

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 M = 6;
int k = M;
while (k > 0)
{
    printf ("%d", M - k + 1);
    k--;
}
```

Q1(a) : _____

```
int p = 1;
for (p = 7; p >= 1; p -= 2)
{
    printf (":%d", (p - 1) / 2);
}
```

Q1(b) : _____

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

Q1(c) : _____

2 Épargne bancaire

Nous allons étudier l'évolution d'argent sur un compte bancaire. Tous les ans, à la fin de l'année, des intérêts sont calculés et ajoutés sur le compte. Par exemple, si on a initialement 10 euros, et que le taux d'intérêts est de 50%, après un an, on aura $10 \times 50\% = 5$ euros d'intérêt et donc 15 euros sur le compte. Après deux ans, on aura $15 \times 50\% = 7,5$ euros d'intérêt et donc 22,5 euros sur le compte. Vous devez écrire une fonction qui calcule combien d'années au minimum il faut attendre, pour obtenir au moins "target" euros, sachant qu'on démarre initialement avec "init" euros et que le taux d'intérêt est de "interest".

```

int nbYears (int init, int interest, 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 uniquement les nombres positifs et pair. Elle renvoie cette valeur maximale. La fonction suppose que le fichier existe bel et bien sur l'ordinateur.

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

int maxPosEven (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	5000
	4000
	...
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 + 2`
6. `*(tab + 2)`
7. `tab[1][2]`
8. `&(tab[1])`
9. `*(tab + 1) + 2`
10. `*(*(tab + 1) + 2)`

5 Ticket de caisse

Soit la structure suivant représentant une ligne d'un ticket de caisse pour un article de magasin :

```

struct item {
    char *name;
    float price;
    int quantity;
};
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 quantity)
{

}
  
```

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 le nombre d'articles achetés.

```

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 :

- *Oasis Framboise*, 3.99 €, 6 bouteilles achetées
- *Bloc de 100 feuilles quadrillées*, 1.20 €, 2 blocs achetés

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.