

## Lista Prove

1. Implementazione di un algoritmo parallelo (np core) per il calcolo dell'operazione  $c=a+b$  con  $a, b$  vettori di dimensione  $N$ , in ambiente openMP
2. Implementazione di un algoritmo parallelo (np core) per il calcolo dell'operazione  $c=\alpha*a+b$ , con  $a, b$  vettori di dimensione  $N$  e  $\alpha$  uno scalare, in ambiente openMP
3. Implementazione di un algoritmo parallelo (np core) per il calcolo degli elementi di un vettore  $c$ , i cui valori sono ottenuti moltiplicando l'identificativo del core per gli elementi di un vettore  $a$  di dimensione  $N$ , in ambiente openMP
4. Implementare un programma parallelo per l'ambiente multicore con np unità processanti impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve leggere una matrice di dimensione  $N \times N$ , quindi i core devo collaborare per ricopiare in parallelo gli elementi della diagonale principale in un vettore di lunghezza  $N$ . Infine, i core devono effettuare la somma degli elementi di tale vettore in parallelo.
5. Implementare un programma parallelo per l'ambiente multicore con np unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve generare

una matrice di dimensione  $N \times N$ , ogni core deve estrarre  $N/n_p$  righe e calcolare il prodotto puntuale tra i vettori corrispondenti alle righe estratte.

6. Implementare un programma parallelo per l'ambiente multicore con  $n_p$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve leggere una matrice  $A$  di dimensione  $N \times N$  ed uno scalare *alpha*. Quindi, i core devono collaborare per calcolare il prodotto tra lo scalare *alpha* e la matrice  $A$ , sfruttano una strategia di parallelizzazione che usi la distribuzione per colonne della matrice  $A$  come decomposizione del dominio.
7. Implementare un programma parallelo per l'ambiente multicore con  $n_p$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve generare una matrice  $A$  di dimensione  $N \times M$ . Quindi, i core devono collaborare per calcolare il minimo tra gli elementi delle colonne di propria competenza e conservare tale valore in un vettore  $c$  di dimensione  $M$ .
8. Implementare un programma parallelo per l'ambiente multicore con  $n_p$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve generare una matrice  $A$  di dimensione  $N \times M$ . Quindi, ogni core deve estrarre  $N/p$  colonne ed effettuare localmente la

somma degli elementi delle sottomatrici estratte, conservando il risultato in un vettore  $b$  di dimensione  $M$ .

9. Implementare un programma parallelo per l'ambiente multicore con  $np$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve generare due vettori casuali di dimensione  $N$ . Ogni core deve sommare ad ogni elemento dei due vettori il proprio identificativo e collaborare alla costruzione di una matrice, di dimensione  $N \times M$ , con  $M=2np$ , le cui colonne, a due a due, sono i due vettori ottenuti nella fase precedente.
10. Implementare un programma parallelo per l'ambiente multicore con  $np$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve leggere una matrice di dimensione  $N \times M$ , quindi i core devono collaborare per sommare in parallelo i vettori corrispondenti alle righe dispari della matrice.
11. Implementare un programma parallelo per l'ambiente multicore con  $np \times np$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve generare una matrice  $A$  di dimensione  $N \times N$ . Quindi, ogni core deve sommare tra loro gli elementi di un blocco

quadrato della matrice A di dimensione  $(N/np) \times (N/np)$ , conservando i valori in un vettore b.

12. Implementare un programma parallelo per l'ambiente multicore con np unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master generare una matrice di dimensione  $N \times M$ . Quindi, ogni core deve estrarre la riga della matrice con indice uguale al proprio identificativo e conservarla in un vettore locale; successivamente deve sommare gli elementi di tale vettore e conservare il risultato in un vettore c di dimensione np. Infine, i core devo collaborare per individuare in massimo tra gli elementi del vettore c.
13. Implementare un programma parallelo per l'ambiente multicore con np unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master generare una matrice A di dimensione  $M \times M$ . Quindi, i core devo collaborare valutare il minimo valore tra gli elementi di A e successivamente costruire la matrice B i cui elementi sono dati dalla somma di tale valore minimo per gli elementi di posto corrispondente della matrice di partenza.
14. Implementare un programma parallelo per l'ambiente multicore con  $np \times np$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve leggere

una matrice  $A$  di dimensione  $M \times M$ . Quindi, ogni core deve estrarre un blocco della matrice  $A$  di dimensione  $(M/np) \times (M/np)$ , conservando i valori in  $np \times np$  matrici. Infine, i core devono collaborare per calcolare la somma di tali matrici.

15. Implementare un programma parallelo per l'ambiente multicore con  $np$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve leggere un vettore  $a$ , di dimensione  $M$ , e uno scalare  $\alpha$ . Quindi i core devono collaborare per costruire una matrice  $A$ , di dimensione  $M \times M$ , la cui diagonale principale ha per elementi quelli del vettore  $a$  moltiplicati per lo scalare  $\alpha$ .
16. Implementare un programma parallelo per l'ambiente multicore con  $nq \times np$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve costruire una matrice  $A$ , di dimensione  $M \times N$ , i cui blocchi sono generati casualmente e in parallelo da ogni core.
17. Implementare un programma parallelo per l'ambiente multicore con  $np$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve leggere quattro valori scalari  $a_0, a_1, a_2, a_3$ , quindi i core devono collaborare per costruire un vettore di dimensione  $4np$ .

i cui elementi sono dati dalla somma di tali valori con il proprio identificativo.

18. Implementare un programma parallelo per l'ambiente multicore con  $np$  unità processanti che impieghi la libreria OpenMP. Il programma deve essere organizzato come segue: il core master deve leggere un vettore  $a$ , di dimensione  $N$  ed uno scalare  $\beta$ . Quindi i core devono collaborare per verificare se nel vettore esista almeno un elemento uguale al valore  $\beta$ .