

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 i = 0;
while (i > -5)
{
    i--;
    printf ("%d", i);
}
  
```

Q1(a) : _____

```

int j;
for (j = 1; j < 10; j += 2)
{
    printf ("%d;", j / 2);
}
  
```

Q1(b) : _____

```

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

Q1(c) : _____

2 Moyenne des températures positives

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

Par exemple, pour le tableau $\{1, 2, 3, -4, 5\}$, le programme doit calculer $1 + 2 + 3 + 5 = 11$ qu’il faut diviser par 4 pour avoir la moyenne. La fonction doit donc renvoyer 2,75.

```
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 : 1000	3000
	...
3000	7000
	6000
	...
6000	-2
	0
	4
	...
7000	2
	9
	...

Quelles sont les valeurs des expressions suivantes :

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

```
{
```

```
}
```