

Esercizio 9

Scrivere un programma che, letto un intero 'a' inserito dall'utente, calcoli le potenze a^0, a^1, \dots, a^{10} e le memorizzi in un array. Prima di terminare, il programma stampa il contenuto dell'array. Utilizzare le funzioni per organizzare in maniera opportuna il codice.

Esercizio 10

Implementare le seguenti funzioni, che ricevono come argomento un array di numeri reali X e la sua dimensione N:

- `verifica_ordinamento()`: verifica se gli elementi di X sono in ordine crescente;
- `verifica_ordinamento_ric()`: verifica **ricorsivamente** se gli elementi di X sono in ordine crescente;
- `sequenza_ordinata()`: calcola la lunghezza della sequenza più lunga di elementi tali che $X_i \leq X_{i+1}$.

Esercizio 11

Scrivere un programma C che, dato un vettore di caratteri A di dimensione N (pari), scambi i primi N/2 elementi con i secondi N/2 e stampi il contenuto di A prima e dopo la modifica.

Esempio:

A	B	C	D	E	F
---	---	---	---	---	---

 →

D	E	F	A	B	C
---	---	---	---	---	---

Esercizio 12

Completare il seguente programma C implementando la funzione `ordina()`. Si vuole che gli interi memorizzati nelle variabili x, y e z siano stampati in ordine crescente, senza aggiungere altre istruzioni alla funzione `main()`.

```
void ordina ( ... ) {
    ...
}

int main()
{
    int x = ...;
    int y = ...;
    int z = ...;

    ordina(...);

    printf("x = %d, y = %d, z = %d\n", x, y, z);
    return 0;
}
```

Esercizio 13

Scrivere un programma che, data una matrice di interi A (NxN), verifichi se A è una matrice diagonale. Se non lo è, verificare se A è una matrice triangolare superiore o inferiore.