

✓ `p.pacman(r, c)` ;

Operazione che assegna la posizione di Pacman nel livello `p` nella casella (r,c) . L'operatore di uscita deve rappresentare la posizione di Pacman nel livello con un carattere '**P**'. Pacman deve essere posizionato in una casella vuota. Se la casella non era vuota, o Pacman era già stato posizionato, o gli indici della casella non sono validi, la funzione non ha effetto.

✓ `p.muovi(d, n)` ;

Operazione che fa camminare Pacman di `n` caselle nella direzione specificata da `d`, che può essere '**d**' (verso destra), '**s**' (verso sinistra), '**b**' (verso il basso), '**a**' (verso l'alto). Se durante la camminata Pacman trova un muro davanti a sé, esso arresta in anticipo la sua camminata. Il mondo di Pacman segue le speciali leggi geometriche dell'*effetto Pacman*: se Pacman esce dal bordo destro del livello rientra dal bordo sinistro e viceversa, se esce dal bordo inferiore del livello rientra dal bordo superiore e viceversa. Se gli input non sono validi, o se non c'è nessun Pacman nel livello, la funzione non ha effetto. Deve essere possibile concatenare le chiamate a `muovi`.

--- SECONDA PARTE ---

✓ `p.spazio(d)` ;

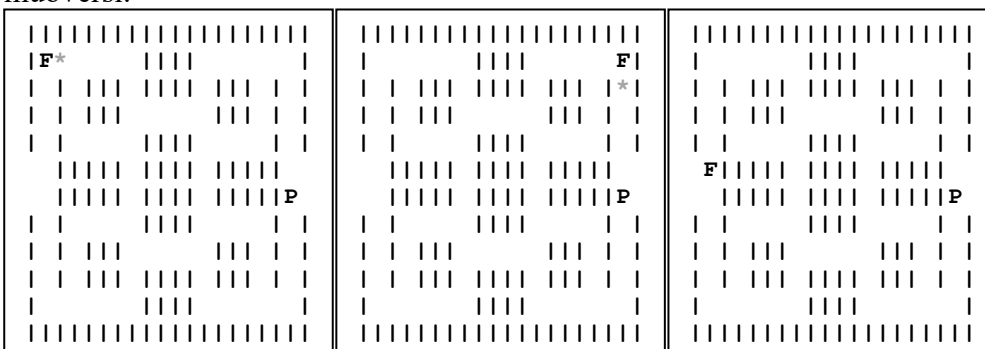
Operazione che restituisce il numero di spazi vuoti visti da Pacman nella direzione specificata da `d` (vedi analogo parametro della funzione `p.muovi(d, n)`). La funzione non è soggetta all'effetto Pacman. Per esempio, se Pacman vede verso destra e non c'è nessun muro (o fantasma), gli spazi da contare terminano sul bordo destro del livello. Se l'input non è valido, o se non c'è nessun Pacman nel livello, la funzione restituisce il valore speciale -1.

✓ `p.fantasma(r, c)` ;

Operazione che assegna la posizione del fantasma nel livello `p` nella casella (r,c) . L'operatore di uscita deve rappresentare la posizione del fantasma nel livello con un carattere '**F**'. Il fantasma deve essere posizionato in una casella vuota. Se la casella non è vuota, o il fantasma è già stato posizionato, o gli indici della casella non sono validi, la funzione non ha effetto.

✓ `p.muovi(d, n)` ; (MODIFICA)

Modificare la funzione `muovi` per tenere conto dell'eventuale fantasma. Pacman non può muovere nella stessa casella del fantasma, in tal caso esso arresta in anticipo la sua camminata. Ogni volta che Pacman muove di una casella, anche il fantasma muove automaticamente di una casella per cercare di prendere Pacman. Il fantasma muove verso Pacman nella direzione di *distanza maggiore*, ovvero nella direzione che diminuisce di uno la maggiore tra la distanza verticale e quella orizzontale tra Pacman e il fantasma stesso. Il fantasma cerca di muoversi in quella direzione, a meno che in tale direzione non ci sia un muro, in quel caso il Fantasma rinuncia a muoversi. Il fantasma non usufruisce dell'effetto Pacman. Se nel suo movimento, il fantasma entra nella stessa casella di Pacman, Pacman muore e viene rimosso dal livello. Se Pacman non è presente nel livello, il fantasma non si muove. Seguono tre esempi di movimento del fantasma, in cui il carattere '*****' grigio rappresenta la casella in cui muove il fantasma. Nel terzo esempio il fantasma rinuncia a muoversi.



✓ `p.fermo()`;

Operazione che fa aspettare Pacman per un po' di tempo. Questo ha l'effetto di far muovere l'eventuale fantasma di una casella secondo le regole specificate dalla funzione `p.muovi(d,n)`. Se il fantasma non è presente nel livello, la funzione non ha effetto. Deve essere possibile concatenare le chiamate a `fermo`.

Mediante il linguaggio C++, realizzare il tipo di dato astratto definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL come il tipo `string`, il tipo `vector`, il tipo `list`, ecc.