

Examen Février 2013

Prénom : Nom :
Formation : ☐ Électronique ☐ Télécom

Vous avez exactement **2h30** pour répondre à toutes les questions de cet examen. Vous n'avez le droit à rien si ce n'est de quoi écrire, ainsi que le formulaire « *Aide-mémoire du langage C* » annoté de manière raisonnable. N'oubliez pas d'écrire vos nom et prénom de manière lisible sur la première page.

Bonne chance !

1 Extraits de code

Pour chacun des extraits de code suivants, vous devez indiquer ce que son exécution affiche à l'écran.

```
int i = 1;
while (i > -5)
{
    i--;
    printf ("%d", -i);
}
```

Q1(a) : _____

```
int j = 1;
for (j = 4; j < 20; j += 5)
{
    printf ("%d;", j + 1);
}
```

Q1(b) : _____

```
int tab[] = {1, 2, 3, 4, 5};
printf ("Valeur : %d", tab[tab[tab[4] - 2] / 2]);
```

Q1(c) : _____

2 Population de nénuphars

Nous allons étudier l'évolution d'une population de nénuphars. Tous les ans, la population totale de nénuphars double. Par exemple, supposons qu'il y ait 10 nénuphars dans un lac, initialement. Après un an, il y aura 20 nénuphars. Après deux ans, il y en aura 40. Et ça continue suivant la même logique : 80, 160... Vous devez écrire une fonction qui calcule combien d'années au minimum il faut attendre, pour obtenir au moins “`target`” nénuphars, sachant qu'on démarre initialement avec “`init`” nénuphars.

```
int nbYears (int init, int target)
{
```

```
}
```

3 Lecture de fichier

Soit un fichier texte contenant, sur chacune de ses lignes, un entier. La fonction suivante, que vous devez compléter, lit le fichier ligne par ligne et cherche le plus grand nombre dans le fichier, sachant qu'on considère la valeur absolue du nombre lu. Elle renvoie cette valeur absolue maximale. La fonction suppose que le fichier existe bel et bien sur l'ordinateur.

```
int max (int a, int b);
int abs (int a);

int maxAbs (char *path)
{
    FILE *file = fopen (path, "r");

    char buffer[80];
    int maxVal = 0;

    while ( Q3(a) )
    {
        Q3(b)

    }

    Q3(c)

    return maxVal;
}
```

Q3(a) : _____

Q3(b) :

Q3(c) : _____

4 Mémoire et pointeurs

Soit la variable `int **tab` et la situation suivante en mémoire :

	...
tab : 1000	2000
	...
2000	4000
	5000
	...
4000	8
	3
	...
5000	16
	7
	-2
	...

Quelles sont les valeurs des expressions suivantes :

1. `tab`
2. `&tab`
3. `*tab`
4. `tab[0]`
5. `tab + 1`
6. `*(tab + 1)`
7. `tab[1][2]`
8. `&(tab[1])`
9. `*(tab + 1) + 2`
10. `*(*(tab + 1) + 2)`

5 Ticket de caisse

Soit la structure suivant représentant un article de magasin :

```

struct item {
    char *name;
    float price;
    int reduc;
};
typedef struct item ITEM;
  
```

Écrivez une fonction permettant de créer un nouvel article, et qui renvoie un pointeur `ITEM*`.

```

ITEM* createItem (char *name, float price, int reduc)
{

}
  
```

Écrivez une fonction qui reçoit en paramètre une liste de `ITEM` (attention, les éléments de la liste ne sont donc pas des pointeurs) de taille `N` et qui calcule le prix total, prenant en compte la réduction.

```

float totalPrice (ITEM *items, int N)
{

}
  
```

Enfin, on s'intéresse à la fonction `main`. Celle-ci doit déclarer et initialiser une liste de deux articles :

- *DVD Titanic*, 19.99 €, 50% de réduction
- *Boite de chocolats*, 24.5 €, 15% de réduction

Et la fonction doit calculer le prix total et l'afficher à l'écran.

```

int main()
{

}
  
```

BONUS : Définissez une procédure `printItem` qui permet d'afficher un `ITEM` à l'écran.