Esercizi su funzioni e procedure ricorsive
Quando si ha a che fare con array in funzioni o procedure ricorsive, si possono usare due metodi: il primo utilizza l'aritmetica dei puntatori per aggiornare ogni volta la testa dell'array, il secondo invece si porta dietro un indice per tenere traccia della posizione raggiunta. Ad ogni modo e' sempre necessario portarsi dietro la dimensione (residua o totale, a seconda del caso.
Ricordate che oltre alla funzione o procedura indicata dall'esercizio, si richiede che scriviate anche un main che usi tale funzione e ne dimostri *ogni* funzionalita'.
Inoltre, se con l'esercizio viene fornita la firma della funzione/procedura da scrivere, si richiede che la soluzione rispetti questa firma in ogni dettaglio.
Ricordate infine che una procedura e' una funzione con tipo di ritorno void.

Esercizi
1) Scrivere una funzione ricorsiva potenza che riceva due interi, base e esponente e ritorni il valore della base elevata alla potenza esponente.
```
int potenza(int base, int esponente);
```

2) Scrivere una funzione ricorsiva fattoriale che riceva un intero e ritorni il fattoriale di quel valore.
```
int fattoriale(int val);
```

3) Scrivere una procedura ricorsiva che riceva un array a lo stampi a video.
Scrivere la procedura con i due metodi:
- con l'aritmetica dei puntatori
```
void stampa1 (int vet[], int dim);
```
- con un indice
```
void stampa2 (int vet[], int i, int dim);
```

4) Modificare le procedure ricorsive dell'esercizio precedente per stampare l'array in senso inverso.
```
void stampaInverso1 (int vet[], int dim);
void stampaInverso2 (int vet[], int i, int dim);
```

5) Scrivere una funzione ricorsiva che calcoli la somma di due numeri senza usare altra operazione matematica che l'incremento o il decremento di uno.
```
int somma(int a, int b);
```

6)Scrivere una funzione che calcoli ricorsivamente il numero di elementi pari di un array passato.

7) Scrivere una funzione che controlli che l'array di caratteri passato per argomento sia palindromo.

8)Scrivere una funzione ricorsiva che azzeri tutti gli elementi pari, raddoppi gli elementi dispari e ritorni il minimo elemento di un array passato.

9) Scrivere una funzione che chieda un valore N all'utente e quindi calcoli ricorsivamente l'N-esimo termine della successione di Fibonacci.
F(n) = F(n-1) + F(n-2)

10) Scrivere una funzione ricorsiva che riceva un array e due indici e calcoli la somma degli elementi dell'array compresi fra quegli indici.
int sumVet(int v[], int from, int to);

11) Scrivere una procedura ricorsiva che inverte la porzione di un array individuata dagli indici from e to.
void inverti(int v[], int from, int to);

12) La funzione di Ackermann e' una delle piu' semplici funzioni totalmente computabili a non essere ricorsiva primitiva.
http://it.wikipedia.org/wiki/Funzione_di_Ackermann
In pratica, la funzione cresce piu' velocemente di qualsiasi funzione ricorsiva primitiva (compreso qualsiasi esponenziale).
La funzione e' definita ricorsivamente per casi (sui naturali):
- A(m,n) = n+1 (se m=0)
- A(m,n) = A(m-1,1) (se m>0 e n=0)
- A(m,n) = A(m-1, A(m, n-1)) (se m>0 e n>0)

Scrivere una funzione che calcoli la funzione di Ackermann.
unsigned long Ackermann(unsigned long m, unsigned long n);
Quanto vale Ackermann(3,10)? Quanto vale Ackermann(4,1)?
Quanto tempo ci mette a calcolare questi valori?
Avvertenza: non andate oltre questi limiti (soprattutto su m) o l'esecuzione
potrebbe non terminare in tempo per il fine settimana...

13) Scrivere una procedura ricorsiva che riceva un array di caratteri e stampi la prima meta' su una riga e la seconda meta' su una riga diversa.
Se l'array ha un numero dispari di elementi, la lettera centrale dovra' essere stampata su una riga a parte, tra la prima meta' e la seconda meta'.

14) Riuscite a scrivere la funzione fibonacci ottimizzando (riducendo al minimo) il numero di chiamate ricorsive?
Suggerimento: Per verificare quante chiamate vengono eseguite, usate una variabile globale che viene incrementata ad ogni chiamata.
Suggerimento 2: potete sfruttare il fatto che per calcolate F(n-1) dovete sapere F(n-2)?
Suggerimento 3: una funzione in C non puo' ritornare piu' di un valore ma se riceve dei puntatori puo' modificare i valori delle variabili originali.

15) Scrivere una funzione che ricevuto un array di interi, calcoli e stampi ricorsivamente tutte le permutazioni dell'array.