

Examen Aout 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 k = 4;
while (k > -1)
{
    printf ("%d", k);
    k--;
}
  
```

Q1(a) : _____

```

int m;
for (m = 1; m <= 20; m *= 2)
{
    printf ("%d", m);
}
  
```

Q1(b) : _____

```

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

Q1(c) : _____

2 Borner les valeurs d'un tableau

Vous devez écrire une procédure dont le but est de faire en sorte que toutes les valeurs d'un tableau se retrouvent bornées entre une valeur *min* et une autre valeur *max*. Votre procédure va parcourir tous les éléments du tableau, et si un élément est strictement plus petit que *min*, l'élément doit être remplacé par *min*. De même, si l'élément est strictement plus grand que *max*, il doit être remplacé par *max*.

Par exemple, soit le tableau $[12, -4, -12, 0, 4, 99]$ et soit les valeurs $min = -10$ et $max = 25$. Le tableau, une fois traité par la procédure, va contenir $[12, -4, -10, 0, 4, 25]$.

```
void boundArray (int *tab, int N, int min, int max)
{
```

```
}
```

3 Écriture dans un fichier

La procédure suivante, que vous devez compléter, écrit dans un fichier texte, les nombres de 1 à N , avec un nombre par ligne.

```

void writeIntegers (char *path, int N)
{
    FILE *file = Q3(a) ;
    int i = 0;
    while ( Q3(b) )
    {
        Q3(c) ;
        i++;
    }
    Q3(d) ;
}
  
```

Q3(a) : _____

Q3(b) : _____

Q3(c) : _____

Q3(d) : _____

Comment ferait-on pour créer un fichier `data.txt` contenant les 12 premiers entiers ?

```

int main()
{
    Q3(e)
    return 0;
}
  
```

Q3(e) : _____

4 Calcul de distance

On s'intéresse à un programme qui permet de calculer la distance parcourue lors d'un jogging. Pour cela, on va définir une structure qui permet de représenter une coordonnée géographique (une position en x et une position en y). Voici la définition de la structure :

```
struct position {
    int x;
    int y;
};
```

On s'intéresse maintenant à pouvoir calculer la distance d entre deux positions. Écrivez une fonction qui permet de calculer cette distance, donnée par la formule suivante pour (x_1, y_1) et (x_2, y_2) :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
double distance (struct position *p1, struct position *p2)
{

}

}
```

Écrivez une fonction `main` qui va calculer la distance entre les deux points $(1, 2)$ et $(8, 12)$:

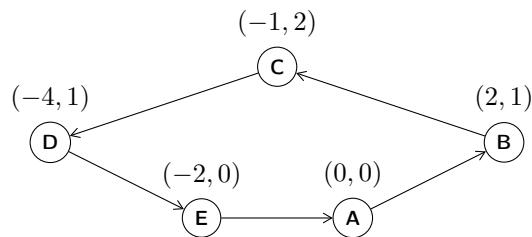
```
int main()
{

}

}
```

5 Distance d'un parcours

Dans la question précédente, on a défini une fonction permettant de calculer la distance entre deux points. Cette fois-ci, on s'intéresse à la distance totale d'un parcours complet. Le parcours est défini par plusieurs positions. Voici par exemple un parcours (qui commence en A) :



On peut facilement représenter un parcours de cette forme par un tableau à une dimension dont les éléments sont des pointeurs vers des éléments de type `struct position`. Vous devez écrire une fonction qui va calculer la distance totale du parcours, sachant qu'on revient toujours au point de départ.

```

double totalDistance (struct position *tour, int N)
{

}
  
```