



Esercitazione di laboratorio n. 5

(Caricamento nella Sezione Elaborati del Portale entro e non oltre 11/11/2014)

Competenze C: *uso della ricorsione per risolvere semplici problemi matematici, uso di algoritmi ricorsivi su dati diversi dagli interi, semplici manipolazioni di strutture linkate.*

Esercizio n. 1: Conversione Ricorsiva tra Basi

Sia dato un numero n in base 10 letto da tastiera. Sia data una base b letta da tastiera ($2 \leq b \leq 10$). Si converta n in base b mediante un algoritmo ricorsivo.

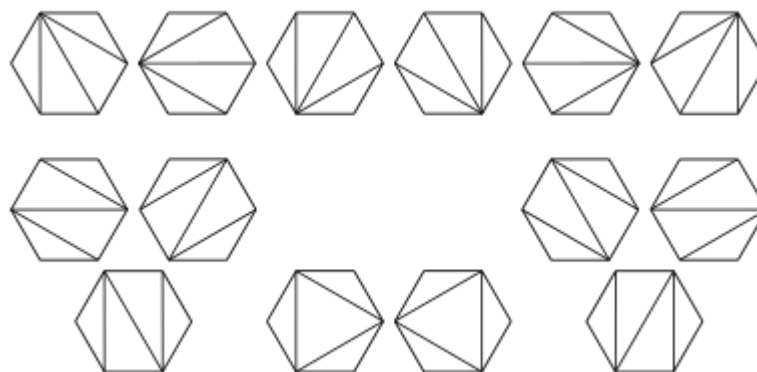
Esercizio n. 2: numeri di Catalan

Il matematico belga Eugène Charles Catalan ha formalizzato nel 1838 i concetti legati alla successione di numeri che prendono il suo nome. In realtà la loro definizione ricorsiva è dovuta al matematico ungherese Segner nel XVIII secolo:

$$C_0 = 1$$
$$C_n = \sum_{i=0}^{n-1} C_i \cdot C_{n-1-i}$$

I primi numeri di Catalan sono: 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862

I numeri di Catalan hanno svariate applicazioni nel calcolo combinatorio, ad esempio C_n è il numero di modi in cui un poligono convesso con $n+2$ lati può essere suddiviso in triangoli. Per $n=4$ il poligono è un esagono e i modi sono 14:



Si scriva un programma C che, letto un intero n , calcoli ricorsivamente il numero di Catalan C_n .

Esercizio n. 3: MergeSort su vettore di “item”

Si scriva un programma in grado di ordinare un vettore di elementi di tipo “item” mediante l'uso dell'algoritmo di MergeSort.

Si supponga che il tipo di dato base “item” memorizzato nel vettore possa essere (in alternativa):

- una stringa di lunghezza massima pari a 50 caratteri e priva di spazi
- un numero intero.



Si supponga che l'elenco di elementi da ordinare sia memorizzato in file di testo il cui nome sia richiesto all'utente. Indipendentemente dal tipo di contenuto, si assuma che i file siano sempre organizzati come segue:

- un intero N sulla prima riga a rappresentare il numero di elementi contenuti nel file
- N righe successive, ognuna delle quali contiene un singolo dato (un intero, una stringa, ...)

Per ognuno dei tipi proposti, si richiede di scrivere una apposita funzione di confronto.

Esercizio n. 4: Sala d'attesa

Si scriva un programma in grado di gestire un elenco di persone in attesa del proprio turno.

Tale elenco sia realizzato facendo uso di una struttura linkata, in cui ogni elemento abbia un puntatore a quello che lo segue nella fila.

Ogni persona sia identificata da un codice univoco di 10 caratteri alfanumerici privo di spazi (es: GVNRS1967).

Il programma deve offrire le seguenti funzionalità:

- aggiunta di una persona in fondo alla fila
- estrazione di una persona dall'inizio della fila
- stampa dell'elenco di persone in coda dal primo all'ultimo.

NB: l'implementazione delle funzioni di deallocazione è opzionale, ma consigliata.