

**Esame di ammissione
alla Scuola Superiore di Udine
Prova scritta di informatica – 14/9/12**

Risolvere i seguenti problemi:

1. Definiamo partizione di un numero naturale n una sequenza (debolmente) crescente di numeri naturali, diversi da zero, la cui somma risulta essere uguale ad n . Una sequenza si dice lentamente crescente se inizia con uno e la differenza di ogni elemento dal precedente è al più uno. Una sequenza si dice strettamente crescente se non esistono due elementi della sequenza identici. Definiamo $LC(n)$ il numero di partizioni lentamente crescenti di n , mentre definiamo $SC(n)$ il numero di partizioni strettamente crescenti di n . Scrivere un programma che dato un numero naturale n verifica se $LC(n) = SC(n)$, oppure dimostrare che per ogni n i due numeri sono uguali.
2. In una matrice di numeri interi, definiamo *percorso* una sequenza di elementi della matrice attigui tra di loro in senso orizzontale o verticale. Scrivere un programma che data una matrice quadrata A di dimensione n , trova il più lungo percorso in A contenente solo numeri distinti tra loro.
3. Definiamo un numero naturale n *primo a cifre* se n è primo e la somma delle cifre di n è a sua volta un numero primo. Per esempi 41 è un numero primo a cifre, in quanto $4+1 = 5$, mentre 17 non lo è, in quanto $1+7 = 8$. Scrivere un programma che dato un numero naturale n determina se questi è primo a cifre.
4. Un *grafo* è costituito da un insieme di nodi, che possiamo immaginare come punti di un piano, e da un insieme di archi, ogni arco collega tra loro due nodi, possiamo immaginare un arco come un segmento avente come estremi due nodi. In un grafo un *percorso* è costituito da una sequenza di nodi in cui ogni coppia di nodi consecutivi è collegati da un arco. Un grafo si dice *connesso* se ogni coppia di nodi è collegata attraverso una sequenza di archi, ossia esiste un percorso avente come primo e ultimo elemento i nodi della coppia.
Dimostrare, per induzione, che un grafo connesso con n archi contiene al più $n + 1$ nodi.

Indicazioni su come presentare le soluzioni.

La descrizione dei programmi va fatta spiegando a parole le idee base e quindi fornendo una descrizione più formale del programma. Questa descrizione può essere fatta nel formalismo che si ritiene più opportuno. È possibile utilizzare un linguaggio di programmazione standard (quali C, Pascal, Java, Scheme, ...) od utilizzare un linguaggio di progetto (come ad esempio un linguaggio Pascal-like).

Nel presentare i programmi, si possono tralasciare aspetti non centrali, quali l'acquisizione dei dati, la stampa del risultato, il controllo della consistenza dei dati in ingresso. Si raccomanda tuttavia di commentare i programmi proposti.

