

# PROG L3-MI

## DS

2020-10-06

*Consignes.* La durée de ce devoir surveillé est d'une heure trente. Aucun document n'est autorisé. Dans tous les exercices, il est possible d'admettre le résultat d'une question pour répondre aux questions suivantes. Toute réponse doit être justifiée.

### Exercice 1 : Questions courtes (environ 8 points)

**Q.1 :** Pour chaque déclaration suivante, quel est le type de la variable déclarée ? Dans chaque cas, que pouvez-vous dire sur sa valeur après déclaration ?

1. `int a = -1;`
2. `unsigned a = 1;`
3. `float a = 7/5;`
4. `double a = 0xA;`

**Q.2 :** Pour chacune des déclarations ou suites d'expressions suivantes, décrivez le résultat attendu et si celui-ci mène à une erreur ou un comportement possiblement non souhaité.

1. `int8_t t[128]; t[127] = 128;`
2. `int8_t t[128]; t[128] = 127;`
3. `float x = 16777217; // 16777217 == 2**24 + 1`
4. `uint8_t a = -1;`

**Q.3 :** Quel problème pose l'extrait de code suivant ? Comment pouvez-vous y remédier ?

```
1 for (uint8_t i = 0; i < 997; i++)
2 {
3     ...
4 }
```

**Q.4 :** Quel problème pose l'extrait de code suivant ? Comment pouvez-vous y remédier ?

```
1 for (uint32_t i = 997; i >= 0; i--)
2 {
3     ...
4 }
```

**Q.5 :** Quel problème pose l'extrait de code suivant ? Comment pouvez-vous y remédier ?

```
1 void inc(uint64_t x, uint64_t i, uint64_t j)
2 {
3     x = x*i + j;
4 }
5 ...
6 uint64_t x = 0, i = 1, j = 3;
7 while (inc(x,i,j) < 65537)
8 {
9     ...
10 }
```

**Q.6 :** Quel problème pose l'extrait de code suivant ? Comment pouvez-vous y remédier ?

```
1 #define QUAT(X) X*X*X*X
2 ...
3 uint64_t a, b, c;
4 ...
5 c = QUAT(a-b); // c = (a-b)**4;
```

## Exercice 2 : Multiplication de petits polynômes entiers (au moins 12 points)

Dans cet exercice on considère des polynômes de  $\mathbb{Z}[X]$  dont les coefficients sont des entiers positifs appartenant (initialement) à l'intervalle  $\llbracket 0, 7 \rrbracket$ . On notera  $\mathcal{B}_d$  l'ensemble de ces polynômes de degré au plus  $d$ . L'objectif est d'implémenter leur produit de deux façons différentes. On rappelle que le produit de  $P, Q \in \mathbb{Z}[X]$  de degré  $d$  est donné par  $P \times Q = \sum_{i=0}^d \sum_{j=0}^d P_i Q_j X^{i+j}$ , où  $P = \sum_{i=0}^d P_i X^i$ ,  $Q = \sum_{i=0}^d Q_i X^i$ . Alternativement,  $R := P \times Q = \sum_{i=0}^{2d} R_i X^i$  avec  $R_i = \sum_{0 \leq j, k \leq d, j+k=i} P_j Q_k$  et on remarquera que le nombre de termes dans la somme calculant  $R_d$  est égal à  $d+1$  et est maximal pour tous les  $R_i$ ,  $i \in \llbracket 0, 2d \rrbracket$ .

**Q.1 :**

1. Montrez qu'un entier  $x \in \llbracket 0, 7 \rrbracket$  peut être représenté sur 3 bits.
2. Soit  $x, y \in \llbracket 0, 7 \rrbracket$ , quelle est la valeur maximale du produit  $xy$  ? Combien de bits sont nécessaires et suffisants pour toujours pouvoir stocker ce produit ?
3. Soit  $s = \sum_{i=1}^n x_i y_i$  avec  $x_i, y_i \in \llbracket 0, 7 \rrbracket$ , combien de bits sont nécessaires et suffisants pour toujours pouvoir stocker  $s$  ? On attend ici une expression exacte, la plus simplifiée possible, utilisant les fonctions usuelles (par exemple  $\log(\cdot)$ ). Il n'est cependant pas nécessaire de chercher à évaluer les logarithmes de constantes.

**Q.2 :**

1. Expliquez pourquoi il est possible de représenter  $P \in \mathcal{B}_d$  en C sous la forme d'un tableau `uint8_t P[]` de taille  $d+1$  ?
2. Même question, pour  $R := P \times Q$  avec  $P, Q \in \mathcal{B}_d$ , pour une taille de tableau `uint8_t R[]` à préciser, et pour une valeur maximale de  $d$  à préciser.

**Q.3 :**

1. Écrivez une fonction

```
void prod_tab(uint8_t R[], uint8_t P[], uint8_t Q[], unsigned d);
```

qui calcule le produit de  $P, Q \in \mathcal{B}_d$  représentés suivant le format de la question précédente, et qui écrit le résultat dans  $\mathbb{R}$  sous ce même format. On supposera que tous les arguments sont des tableaux qui ont déjà été alloués et que les contraintes déterminées dans la question précédente sont respectées.



On ne considère à présent plus que des polynômes de  $\mathcal{B}_3$ , qu'on propose de représenter de la façon suivante :  $P \in \mathcal{B}_3$  est représenté par l'entier  $p := \sum_{i=0}^3 P_i 2^{8i}$ .

**Q.4 :**

1. Écrivez une fonction

```
uint64_t tab2int(uint8_t P[4]);
```

qui prend en entrée  $P \in \mathcal{B}_3$  représenté comme dans la question *Q.2* et renvoie une représentation de  $P$  suivant le format décrit ci-dessus.

2. Écrivez une fonction

```
void int2tab(uint8_t P[7], uint64_t p);
```

qui effectue la conversion inverse pour un polynôme de degré 6 dont les coefficients sont au plus égaux à 255. On supposera que l'argument  $P$  a déjà été alloué.

**Q.5 :** Soit  $P, Q \in \mathcal{B}_3$  représentés comme ci-dessus sous la forme d'entiers  $p$  et  $q$  de type `uint64_t` :

1. Montrez que la représentation (sous ce même format) du produit  $R = P \times Q$  est donnée par le résultat de type `uint64_t` du produit des deux entiers  $p$  et  $q$ . *Faites bien attention à justifier les tailles des coefficients du résultat.*
2. Écrivez une fonction

```
uint64_t prod_int(uint64_t p, uint64_t q);
```

qui calcule et renvoie le produit sous ce format.

**Q.6 :** Comparez l'efficacité des deux approches étudiées ici pour calculer le produit de polynômes de  $\mathcal{B}_3$  quand :

1. Le format de représentation des polynômes est imposé et égal à celui de la question *Q.2*.
2. Le format de représentation des polynômes est libre.