

Indice

loadVerticalAxis	1
loadHorizontalAxis.....	22
createPlayer.....	43
createPlayground	46
createHeatMap.....	47
pullColumn	48
pullRow	49
getIntegerColumn.....	51
getCharColumn.....	52
concatString.....	53
buildVerticalCoord.....	54
buildHorizontalCoord	56
concatCoordinates.....	58
getFirstCell.....	61
getLastCell	62
buildPlayerNavy.....	63
buildShipCoordinate.....	67
buildNextCoord	68
createShip.....	72
toUpperCase.....	77
getLength.....	77
numberToString.....	78
stringToNumber	79
isImpossible	80
checkBoundaries	86
checkCollisions.....	88

loadVerticalAxis**INPUT:**

Nome	Descrizione	Tipo	Vincoli
player	Giocatore a cui inserire una nave	Player	\
startColumn	Colonna di partenza della nave	intero	startColumn >= TABLE_MIN AND startColumn <= TABLE_MAX

startRow	Riga di partenza della nave	intero	startRow >= TABLE_MIN AND startRow <= TABLE_MAX
label	Etichetta della nave	carattere	label >= 'a' AND label <= 'o'
shipSize	Dimensioni della nave	intero	shipSize >= MIN_SHIP_SIZE AND shipSize <= MAX_SHIP_SIZE

OUTPUT:

Nome	Descrizione	Tipo	Vincoli
player	Giocatore aggiornato con all'interno del suo playground una nave	Player	\

DESCRIZIONE DELLA FUNZIONE: La funzione carica una nave nel playground del giocatore utilizzando tutte le informazioni presenti in input

[illegible]

	<pre> ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship= { Label='a' Size=5 Coords = "A-4_A-8" Direction='V' lifePoints=5 } } startColumn = 1 startRow = 4 label = 'a' shipSize = 5 </pre>	<pre> ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } } </pre>
2	<pre> Player = { Id=1 playground= ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ </pre>	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ </pre>

<pre> ~ heatmap= ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship= { Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship= { Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } } startColumn = 16 startRow = 1 </pre>	<pre> ~ heatmap= ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship= { Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship= { Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } } </pre>
--	--

	<pre>label = 'b' shipSize = 4</pre>	
3	<pre>Player = { Id=1 playground= ~~~~~ b ~~~~~ b ~~~~~ b a ~~~~ b a ~~~~~ a ~~~~~ a ~~~~~ a ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ heatmap= ??????? ??????? ??????? ??????? ??????? ??????? ??????? ??????? ??????? ??????? ??????? ??????? ??????? ??????? ??????? ????? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship= { Label='a'</pre>	CASO IMPOSSIBILE DA ESEGUIRE: La label in input corrisponde sempre con la label dell’ultima ship nel campo Ships di player (input controllato nella funzione chiamante)

	Size=5 Coords=""A-4_A-8" Direction='V' lifePoints=5	
	} Ship={ Label='b' Size=4 Coords=""P-1_P-4" Direction='V' lifePoints=4	
	}	
	Ship={ Label='c' Size=4 Coords=""M-10_M-13" Direction='V' lifePoints=4	
	}	
	}	
	startColumn = 13 startRow = 10 label = 'd'	
	shipSize = 4	
4	Player = { Id=1 playground= ~~~~~ b ~~~~~ b ~~~~~ b a ~~~~ b a ~~~~ a ~~~~ a ~~~~ a ~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ heatmap=	CASO IMPOSSIBILE DA ESEGUIRE: La nave da caricare è la terza e dunque la sua dimensione deve essere 4 (input controllato nella funzione chiamante)

	<pre>???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship= { Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship= { Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship= { Label='c' Size = 3 Coords = "M-10_M-13" Direction = 'V' lifePoints = 4 } } startColumn = 13 startRow = 10</pre>	
--	--	--

	label = 'c' shipSize = 4	
5	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ heatmap= ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship= { Label='a' </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE: La colonna di partenza non potrà mai avere un valore superiore di 16</p> <p>(input controllato nella funzione chiamante)</p>

