ESERCIZI - PUNTATORI

1. STRINGS:

   (a) scrivere un programma che conti il numero di caratteri di una stringa utilizzando l'aritmetica dei puntatori. Controllare il risultato utilizzando la funzione di libreria *strlen*. Soluzione in *string_pointer.c*.

   (b) scrivere un programma che concateni due stringhe utilizzando l'aritmetica dei puntatori. Soluzione in *cat_strings.c*.

2. SWAP:

   (a) scrivere una funzione che, invocata dalla *main()*, prenda come argomenti due variabili (per esempio *a=5* e *b=10*), e ne scambi i valori (ovvero dopo l'esecuzione della funzione swap si abbia *a=10* e *b=5*). Implementare il programma utilizzando solo variabili locali alle funzioni, ricorrendo all'utilizzo dei puntatori. Soluzione in *swap_pointer.c*.

3. ARRAYS:

   (a) scrivere un programma che allochi dinamicamente un array di N numeri float (es. $N = 10$), controlli che l'allocazione sia andata a buon fine e, tramite una funzione, lo inizializzi con dei numeri random distribuiti uniformemente nell'intervallo $[0, N]$. Il programma, attraverso l'uso di una funzione, deve calcolare e stampare a schermo la somma degli elementi dell'array. Prima della fine del programma la memoria allocata deve essere liberata. Soluzione in *array_pointers.c*.

   (b) scrivere un programma che allochi dinamicamente una matrice *NROW x NCOL* (con *NROW != NCOL*) di interi e ne verifichi la corretta allocazione. La matrice deve essere allocata dinamicamente come un blocco contiguo in memoria. Utilizzare una funzione per inizializzare la matrice tale che *MATRIX[i, j] = i+j*. Utilizzare una funzione per stampare a schermo i valori della matrice controllandone così la sua inizializzazione. NON si possono usare variabili globali. Soluzione in *matrix_pointer.c*.