

2

Задача 1. Да се напише OpenMP програма во која ќе се провери бројот на максимални можни и креирани нитки. Дополнително, секоја нитка да испише порака на екран.

8

Задача 2. Да се напише OpenMP програма која ќе ги бара целобројните решенија на равенката:

$$x^4 * z + 3 * y^2 * z + 4 * z = 20$$

Работата да се распредели рамномерно помеѓу повеќе процеси. Да се испробаат сите можни комбинации на променливите x , y , z во опсегот $[-10, 10]$.

Најдените решенија да се прикажат на екран.

9

Задача 3. Да се напише програма во која од тастатура се вчитува низа од цели броеви. Програмата треба да ја најде најдолгата подниза за која важи правилото: секој број е еднаков на збирот од претходните два броја. Пребарувањето да се раздели рамномерно помеѓу повеќе нитки, при што на крајот програмата треба да го отпечати редниот број на елементот од каде започнува бараната подниза и нејзината должина.

10

Задача 4. Да се напише програма која во сортирана низа ќе направи бинарно пребарување на елемент. Програмата да отпечати дали елементот бил најден или не.

7

Задача 5. Да се напише MPI програма која ќе овозможи пронаоѓање на фигурата крал во слика од шаховска табла. Сликата од шаховската табла претставува матрица со димензи $n \times n$, при што на почеток димензијата на сликата не е позната (се внесува од тастатура). За споредбата на располагање е уште една фигура, т.е. помошна слика која е квадратна матрица со димензии $m \times m$, при што димензиите се познати преку вредноста n .

Главниот процес на крај треба да провери дали фигурата се појавува точно еднаш и во тој случај треба да ги испечати информациите за позицијата на која биле пронајдени решенијата. Ако бројот на решенија се разликува од еден, главниот процес треба да испечати порака за грешка.

Напомена: Содржината на матриците можете да ја генерирате произволно во опсег на вредности $[0-255]$. Генерирањето се врши во главниот процес. Малата матрица може и да ја позајмите од матрицата за шаховска табла за проверка на задачата.

Задача 6. Да се напише програма која ќе проверува дали даден елемент постои во бинарно пребарувачко дрво.

Напомена: Задачата е прикажана без `pragma omp single`, за да се илустрира значењето на наредбата.

Задача 7. Да се напише програма која ќе проверува дали даден елемент постои во бинарно пребарувачко дрво.