	<pre>???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? ???????????????? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship= { Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship= { Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship= { Label='c' Size=4 Coords="M-10_M-13" Direction='V' lifePoints=4 } } startColumn = 13 startRow = 17</pre>	
--	--	--

	label = 'c' shipSize =4	
7	Player = { Id=2 playground= ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ heatmap= ?????? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a'	CASO IMPOSSIBILE DA ESEGUIRE: in base all'ordine impostato dalla tabella ASCII, la label della nave non potrà mai essere un carattere successivo a 'o' (input controllato nella funzione chiamante)

	<pre> Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } } startColumn = 1 startRow = 4 label = 'z' shipSize = 5 </pre>	
8	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ heatmap= ? </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE:</p> <p>La size della nave non potrà mai superare il valore di 5</p> <p>(input controllato dalla funzione chiamante)</p>

	<pre> ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship= { Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship= { Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship= { Label='c' Size=4 Coords="M-10_M-13" Direction='V' lifePoints=4 } } startColumn = 13 startRow = 10 label = 'c' shipSize =7 </pre>	
9	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE: La colonna di partenza non potrà mai avere un valore inferiore a 1</p> <p>(input controllato nella funzione chiamante)</p>

<pre> ~ heatmap= ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship= { Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' </pre>	
---	--

	<pre> Size=4 Coords="M-10_M-13" Direction='V' lifePoints=4 } } startColumn = -4 startRow = 10 label = 'c' shipSize =4 </pre>	
10	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ heatmap= ? </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE: La riga di partenza non potrà mai avere un valore inferiore a 1</p> <p>(input controllato nella funzione chiamante)</p>

	<pre> ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_M-13" Direction='V' lifePoints=4 } } startColumn = 13 startRow = 0 label = 'c' shipSize = 4 </pre>	
11	<pre> Player = { Id=2 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE: In base all'ordine impostato dalla tabella ASCII, la label della nave non potrà mai essere un carattere precedente a 'a'</p> <p>(input controllato nella funzione chiamate)</p>

	<pre>~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ heatmap= ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? ????????? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } } startColumn = 1 startRow = 4 label = '9' shipSize = 5</pre>	
--	---	--

12	<pre>Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ heatmap= ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8"</pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE:</p> <p>La size della nave non potrà mai essere inferiore a 1</p> <p>(input controllato dalla funzione chiamante)</p>
----	--	---

	<pre>Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_M-13" Direction='V' lifePoints=4 }</pre> <p>} startColumn = 13 startRow = 10 label = 'c' shipSize = 0</p>	
13	<pre>Player = { Id=1 playground= ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ heatmap= ? ? ? ? ? ? ? ? ? ? </pre>	CASO IMPOSSIBILE DA ESEGUIRE: Label non può essere vuoto (input controllato dalla funzione chiamante)

	<pre>???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_M-13" Direction='V' lifePoints=4 } } startColumn = 13 startRow = 10 label = '' shipSize =0</pre>	
--	--	--

14	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ heatmap= ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? longshots = 3 radar = 3 airStrike = 1 availableShips = 15 Ships = { Ship={ Label='a' Size = 5 Coords = "A-4_A-8" </pre>	<p>CASO IMPOSSIBILE</p> <p>Il campo sizeShip non corrisponde con il campo Size dell'ultima nave inserita</p> <p>(input controllato da funzione chiamante)</p>
----	--	---

	<pre> Direction = 'V' lifePoints = 5 } } } startColumn = 1 startRow = 4 label = 'a' shipSize = 4 </pre>	
--	---	--

loadHorizontalAxis**INPUT:**

Nome	Descrizione	Tipo	Vincoli
player	Giocatore a cui inserire una nave	Player	\
startColumn	Colonna di partenza della nave	intero	startColumn >= TABLE_MIN AND startColumn <= TABLE_MAX
startRow	Riga di partenza della nave	intero	startRow >= TABLE_MIN AND startRow <= TABLE_MAX

[illegible]

3

	<pre>playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ heatmap= ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } }</pre>	(input controllato nella funzione chiamante)
--	---	--

	<pre> Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_P-10" Direction='O' lifePoints=4 } } startColumn = 13 startRow = 10 label = 'd' shipSize =4 </pre>	
4	<pre> Player = { Id=1 playground= ~~~~~ b ~~~~~ b ~~~~~ b a ~~~~ b a ~~~~~ a ~~~~~ a ~~~~~ a ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ heatmap= ???????????? ???????????? ???????????? ???????????? ???????????? </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE:</p> <p>La nave da caricare è la terza e dunque la sua dimensione deve essere 4</p> <p>(input controllato nella funzione chiamante)</p>

