ESPACE ENSEIGNEMENT CARDINAL MERCIER

mercredi 28 janvier 2015
Sébastien Combéfis

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 p = 0;
while (p + 1 <= 5)
{
    p++;
    printf ("%d", p);
}
```

Q1(a) : _____

```
int t;
for (t = 8; -t < -1; t /= 2)
{
    printf ("%d,", t);
}
```

Q1(b) : _____

```
int tab[] = {2, 4, 6, 8, 10};
printf ("Valeur : %d", tab[tab[tab[3] / tab[1]] - 2]);
```

Q1(c) : _____

2 Rapport positifs/négatifs (5 points)

Écrivez une fonction qui calcule le rapport entre la somme des valeurs positives d'un tableau "`tab`" et la somme de ses valeurs négatives. Par exemple, pour le tableau $\{1, -5, 2, -1, 0\}$, le programme doit calculer le rapport entre $1 + 2 + 0 = 3$ (la somme des valeurs positives) et $(-5) + (-1) = -6$ (la somme des valeurs négatives), c'est-à-dire 3 divisé par -6, ce qui vaut -0.5.

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









}
```

3 Représentation des données et pointeurs (6 points)

1. Donnez la représentation en bit de signe sur 5 bits des nombres décimaux -7 et 12.

2. Calculez, en passant par la représentation en complément à deux sur 8 bits, $6 - 11$ (donnez les détails du calcul).

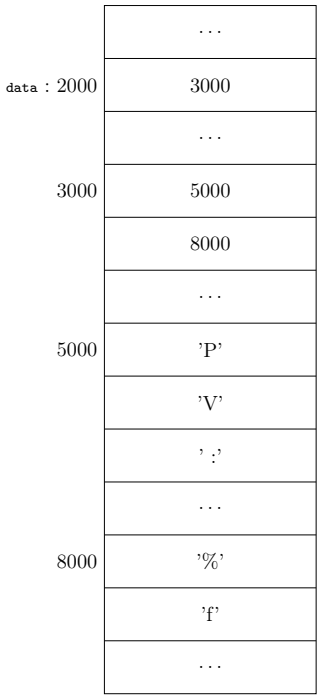
3. Qu'affiche le code suivant après exécution ?

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

int s = 0;
int *p = data + 4;
while (p >= data)
{
    s += *p;
    p -= 2;
}

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

Q3(c) : _____

4 Mémoire et pointeurs (5 points)

Soit la variable **char** `**data` et la situation suivante en mémoire (pour rappel, un **char** occupe 1 octet en mémoire et un pointeur occupe 8 octets en mémoire).



Quelles sont les valeurs des expressions suivantes :

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

5 Menu de restaurant (11 points)

Soit la structure suivant représentant un plat dans un restaurant dont on donne le nom et le prix stocké en eurocents :

```
struct dish {
   char *name;        // Nom du plat
   int price;         // Prix du plat en eurocents
};
```

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

```
Spaghetti carbonara (12.50 euros)
```

```
void printDish (struct dish d)
{









}
```

Écrivez une fonction qui construit un nouveau plat à partir des deux paramètres `n` et `p` représentant le nom du plat et son prix en euros.

```
struct dish* createDish (char *n, float p)
{









}
```

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

```
int main()
{
    struct dish *d = createDish ("Spaghetti carbonara", 12.50);









    return 0;
}
```