ESERCIZI - FUNZIONI

1. **STACK OVERFLOW:**
   (a) scrivere un programma che allochi staticamente un array di interi aumentandone la dimensione allo scopo di determinare la dimensione della stack memory;
   (b) successivamente, utilizzare il comando *ulimit* per determinare la dimensione di sistema dello stack;
   (c) infine, dare da terminale il comando $ *ulimit -s unlimited* ed eseguire nuovamente il programma come al punto a).

2. **SWAP FUNCTION:**
   (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 *a=10* e *b=5*). Implementare la funzione senza ricorrere ai puntatori, che non sono ancora stati trattati a lezione.

3. **FATTORIALE CON FUNZIONE RICORSIVA:**
   (a) implementare il calcolo del fattoriale con una chiamata ricorsiva a funzione;
   (b) modificare il programma in modo tale che venga stampato a terminale il numero di invocazioni della funzione senza utilizzare per il conteggio una variabile globale.

4. **PREPROCESSORE:**
   (a) definire una macro per la determinazione del massimo tra due valori (suggerimento: *#define MAX(A, B) ....*);
   (b) verificare come il preprocessore sostituisce la macro nel codice sorgente (suggerimento: $ *gcc -E nome_sorgente ....*);
   (c) assegnare dei valori iniziali a due variabili intere, per esempio *i= j=2;* la macro *MAX(++i, j)* restituisce il valore atteso?
   (d) definire la macro *#define SQUARE(x) (x * x)* . Assegnare ad *i* un valore non nullo, invocare la macro come *SQUARE(i+1)* e controllare il risultato.

5. **ZERI DI UNA FUNZIONE (opzionale):**
   (a) riprendere l'esercizio del *root-finding* presentato in *es_2.pdf* e scrivere delle funzioni per il calcolo della f(x) = 2x^3 -4x + 1 e per l'algoritmo della ricerca delle radici.