

Esercizi di verifica

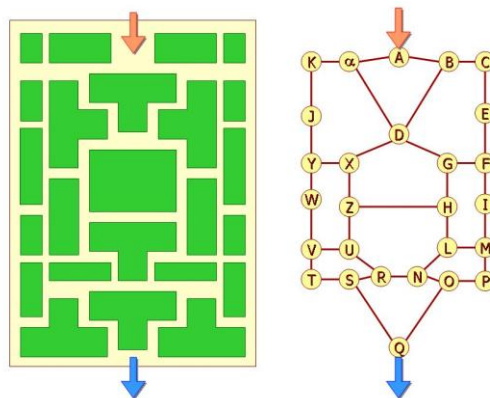
10 – Strutture dati dinamiche reticolari: grafi

P2_10_02_T

56. [liv.1] Scrivere *function C* per la costruzione di un *grafo non orientato* mediante *matrice di adiacenze*: in input per ogni nodo sono specificati quelli *adiacenti*. Scegliendo in input un nodo, scrivere una *function C* che restituisca il suo grado.
57. [liv.1] Scrivere *function C* per la costruzione di un *grafo orientato* mediante *matrice di adiacenze*: in input per ogni nodo sono specificati quelli *raggiungibili*. Scegliendo in input un nodo, scrivere una *function C* che restituisca il numero degli archi uscenti e quello degli archi entranti.
58. [liv.2] Scrivere *function C* per la costruzione di un *grafo non orientato* mediante liste di adiacenze: in input per ogni nodo sono specificati quelli *adiacenti*.

P2_10_03_T

59. [liv.2] Scrivere *function C* per la visita in *ordine anticipato* di un *albero qualsiasi*.
60. [liv.3] Scrivere *function C* per la visita di un *grafo* mediante l'algoritmo *Depth First Search* iterativo. Applicare l'algoritmo al grafo che descrive il seguente labirinto per determinare un cammino che parte dall'ingresso A e termina nell'uscita Q.



P2_10_04_T

61. [liv.3] Scrivere *function C* per la visita di un *grafo* mediante l'algoritmo *Breadth First Search*. Applicare l'algoritmo per stabilire se esiste un cammino che unisce due nodi qualsiasi di un grafo stabiliti in input.