sur la fiche de réponse.

Durée indicative : 1h30.

Documents autorisés (conjointement avec la partie C++) : 4 feuilles A4 recto-verso, avec contenu libre imprimé et/ou manuscrit.

Calculatrices et ordinateurs interdits.

Partie A (2 points)

Question 1 (2 points)

Définissez la fonction...

```
void printDur(int h1, int m1, int h2, int m2);
```

...qui affiche en heures et minutes la somme des durées représentées en heures et minutes par `h1`, `m1` et `h2`, `m2`.

On suppose que `m1` et `m2` on des valeurs comprises entre 0 et 59.

Par exemple, l'appel...

```
printDur(3,37,4,43);
```

...produit l'affichage 8h20, car la somme de 3 heures 37 minutes et de 4 heures 43 minutes représente 8h et 20 minutes.

Partie B (3 points)

Question 2 (1.5 point)

Donnez les valeurs affichées par l'exécution des lignes de code suivantes :

```
printf("%d\n", 0b00001111 & 0b00111100);  
printf("%x\n", (2<<2)|1);  
printf("%x\n", 0xFF00 & 0x0330);
```

Question 3 (1.5 point)

Sans utiliser de boucle, définissez la fonction...

```
uint16_t f(int k1, int k2);
```

...qui produit un mot binaire de 16 bits dont les k_1 premiers bits en partant de la gauche sont à 1, puis les k_2 bits suivants sont à 0, puis les $16 - k_1 - k_2$ derniers bits sont à 1.

On suppose que $k_1 + k_2$ est inférieur ou égal à 16. Aucune vérification n'est demandée.

Par exemple, si k_1 vaut 5 et k_2 vaut 8, alors la représentation binaire de la valeur de retour doit être :

```
111110000000111
```

Partie C et E (5 points)

Soit `VMAX` une constante globale. Dans la suite, on considérera que cette constante a la valeur 7, mais les réponses aux questions devront rester correctes si on change cette valeur.

```
#define VMAX 7
```

On représente un ensemble d'entiers compris entre 0 et `VMAX` par un tableau de `VMAX+1` cellules contenant chacune un Booléen représenté par un entier pouvant avoir pour valeur 0 ou 1.

Voici quelques exemples d'ensembles et leurs représentations avec la convention utilisée :

Ensemble	Représentation par un tableau d'entiers
{}	0, 0, 0, 0, 0, 0, 0, 0
{0, 1, 2, 3}	1, 1, 1, 1, 0, 0, 0, 0
{0, 1, 2, 3, 4, 5, 6, 7}	1, 1, 1, 1, 1, 1, 1, 1

Question 4 (1.5 point)

Définissez la fonction...

```
int card(int e[]);
```

...qui retourne le cardinal, c'est à dire le nombre d'éléments, de l'ensemble représenté par le tableau désigné par le paramètre `e`.

Question 5 (1.5 point)

Donnez les lignes de code permettant :

- De créer dans la pile l'ensemble {1, 2, 3, 5}.
- D'utiliser la fonction `card` pour calculer le cardinal de cet ensemble et d'afficher le résultat.

Question 6 (1 point)

Complétez la définition de la fonction suivante, qui crée dans le tas un ensemble représentant l'intersection des deux ensembles désignés par les paramètres e1 et e2 .

```
int* inter(int e1[], int e2[])
{
    int* r = ..... ;
    for(int i=0; i <=VMAX; i++)
    {
        if(e1[i] && e2[i]) r[i] = 1;
    }
    return r;
}
```

Donnez juste la ligne manquante sur la feuille de réponse.

Question 7 (1 point)

On suppose que les variables `a` et `b`, de type `int*`, désignent deux ensembles situés dans le tas. Donnez les lignes de code permettant :

- De calculer l'intersection de ces deux ensembles en utilisant la fonction `inter`.
- De calculer avec la fonction `card` et d'afficher le cardinal de cette intersection.
- De libérer la mémoire occupée dans le tas par les deux ensembles désignés par `a` et `b` et par l'ensemble représentant leur intersection.