	<pre> ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=3 Coords="M-10_P-10" Direction='O' lifePoints=4 } } startColumn = 13 startRow = 10 label = 'c' shipSize =4 </pre>	
5	<pre> Player = { Id=1 </pre>	CASO IMPOSSIBILE DA ESEGUIRE: La colonna di partenza non potrà mai avere un valore superiore di 16

	<pre>playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ heatmap= ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } }</pre>	(input controllato nella funzione chiamante)
--	---	--

	<pre> Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_P-10" Direction='O' lifePoints=4 } } startColumn = 18 startRow = 10 label = 'c' shipSize =4 </pre>	
6	<pre> Player = { Id=1 playground= ~~~~~b ~~~~~b ~~~~~b a~~~~~b a~~~~~ a~~~~~ a~~~~~ a~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ heatmap= ???????????? ???????????? ???????????? ???????????? ???????????? </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE: La riga di partenza non potrà mai avere un valore superiore di 16</p> <p>(input controllato nella funzione chiamante)</p>

	<pre> ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_P-10" Direction='O' lifePoints=4 } } startColumn = 13 startRow = 20 label = 'c' shipSize =4 </pre>	
7	<pre> Player = { Id=2 </pre>	CASO IMPOSSIBILE DA ESEGUIRE: In base all'ordine impostato dalla tabella ASCII, la

	<pre>playground= ~ heatmap= ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_E-4" Direction='O' lifePoints=5 } }</pre>	<p>label della nave non potrà mai essere un carattere successivo a 'o'</p> <p>(input controllato nella funzione chiamate)</p>
--	---	---

	<pre> } } startColumn = 1 startRow = 4 label = 'z' shipSize = 5 </pre>	
8	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ heatmap= ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? longshots=3 radar=3 airStrike=1 </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE:</p> <p>La size della nave non potrà mai superare il valore di 5</p> <p>(input controllato dalla funzione chiamante)</p>

	<pre> availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_P-10" Direction='O' lifePoints=4 } } startColumn = 13 startRow = 10 label = 'c' shipSize = 7 </pre>	
9	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE: La colonna di partenza non potrà mai avere un valore inferiore a 1</p> <p>(input controllato nella funzione chiamante)</p>

<pre> ~ heatmap= ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_P-10" Direction='O' lifePoints=4 } } </pre>	
---	--

	<pre> } } startColumn = 0 startRow = 10 label = 'c' shipSize =4 </pre>	
10	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ heatmap= ? longshots=3 radar=3 airStrike=1 </pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE: La riga di partenza non potrà mai avere un valore inferiore a 1</p> <p>(input controllato nella funzione chiamante)</p>

	<pre>availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords=""A-4_A-8"" Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords=""P-1_P-4"" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords=""M-10_P-10"" Direction='O' lifePoints=4 } } startColumn = 13 startRow = -2 label = 'c' shipSize =4</pre>	
11	<pre>Player = { Id=2 playground= ~</pre>	CASO IMPOSSIBILE DA ESEGUIRE: In base all’ordine imposto dalla tabella ASCII, la label della nave non potrà mai essere un carattere precedente a ‘a’ (input controllato nella funzione chiamate)

<pre> ~ heatmap= ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_E-4" Direction='O' lifePoints=5 } } startColumn = 1 startRow = 4 label = '9' shipSize = 5 </pre>	
--	--

12	<p>Player =</p> <pre>{ Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ heatmap= ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8"</pre>	<p>CASO IMPOSSIBILE DA ESEGUIRE:</p> <p>La size della nave non potrà mai essere inferiore a 1</p> <p>(input controllato dalla funzione chiamante)</p>
----	---	---

	<pre> Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_P-10" Direction='O' lifePoints=4 } } startColumn = 13 startRow = 10 label = 'c' shipSize =0 </pre>	
13	<pre> Player = { Id=1 playground= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ b a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ heatmap= ? </pre>	CASO IMPOSSIBILE DA ESEGUIRE: Label non può essere vuoto (input controllato dalla funzione chiamante)

	<pre>???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? ???????????? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_A-8" Direction='V' lifePoints=5 } Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } Ship={ Label='c' Size=4 Coords="M-10_P-10" Direction='O' lifePoints=4 } } startColumn = 13 startRow = 10 label = "" shipSize =4</pre>	
--	--	--

