

Examen Février 2014

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

Q1(a) : _____

```

int m;
for (m = 32; m >= 1; m /= 2)
{
    printf ("%d=", m);
}
  
```

Q1(b) : _____

```

int tab[] = {5, 4, 3, 2, 1};
printf ("Valeur : %d", tab[tab[tab[0] - tab[3]] + 1]);
  
```

Q1(c) : _____

2 Moyenne des températures négatives

Écrivez une fonction qui calcule la moyenne des températures négatives du tableau “tab” qui contient “N” cases. **Faites donc bien attention de ne prendre en compte que les valeurs négatives dans le tableau.**

Par exemple, pour le tableau $\{-1, -2, 3, -4, 5\}$, le programme doit calculer $(-1) + (-2) + (-4) = -7$ qu’il faut diviser par 3 pour avoir la moyenne. La fonction doit donc renvoyer 2,33...

```
float sumTemperatures (int *tab, int N)
{
```

```
}
```

3 Mémoire et pointeurs

Soit la variable `int **tab` et la situation suivante en mémoire (pour cet exercice, on considère qu'un `int` occupe 8 octets en mémoire).

	...
tab : 2000	4000
	...
4000	5000
	6000
	...
5000	-4
	6
	...
6000	-9
	0
	3
	...

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][1]`
8. `&(tab[1])`
9. `*(tab + 1) + 2`
10. `*(*(tab + 1) + 2)`

4 Programme TV

Soit la structure suivant représentant une émission de télé :

```

struct tvshow {
    char *name;           // nom de l'émission
    char *channel;        // nom de la chaîne
    int duration;         // durée en minutes
};
  
```

Écrivez une fonction permettant de créer une nouvelle émission de télé.

```

struct tvshow* createTVShow (char *name, char *channel, int duration)
{

}
  
```

Écrivez une fonction qui reçoit en paramètre une émission de télé et qui teste si sa durée est plus grande que la durée `max` également reçue en paramètre.

```

bool isTVShowLonger (struct tvshow *show, int max)
{

}
  
```

Écrivez une fonction qui reçoit en paramètre une liste de `struct tvshow` (attention que les éléments de la liste ne sont donc pas des pointeurs) de taille `N` et qui calcule la durée totale de toutes les émissions, en minutes.

```

int totalDuration (struct tvshow *items, int N)
{

}

```

Écrivez une fonction qui reçoit en paramètre deux listes de `struct tvshow` sous forme d'un tableau à deux cases (la première liste a `N` éléments et la seconde a `M` éléments). La fonction doit renvoyer `TRUE` si la durée totale de la première liste est plus longue que celle de la seconde et `FALSE` sinon.

```

int isProgramLonger (struct tvshow **items, int N, int M)
{

}

```

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

- *Titanic* sur *RTL-TVI*, durée de 194 minutes
- *The Lego Movie* sur *TF1*, durée de 100 minutes

La fonction `main` doit construire les deux `struct tvshow`, puis faire une liste avec ces deux éléments. Ensuite, elle doit tester si la durée totale des deux émissions dépasse 5 heures.

```

int main()
{

}

```