

Programmation Parallèle: Examen Écrit

Exercice 1: OpenMP sections

Lister toutes les possibilités d'affichage (une par ligne) de l'exécution du programme suivant:

```
1 #pragma omp parallel num_threads(2)
2 {
3   #pragma omp sections
4   {
5     #pragma omp section
6     printf("A");
7     #pragma omp section
8     printf("B");
9   }
10  #pragma omp sections
11  {
12    #pragma omp section
13    printf("C");
14    #pragma omp section
15    printf("D");
16  }
17 }
```

Answer:

ABCD

ABDC

BACD

BADC

Exercice 2: OpenMP tasks

Quelle est la valeur (ou les valeurs possibles) de la variable `a` à la fin de l'exécution du code suivant? Astuce: Une barrière OpenMP implique toujours un `taskwait` (ce qui force d'attendre l'exécution de toutes les tâches soumises précédemment par le même thread/tâche).

```
1  int a = -3;
2  #pragma omp parallel num_threads(3)
3  {
4      #pragma omp single
5      #pragma omp task
6          a = 3;
7      #pragma omp task
8          a++;
9  }
```

Answer:

4, 5, 6

Exercice 3: Debugage AVX

Le programme ci-dessous a pour but d'initialiser un tableau d'entiers aléatoirement, ensuite le copier, de manière vectorisée, dans un autre tableau. Préciser **tous les bogues** dans le code (faire une liste et mentionner les un par un sans devoir les corriger), éventuellement en faisant référence au numéro de ligne correspondant à chaque bogue.

```
1  int main()
2  {
3      int N = 100;
4      int *A = malloc(N * sizeof(int));
5      int *B = malloc(N * sizeof(int));
6      // Init
7      for (int i = 0; i < N; i++) { A[i] = rand(); }
8      // Copie
9      for (int i = 0; i < N; i++) {
10         __m256i temp = _mm256_load_si256(A[i]);
11         _mm256_store_si256(B[i], temp);
12     }
13     free(A); free(B);
14     return 0;
15 }
```

Answer:

1. #9: i doit être incrémenté par 8
2. #10,#11: A[i] and B[i] ne sont pas passés par adresse
3. #12: Si N n'est pas divisible par 8, les derniers éléments ne seront pas traités
4. #10,#11: Load/store utilisé mais pas de garantie d'alignement des tableaux (utiliser loadu/storeu plutôt)

END OF EXAM