14	<pre> Player = { Id=1 playground= ~ heatmap= ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? longshots=3 radar=3 airStrike=1 availableShips=15 Ships = { Ship={ Label='a' Size=5 Coords="A-4_E-4" </pre>	<p>CASO IMPOSSIBILE</p> <p>Il campo sizeShip non corrisponde con il campo Size dell’ultima nave inserita</p> <p>(input controllato da funzione chiamante)</p>
----	--	---

	<pre> Direction='O' lifePoints=5 } } } startColumn = 1 startRow = 4 label = 'a' shipSize = 3 </pre>	
--	---	--

createPlayer**INPUT**

Nome	Descrizione	Tipo	Vincoli
Player	Giocatore da inizializzare	player	/
id	Identificatore del giocatore	Numero intero	1 = Player 1 2 = Player 2
MAX_LONG_SHOT	Numero massimo di longShot a disposizione di un player	Numero intero	3
MAX_RADAR	Numero massimo di radar a disposizione di un player	Numero intero	3
MAX_AIR_STRIKE	Numero massimo di airStrike a disposizione di un player	Numero intero	1
MAX_SHIP_AMOUNT	Numero massimo di navi in un playground	Numero intero	15

OUTPUT

Nome	Descrizione	Tipo	Vincoli
------	-------------	------	---------

[illegible]

		availableShips=15 Ships = { } }
3	Player = { } Id= 4 MAX_LONG_SHOT= 3 MAX_RADAR = 3 MAX_AIR_STRIKE = 1 MAX_SHIP_AMOUNT = 15	CASO IMPOSSIBILE Il valore dell'id non può essere un valore superiore a 2 (input controllato nella funzione chiamante)
4	Player = { } Id= 0 MAX_LONG_SHOT= 3 MAX_RADAR = 3 MAX_AIR_STRIKE = 1 MAX_SHIP_AMOUNT = 15	CASO IMPOSSIBILE Il valore dell'id non può essere un valore inferiore a 1 (input controllato nella funzione chiamante)

createPlayground**INPUT**

Nome	Descrizione	Tipo	Vincoli
TABLE_MAX	Riga/Colonna massima	Numero intero	16
WATER	Carattere che indica la presenza di acqua nel playground e nell'heatMap	Carattere	'~'

OUTPUT

Nome	Descrizione	Tipo	Vincoli
playGround	Playground del giocatore	Array di caratteri a due dimensioni	Dimensione 16x16 Caratteri consentiti: Da 'a' a 'o' per le navi '~' per il mare 'x' nave colpita '*' nave affondata

DESCRIZIONE DELLA FUNZIONE: La funzione inizializza il playground di un giocatore

[illegible]

Nome	Descrizione	Tipo	Vincoli
TABLE_MAX	Riga/Colonna massima	Numero intero	16
UNKNOWN	Carattere che indica le celle di cui non si hanno informazioni nell'heatMap	Carattere	'?'

Nome	Descrizione	Tipo	Vincoli
heatMap	HeatMap del giocatore	Array di caratteri a due dimensioni	Dimensione 16x16 '~' per il mare '?' per ignoto '!' per nave colpita '*' nave colpita e affondata '#' nave scansionata

DESCRIZIONE DELLA FUNZIONE: La funzione inizializza la heatMap di un giocatore

Gruppo 12 – Curri Mattia, Festa Donato, Fontana Emanuele, Labianca Stefano

[illegible]

INPUT

Nome	Descrizione	Tipo	Vincoli
Ship	Nave da cui estrarre il valore della colonna	Ship	/

Nome	Descrizione	Tipo	Vincoli
integerColumn	Valore numerico della colonna	Numero intero	integerColumn >= TABLE_MIN AND integerColumn <= TABLE_MAX

TEST

Numero di test	INPUT	OUTPUT
1	Ship={ Label = 'b' Size = 4 Coords = "P-1_P-4"	integerColumn = 16