Partie D et F (5 points)

On représente un point par la structure suivante :

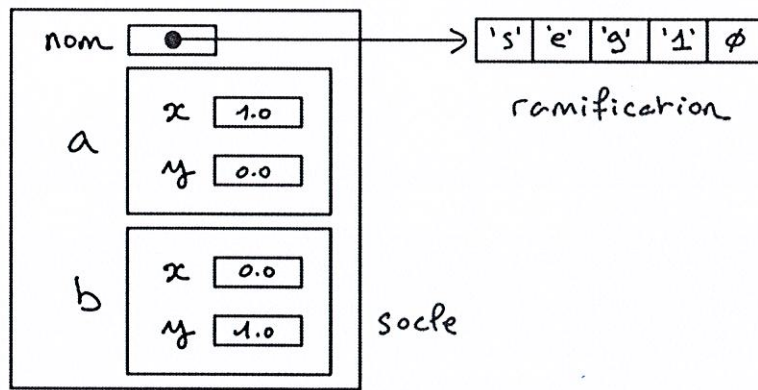
```
typedef struct
{
    double x;
    double y;
}point;
```

On appelle segment nommé l'association de deux points (qui représentent les extrémités d'un segment) et d'une chaîne de caractères représentant le nom du segment.

Un segment nommé est représenté par la structure suivante :

```
typedef struct
{
    char* nom;
    point a;
    point b;
}segn;
```

Voici un exemple de représentation en mémoire d'un segment nommé.



Question 8 (2 points)

Vous devez définir la fonction...

```
segn* makeSegn(const char* nom);
```

...qui crée un segment nommé situé entièrement dans le tas. Le nom de ce segment doit être une copie (situées dans le tas) de la chaîne désignée par le paramètre. Les deux extrémités de ce segment doivent avoir pour coordonnées (0.0, 0.0).

Question 9 (2 points)

Donnez les lignes de code permettant :

- De créer un segment nommé "seg" situé dans le tas par un appel de la fonction makeSeg.
- De modifier les coordonnées de la première extrémité du segment en leur donnant les valeurs (1.0, 2.0).
- De modifier le nom du segment. Le nouveau nom doit être "segbis".

Question 10 (1 point)

Définissez la fonction...

```
void delSegn(segn* s);
```

...qui libère toute la mémoire occupée par le segment désigné par le paramètre s. On suppose que ce segment a été créé par la fonction makeSeg.

Examen – Info4A

Partie programmation C++

Numéro anonymat : _____

Question 1 : (2 points) :

1. Quel opérateur, un objet d'une classe utilise pour accéder à ses variables membres et à ses fonctions membres ?

- :
 →
 .
 ::

2. Quelle est la syntaxe correcte pour accéder à une méthode `operation()` d'une classe `A` ?

- `A->operation()`
 `A::operation()`
 `A.operation()`
 `A^operation()`

3. Lorsque certaines données membres sont ou contiennent des pointeurs, à quoi faut-il faire attention lors de la copie d'une instance ?

4. Lorsqu'on surcharge une fonction, on peut modifier :

- le type de sa valeur de retour
 son nom
 les types de ses arguments

Question 2 : (3 points) :

1. Créer une classe pour représenter un rectangle, avec des attributs pour sa largeur et sa hauteur, et des constructeurs sans et avec arguments
2. Ajouter des méthodes pour calculer son périmètre et son aire.
3. Écrire un programme qui crée des instances de la classe `Rectangle`, les initialise avec différentes valeurs de largeur et de hauteur, et affiche leur périmètre et leur aire.
4. Ajouter une méthode pour déterminer si deux rectangles ont la même aire, et utiliser cette méthode pour trouver tous les rectangles qui ont la même aire parmi les instances créées précédemment.

Indications :

Le périmètre d'un rectangle est la somme de ses quatre côtés :

$$P = 2(\text{largeur} + \text{hauteur})$$

L'aire d'un rectangle est le produit de sa largeur et de sa hauteur :

$$A = \text{largeur} \times \text{hauteur}$$

Il faut donner les fichiers `.h`, `.cpp` et le `main`