

Examen Aout 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 < 5)
{
    k++;
    printf ("%d", k - 2);
}
  
```

Q1(a) : _____

```

int e;
for (e = 1; e < 5; e += 1)
{
    printf ("%d:", e * 2);
}
  
```

Q1(b) : _____

```

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

Q1(c) : _____

2 Point maximum

Écrivez une fonction qui trouve la plus grande note dans le tableau “`tab`” qui contient “`N`” cases. **Faites bien attention qu’il ne faut prendre en compte que les valeurs comprises entre 0 et 20.**

Par exemple, pour le tableau $\{4, 12.5, -4, 21.3, 19.8\}$, le programme doit renvoyer 19.8. La valeur -4 est ignorée car plus petite que 0 et la valeur 21.3 est ignorée car plus grande que 20.

```
float maxPoints (float *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	4000
	...
4000	8000
	9000
	...
8000	5
	2
	...
9000	-5
	0
	99
	...

Quelles sont les valeurs des expressions suivantes :

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

4 Liste de personnes

Soit la structure suivante représentant une personne :

```

struct person {
    char *name;           // nom de la personne
    int age;              // âge de la personne
};
  
```

Écrivez une fonction permettant de créer une nouvelle personne.

```

struct person* createPerson (char *name, int age)
{

}
  
```

Écrivez une fonction qui reçoit en paramètre une personne et qui teste si elle est majeure (c'est-à-dire que son âge est plus grand ou égal à 18).

```

bool isAdult (struct person *p)
{

}
  
```

Écrivez une fonction qui reçoit en paramètre une liste de `struct person` (attention que les éléments de la liste ne sont donc pas des pointeurs) de taille `N` et qui renvoie l'âge de la plus jeune personne.

```

int minAge (struct person *list, int N)
{

}
  
```

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

- *Arnaud Van Roy*, 12 ans
- *Andy Von Pemak*, 78 ans
- *Peter Géronils*, 7 ans

La fonction `main` doit construire les trois `struct person`, puis faire une liste avec ces trois éléments. Ensuite, elle doit appeler la fonction `minAge` pour trouver puis afficher l'âge de la plus jeune personne.

```

int main()
{

```

```

}
```