Casi di test

Gruppo 12 – Curri Mattia, Festa Donato, Fontana Emanuele, Labianca Stefano

	<pre> Direction = 'V' lifePoints = 4 </pre>	
2	<pre> Ship={ Label = 'o' Size = 1 Coords = "A-1_A-1" Direction = 'O' lifePoints = 1 } </pre>	integerColumn = 1
3	<pre> Ship={ Label='o' Size=1 Coords="Z-1_Z-1" Direction='O' lifePoints=1 } </pre>	<p>CASO IMPOSSIBILE</p> <p>La colonna non può avere un valore superiore a 'P'</p> <p>(input controllato da funzione chiamante)</p>
4	<pre> Ship={ Label='o' Size=1 Coords="1-1_1-1" Direction='O' lifePoints=1 } </pre>	<p>CASO IMPOSSIBILE</p> <p>La colonna non può avere un valore inferiore a 'A'</p> <p>(input controllato da funzione chiamante)</p>
5	<pre> Ship={ Label='o' Size=1 Coords="C-1_C-1" Direction='O' lifePoints=1 } </pre>	integerColumn = 3
6	<pre> Ship={ Label='o' Size=1 Coords="" Direction='O' lifePoints=1 } </pre>	<p>CASO IMPOSSIBILE:</p> <p>Coords non può essere vuoto</p> <p>(input controllato da funzione chiamante)</p>

pullRow

INPUT

Nome	Descrizione	Tipo	Vincoli
Ship	Nave da cui estrarre il valore della riga	Ship	/

Casi di test

Gruppo 12 – Curri Mattia, Festa Donato, Fontana Emanuele, Labianca Stefano

COORD_SEPARATOR	Separatore di coordinate	Carattere	'_'
START_ROW_COORD	Inizio della riga nelle coordinate	Numero intero	3

OUTPUT

Nome	Descrizione	Tipo	Vincoli
integerRow	Valore numerico della riga	Numero intero	integerRow >= TABLE_MIN AND integerRow <= TABLE_MAX

DESCRIZIONE DELLA FUNZIONE: Restituisce, sotto forma di intero, la riga della prima coordinata della nave

Numero di test	INPUT	OUTPUT
1	Ship={ Label='b' Size=4 Coords="P-1_P-4" Direction='V' lifePoints=4 } COORD_SEPARATOR = '_' START_ROW_COORD=3	integerRow = 1
2	Ship={ Label='o' Size=1 Coords="A-4_A-4" Direction='O' lifePoints=1 } COORD_SEPARATOR = '_' START_ROW_COORD=3	integerRow = 4
3	Ship={ Label='o' Size=1 Coords="A-16_A-16" Direction='O' lifePoints=1 } COORD_SEPARATOR = '_' START_ROW_COORD=3	integerRow = 16
4	Ship={ Label='o' Size=1	CASO IMPOSSIBILE:

	Coords="A-0_A-0" Direction='O' lifePoints=1 } COORD_SEPARATOR = ' _ ' START_ROW_COORD=3	Il valore del terzo elemento di coords è sempre maggiore o uguale di 1 (input controllato da funzione chiamante)
5	Ship={ Label='o' Size=1 Coords="A-17_A-17" Direction='O' lifePoints=1 } COORD_SEPARATOR = ' _ ' START_ROW_COORD=3	CASO IMPOSSIBILE: Il valore del terzo elemento di coords è sempre minore o uguale di 16 (input controllato da funzione chiamante)
6	Ship={ Label='o' Size=1 Coords="" Direction='O' lifePoints=1 } COORD_SEPARATOR = ' _ ' START_ROW_COORD=3	CASO IMPOSSIBILE: Coords non può essere vuoto (input controllato da funzione chiamante)

getIntegerColumn**INPUT:**

Nome	Descrizione	Tipo	Vincoli
column	Carattere da convertire	Carattere	Contiene un valore tra 'A' e 'P'
START_UPPERCASE_ASCII	Prima lettera dell'alfabeto maiuscola in ASCII	Numero intero	65

OUTPUT

Nome	Descrizione	Tipo	Vincoli
numericColumn	Valore di column tradotto in numero	Intero	Column >= TABLE_MIN AND column <= TABLE_MAX

