

High Performance Coding (HPC) Semestre printemps 2022-2023 Contrôle continu 1 14-05-2022

-					
,	T.C	m	-	m	٠
	16			111	

Nom:

- Aucune documentation n'est permise, y compris la feuille de vos voisins
- La calculatrice n'est pas autorisée
- Aucune réclamation ne sera acceptée en cas d'utilisation du crayon
- Ne pas utiliser de couleur rouge

Question	Points	Score
1	6	
2	6	
3	12	
4	10	7
5	6	
6	6	
7	10	
Total:	56	

Note:

Question	1:	6 1	poin	ts
----------	----	-----	------	----

Pourquoi les *bit-hacks* sont souvent bénéfiques à la vitesse d'exécution par rapport à l'implémentation standard d'un algorithme? **Quand** peuvent-ils être appliqués?

Question 2: 6 points

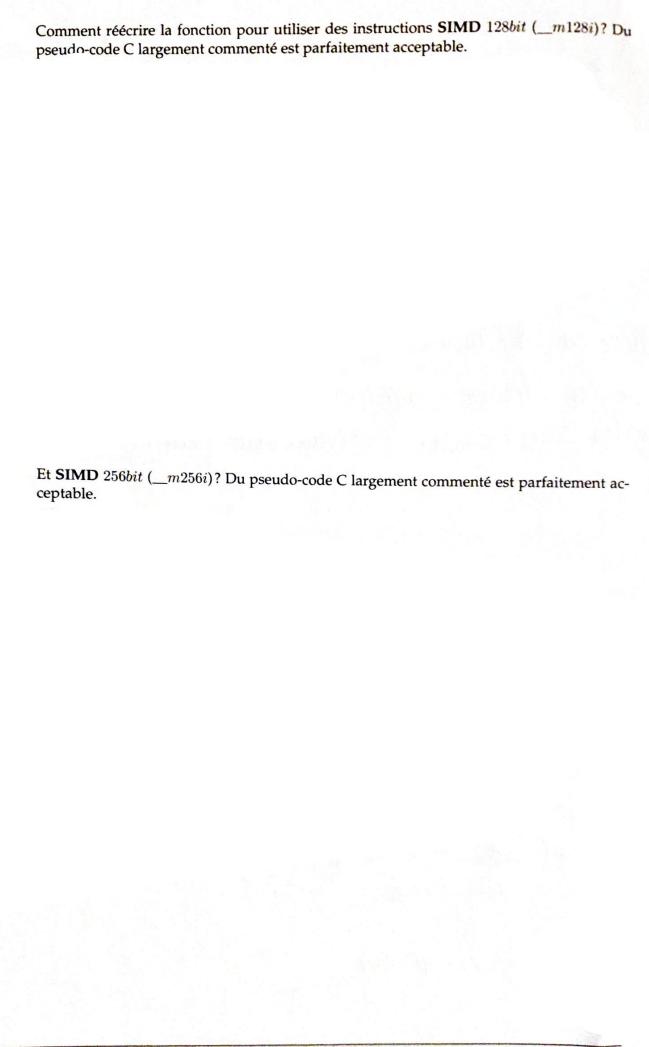
GCC et LLVM utilisent un flag -march=. Quand est-il judicieux de l'utiliser et quels compromis doivent être faits?

Question 3: 12 points

Pourquoi le compilateur ne peut pas optimiser cette fonction?

```
void add_tensor(int *src, int *dest, size_t size){
    size_t i;
    for(i=0; i<size; i++){
        dest[i] += src[i];
    }
    return;
}</pre>
```

Comment réécrire la fonction pour qu'elle soit optimisable?



Question 4: 10 points

Décrivez trois optimisations qu'un compilateur peut faire. Illustrez chaque exemple avec du code C.

Question 5: 6 points

Le code de la fonction initializeF peut être plus rapide que initialize. Pourquoi?

```
/* https://godbolt.org/
    gcc and LLVM do different things */
#include <string.h>

typedef struct{
    int a;
    char b;
    short c;
} MyStruct;

void initialize(MyStruct *s){
    s->a = 1;
    s->b = 2;
    s->c = 3;
}

void initializeF(MyStruct *s){
    memset(s,0,sizeof(*s));
    s->a = 1;
    s->b = 2;
    s->c = 3;
}
```

Question 6: 6 points

Définissez la notion de *profiler*. **Comparez** les avantages et desavantages des approaches *Time-based sampling, Event-based sampling* et *program Instrumentation* pour profiler une application.

Question 7: 10 points

Dessinez le *roofline model* (ne pas oublier d'indiquer l'unité sur les axes) d'un ordinateur hypothétique (A) sans *cache* et *single issue*. Ajoutez sur le même diagramme l'ordinateur (B), dérivé de A, mais avec support pour instructions *SIMD FP 4-way*.

Placer le code suivant sur le diagramme pour deux machines (DA, DB) et **expliquez** votre raisonnement. (vous pouvez supposer que le compilateur est capable de émettre le code de manière optimale)

```
#define SIZE 10000000

double Ad[SIZE], Bd[SIZE], Cd[SIZE];
const int bias = 123;

void D() {
    size_t i;
    for(i=0; i<SIZE; ++i) {
        Ad[i] = Bd[i] * Bd[i] + 6*Cd[i] + bias;
    }
}</pre>
```