

## Examen Février 2012

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;
for (j = 10; j > 0; j -= 2)
{
    printf ("%d", j);
}
  
```

Q1(b) : \_\_\_\_\_

```

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

Q1(c) : \_\_\_\_\_

## 2 Trouver le chiffre des dizaines

Vous devez écrire une fonction dont le but est d'extraire le chiffre des dizaines d'un nombre entier donné. La fonction reçoit donc un entier  $N$  en paramètre et renvoie un entier compris entre  $-9$  et  $9$  correspondant au chiffre des dizaines (ou à son opposé si  $N$  est négatif) . Par exemple, pour le nombre 9384, la fonction doit renvoyer 8. Et pour le nombre  $-267$ , la fonction doit renvoyer  $-6$ . Enfin, pour le nombre 2, la fonction doit renvoyer 0.

```
int getDecadeDigit (int N)
```

```
{
```

```
}
```

### 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 calcule la somme des entiers de chacune des lignes du fichier. Elle renvoie la valeur de cette somme. La fonction suppose que le fichier existe bel et bien sur l'ordinateur.

```
int computeSum (char *path)
{
    FILE *file = fopen (path, "r");
    char buffer[80];
    int total = 0;
    while ( Q2(a) )
    {
        total += Q2(b) ;
    }
    Q2(c)
    return total;
}
```

Q3(a) : \_\_\_\_\_

Q3(b) : \_\_\_\_\_

Q3(c) : \_\_\_\_\_

**BONUS :** Si le fichier est vide (le fichier existe bel et bien, mais ne contient aucune ligne), quelle sera la valeur calculée par la fonction ?

\_\_\_\_\_

## 4 Calcul de temps

On s'intéresse à un programme qui permet de gérer des horaires de train. Pour cela, on va définir une structure qui représente une heure (que ce soit une heure de départ ou une heure d'arrivée d'un train). Voici la définition de la structure :

```

struct time {
    int hour;
    int minute;
};

```

On s'intéresse maintenant à pouvoir calculer le temps de parcours d'un train, connaissant l'heure de départ ainsi que l'heure d'arrivée (on suppose que le train démarre et arrive le même jour à destination). Écrivez une fonction qui permet de calculer ce temps de parcours, **en minutes**.

**Attention**, vous n'avez aucune garantie quant à l'ordre des paramètres. La variable `t1` pourrait aussi bien être l'heure de départ que celle d'arrivée (et inversement pour `t2`). Votre fonction doit pouvoir être utilisée dans les deux cas.

```

int timeDifference (struct time *t1, struct time *t2)
{

}

```

Pour tester votre fonction, vous décidez d'écrire une fonction `main` qui va calculer le temps de trajet du train L6266 entre les gares de Louvain-la-Neuve Université et Ottignies. Le train part à 16h25 et arrive à 16h36. Complétez la fonction `main` qui est proposée.

```

int main()
{
    struct time a, b;
    Q4(b)

    printf ("Temps du trajet : %d min\n", Q4(c) );

    Q4(d)
}
  
```

Q4(b) : \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Q4(c) : \_\_\_\_\_

Q4(d) : \_\_\_\_\_

## 5 Temps de trajet d'un voyage en train

Parfois, lorsqu'on fait un voyage en train, on prend plusieurs trains et on a des correspondances. Par exemple, pour rejoindre Schaerbeek depuis Louvain-la-Neuve, on effectue le trajet suivant :

<b>Départ de</b>	Louvain-la-Neuve Université	Ottignies	Bruxelles-Nord
<b>à</b>	16h25	16h47	17h27
<b>Arrivée à</b>	Ottignies	Bruxelles-Nord	Schaerbeek
<b>à</b>	16h36	17h18	17h30

On peut facilement représenter un trajet de cette forme par un tableau à deux-dimensions comportant **toujours** deux lignes et autant de colonnes qu'il y a d'étapes au trajet. Pour l'exemple, on aurait :

$$journey = \begin{pmatrix} 16h25 & 16h47 & 17h27 \\ 16h36 & 17h18 & 17h30 \end{pmatrix}$$

En C, vous auriez un tableau à deux dimensions dont les éléments sont des pointeurs vers des éléments de type `struct time`. Vous devez écrire une fonction qui va calculer le temps passé dans les trains, pour un trajet donné, en minutes. Pour l'exemple, la fonction doit calculer **45** (11 + 31 + 3).

```

int totalTime (struct time ***journey, int columns)
{

}

```

La variable `columns` donne donc le nombre de trajets intermédiaires. Remarquez qu'il y a trois `*` pour la variable `journey`. Il y a en effet deux `*` puisque c'est un tableau à deux dimensions et il y en a un troisième car les éléments du tableau sont des pointeurs. Pour information, voici la méthode `main` utilisée pour tester l'exemple donné.

```

int main()
{
    struct time a, b, c, d, e, f;
    a.hour = 16; a.minute = 25; b.hour = 16; b.minute = 36; c.hour = 16; c.minute = 47;
    d.hour = 17; d.minute = 18; e.hour = 17; e.minute = 27; f.hour = 17; f.minute = 30;

    struct time ***matrix = malloc (2 * sizeof (struct time*));
    matrix[0] = malloc (3 * sizeof (struct time*));
    matrix[1] = malloc (3 * sizeof (struct time*));

    matrix[0][0] = &a; matrix[0][1] = &c; matrix[0][2] = &e;
    matrix[1][0] = &b; matrix[1][1] = &d; matrix[1][2] = &f;

    printf ("Temps total : %d min\n", totalTime (matrix, 3));
    return 0;
}

```

**BONUS :** Modifiez la fonction `totalTime` pour non plus calculer seulement le temps passé dans les trains, mais le temps total du trajet depuis le premier départ, jusqu'à la dernière arrivée, en comptabilisant donc en plus tout le temps passé à attendre les correspondances.

```

int totalTime (struct time ***journey, int columns)
{

}

```