DESCRIZIONE DELLA FUNZIONE: Restituisce column sotto forma di intero

Numero di test	INPUT	OUTPUT
1	Column = 'A' START_UPPERCASE_ASCII= 65	numericColumn= 1
2	Column = 'P' START_UPPERCASE_ASCII= 65	numericColumn= 16
3	Column = 'Z' START_UPPERCASE_ASCII= 65	CASO IMPOSSIBILE: La colonna non potrà mai avere un valore maggiore di 'P' (input controllato da funzione chiamante)
4	Column = '7' START_UPPERCASE_ASCII= 65	CASO IMPOSSIBILE: La colonna non potrà mai avere un valore minore di 'A' (input controllato da funzione chiamante)
5	Column = 'C' START_UPPERCASE_ASCII= 65	numericColumn= 3
6	Column = '' START_UPPERCASE_ASCII= 65	CASO IMPOSSIBILE: Column non può essere vuoto (input controllato da funzione chiamante)

getCharColumn**INPUT:**

Nome	Descrizione	Tipo	Vincoli
column	Carattere da convertire	Intero	Column >= TABLE_MIN AND column <= TABLE_MAX
START_UPPERCASE_ASCII	Prima lettera dell'alfabeto maiuscola in ASCII	Numero intero	65

OUTPUT

Nome	Descrizione	Tipo	Vincoli
------	-------------	------	---------

Casi di test

Gruppo 12 – Curri Mattia, Festa Donato, Fontana Emanuele, Labianca Stefano

charColumn	Posizione della colonna convertita nel suo carattere associato	Carattere	Compreso tra 'A' e 'P'
------------	--	-----------	------------------------

DESCRIZIONE DELLA FUNZIONE: Converte column nel rispettivo carattere

Numero di test	INPUT	OUTPUT
1	START_UPPERCASE_ASCII= 65 Column = 1	charColumn='A'
2	START_UPPERCASE_ASCII= 65 Column = 16	charColumn='P'
3	START_UPPERCASE_ASCII= 65 Column = 4	charColumn='D'
4	START_UPPERCASE_ASCII= 65 Column = 0	CASO IMPOSSIBILE: column non può avere un valore minore di 1 (input controllato da funzione chiamante)
5	START_UPPERCASE_ASCII= 65 Column = 17	CASO IMPOSSIBILE: column non può avere un valore maggiore di 16 (input controllato da funzione chiamante)

concatString

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Str1	Prima stringa	Stringa	/
Str2	Seconda stringa da concatenare a Str1	Stringa	/

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Str1	Prima stringa con seconda concatenata	Stringa	/

DESCRIZIONE DELLA FUNZIONE: Concatena le stringhe Str1 e Str2

Numero di test	INPUT	OUTPUT
1	Str1 = "abc" Str2= "def"	Str1="abcdef"

2	Str1 = "" Str2= "def"	Str1="def"
3	Str1 = "abc" Str2= ""	Str1="abc"
4	Str1 = "" Str2= ""	Str1=""

buildVerticalCoord**INPUT**

Nome	Descrizione	Tipo	Vincoli
startingCoord	Coordinata di partenza	Stringa	Formato: riga-colonna
shipSize	Dimensione della nave	Numero intero	shipSize >= MIN_SHIP_SIZE AND shipSize <= MAX_SHIP_SIZE
START_ROW_COORD	Inizio della riga nelle coordinate	Numero intero	3
SEPARATOR	Separatore tra riga e colonna	Carattere	'-'

OUTPUT

Nome	Descrizione	Tipo	Vincoli
nextCoord	Coordinata finale	Stringa	Formato: riga-colonna

Descrizione della funzione: La funzione crea l'ultima coordinata della nave partendo dai dati che riceve in input

Numero di test	INPUT	OUTPUT
1	START_ROW_COORD = 3 SEPARATOR = '-' startingCoord="A-1" shipSize=1	nextCoord="A-1"
2	START_ROW_COORD = 3 SEPARATOR = '-' startingCoord="A-1" shipSize=5	nextCoord="A-5"
3	START_ROW_COORD = 3 SEPARATOR = '-' startingCoord="A-1" shipSize=3	nextCoord="A-3"
4	START_ROW_COORD = 3 SEPARATOR = '-' startingCoord="A-1"	CASO IMPOSSIBILE: shipSize non potrà mai essere un valore inferiore a 1

