Esercizi 21-19-10

Riccardo Gennaro

October 2021

Esercizio 1

Scrivere un programma che legga una stringa - char input[DIM] - da *stdin* e la stampi **al contrario** due volte; la prima volta attraverso una funzione iterativa, la seconda volta con una funzione ricorsiva Le firme delle funzioni sono:

```
1    void stampa_iterativa(char*);
2    void stampa_ricorsiva(char*);
```

Esercizio 2

Scrivere la seguente funzione

```
1    float sum(float* p[], int n);
```

che ritorna la somma dei *floats* puntati dai primi n pointers nell'array **p**.

Esercizio 3

Scrivere nel file esercizio3.cc la dichiarazione e la definizione della funzione **ricorsiva** *somma_prodotto_incrociato* che, presi come parametri due array di numeri interi *primo* e *secondo*, della stessa dimensione, e un terzo parametro intero *dim*, pari alla dimensione dei due array, restituisca la somma dei prodotti di elementi dei due array, calcolati come segue. I prodotti vanno calcolati moltiplicando il primo elemento del primo array con l'ultimo elemento del secondo, poi il secondo elemento del primo array con il penultimo del secondo, il terzo con il terzultimo e così via.

Per esempio, dati due array

```
a:
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
e b:
    {1, 1, 2, 3, 5, 8, 13, 21, 34, 55}
```

così definiti il risultato del calcolo sarà **364**.

NOTA 1: La funzione **deve essere ricorsiva** ed al suo interno **non ci possono essere cicli o chiamate a funzioni contenenti cicli**. Può fare uso di eventuali funzioni ausiliarie purché a loro volta ricorsive.

NOTA 2: La funzione deve funzionare senza errore con ogni possibile array di dimensione uguale.

Esercizio 4

Scrivere un programma che, attraverso la funzione

```
1    void rotate(int a[], int n, int k);
```

ruota i primi n elementi dell'array **a**, k posizioni a destra (o k posizioni a sinistra se k è negativo).
Per esempio, la chiamata *rotate(a, 8, 3)* trasformerebbe {22, 33, 44, 55, 66, 77, 88, 99} in {77, 88, 99, 22, 33, 44, 55, 66}. La chiamata *rotate(a, 8, -5)* ha lo stesso effetto.

Esercizio 5

Scrivere un programma che, attraverso la funzione

```
1    int cmp(char* s1, char* s2);
```

compara n bytes a cominciare da **s2** con quelli corrispondenti di **s1**, dove n è il numero di bytes necessari affinché, sommati a **s2**, questo punti al carattere nullo '\0'. L'intero restituito deve essere pari a

- **0**, se tutti gli n bytes coincidono;
- **-1**, se il byte di **s1** è minore o uguale al byte di **s2** al primo mismatch;
- **1**, se il byte di **s1** è maggiore stretto al byte di **s2** al primo mismatch;