

Examen Janvier 2015

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 (3 points)

Pour chacun des extraits de code suivants, vous devez indiquer ce que son exécution affiche à l'écran.

```

int e = 10;
while (e * e > 10)
{
    e--;
    printf ("%d", e + 1);
}
  
```

Q1(a) : _____

```

int z;
for (z = 1; z * z < 100; z *= -2)
{
    printf ("%d=", z);
}
  
```

Q1(b) : _____

```

int tab[] = {5, 10, 15, 20, 25};
printf ("Valeur : %d", tab[tab[tab[2] / tab[0]] / 5]);
  
```

Q1(c) : _____

2 Rapport positifs/négatifs (5 points)

Écrivez une fonction qui calcule le rapport entre le nombre de valeurs positives d'un tableau "tab" et le nombre de valeurs négatives. Par exemple, pour le tableau $\{1, -5, 2, -1, 0\}$, le programme doit calculer le rapport entre 3 (le nombre de valeurs positives) et 2 (le nombre de valeurs négatives), c'est-à-dire 3 divisé par 2, ce qui vaut 1.5. Remarquez donc qu'on considère 0 comme étant un nombre positif.

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

```
}
```

3 Représentation des données et pointeurs (6 points)

1. Donnez la représentation en bit de signe sur 7 bits des nombres décimaux -22 et 17 .

2. Calculez, en passant par la représentation en complément à deux sur 8 bits, $11 - 19$ (donnez les détails du calcul).

3. Qu'affiche le code suivant après exécution ?

```

int data[] = {1, 2, 3, 4, 5};

int m = 0;
int *q = data;
while (q < data + 4)
{
    m += *q;
    q++;
}

printf ("Somme : %d", m);
  
```

Q3(c) : _____

4 Mémoire et pointeurs (5 points)

Soit la variable `short **data` et la situation suivante en mémoire (pour rappel, un `short` occupe 2 octets en mémoire et un pointeur occupe 8 octets en mémoire).

| | |
|-------------|------|
| | ... |
| 3000 | 9000 |
| | 8000 |
| | ... |
| data : 7000 | 3000 |
| | ... |
| 8000 | 4 |
| | -2 |
| | 1 |
| | ... |
| 9000 | 4 |
| | 1 |
| | ... |

Quelles sont les valeurs des expressions suivantes :

1. `data`
2. `*data`
3. `&data`
4. `data[1]`
5. `data + 2`
6. `*(data + 1)`
7. `data[1][2]`
8. `&(data[1][0])`
9. `*(data + 1) + 1`
10. `*(*(data + 1) + 1)`

5 Compagnie aérienne (11 points)

Soit la structure suivant représentant un avion dans un compagnie aérienne dont on donne le nom et la masse maximale au décollage stockée en kilogrammes :

```

struct plane {
    char *name;           // Nom du modèle de l'avion
    int mtow;             // Masse maximale au décollage en kilogrammes
};
  
```

Écrivez une procédure qui reçoit un avion en paramètre et qui permet de l'afficher. Par exemple, la procédure doit afficher :

Boeing 747-8I (448.00 tonnes)

```

void printPlane (struct plane d)
{

}
  
```

Écrivez une fonction qui construit un nouvel avion à partir des deux paramètres `n` et `m` représentant le nom de l'avion et son poids maximal au décollage en tonnes.

```

struct plane* createPlane (char *n, float m)
{

}
  
```

Enfin, complétez la fonction `main` suivante, de manière à ce qu'elle affiche l'avion qui est créé et stocké dans la variable `p`.

```

int main()
{
    struct plane *p = createPlane ("Boeing 747-8I", 448.0);

    return 0;
}

```