	shipSize=0	(input controllato da funzione chiamante)
5	START_ROW_COORD = 3 SEPARATOR = ' - ' startingCoord="A-1" shipSize=10	CASO IMPOSSIBILE: shipSize non potrà mai essere un valore maggiore a 5 (input controllato da funzione chiamante)
6	START_ROW_COORD = 3 SEPARATOR = ' - ' startingCoord="Z-1" shipSize=4	CASO IMPOSSIBILE: startingCoord non potrà mai avere come primo carattere un valore superiore a 'P' (valore controllato da funzione chiamante)
7	START_ROW_COORD = 3 SEPARATOR = ' - ' startingCoord="1-1" shipSize=4	CASO IMPOSSIBILE: startingCoord non potrà mai avere come primo carattere un valore inferiore a 'A' (valore controllato da funzione chiamante)
8	START_ROW_COORD = 3 SEPARATOR = ' - ' startingCoord="A-17" shipSize=4	CASO IMPOSSIBILE: in startingCoord dopo '-' ci sarà sempre un numero minore o uguale a 16 (input controllato da funzione chiamante)
9	START_ROW_COORD = 3 SEPARATOR = ' - ' startingCoord="A-0" shipSize=4	CASO IMPOSSIBILE: in startingCoord dopo '-' ci sarà sempre un numero maggiore o uguale a 1 (input controllato da funzione chiamante)
10	START_ROW_COORD = 3 SEPARATOR = ' - ' startingCoord=""	CASO IMPOSSIBILE: StartingCoord non può essere vuoto

	shipSize=4	(input controllato da funzione chiamante)
11	START_ROW_COORD = 3 SEPARATOR = ' - ' startingCoord="A1" shipSize=4	CASO IMPOSSIBILE: StartingCoord non contiene un formato valido (input controllato da funzione chiamante)

buildHorizontalCoord**INPUT**

Nome	Descrizione	Tipo	Vincoli
startingCoord	Coordinata di partenza della nave	Stringa	Formato: riga-colonna
shipSize	Dimensione della nave	Numero intero	shipSize >= MIN_SHIP_SIZE AND shipSize <= MAX_SHIP_SIZE
START_ROW_COORD	Inizio della riga nelle coordinate	Numero intero	3
SEPARATOR	Valore costante che separa una riga da una colonna	Carattere	SEPARATOR = ' - '

OUTPUT

Nome	Descrizione	Tipo	Vincoli
nextCoord	Coordinata successiva calcolata	Stringa	Formato: riga-colonna

Descrizione della funzione: Costruisce e restituisce la successiva coordinata, prendendo in considerazione la coordinata di partenza, le dimensioni della nave e la sua direzione.

Numero di test	INPUT	OUTPUT
1	SEPARATOR = ' - ' startingCoord="A-1" shipSize=1 START_ROW_COORD = 3	nextCoord="A-1"
2	SEPARATOR = ' - ' startingCoord="A-1" shipSize=5	nextCoord="E-1"

3	SEPARATOR = ' - ' startingCoord="A-1" shipSize=3 START_ROW_COORD = 3	nextCoord="C-1"
4	SEPARATOR = ' - ' startingCoord="A-1" shipSize=0 START_ROW_COORD = 3	CASO IMPOSSIBILE: shipSize non potrà mai essere un valore inferiore a 1 (input controllato da funzione chiamante)
5	SEPARATOR = ' - ' startingCoord = "A-1" shipSize = 10 START_ROW_COORD = 3	CASO IMPOSSIBILE: shipSize non potrà mai essere un valore maggiore a 5 (input controllato da funzione chiamante)
6	SEPARATOR = ' - ' startingCoord = "Z-1" shipSize = 4 START_ROW_COORD = 3	CASO IMPOSSIBILE: startingCoord non potrà mai avere come primo carattere un valore superiore a 'P' (valore controllato da funzione chiamante)
7	SEPARATOR = ' - ' startingCoord = "1-1" shipSize = 4 START_ROW_COORD = 3	CASO IMPOSSIBILE: startingCoord non potrà mai avere come primo carattere un valore inferiore a 'A' (valore controllato da funzione chiamante)
8	SEPARATOR = ' - ' startingCoord = "A-17" shipSize = 4 START_ROW_COORD = 3	CASO IMPOSSIBILE: in startingCoord dopo ' - ' ci sarà sempre un numero minore o uguale a 16 (input controllato da funzione chiamante)
9	SEPARATOR = ' - ' startingCoord="A-0" shipSize = 4 START_ROW_COORD = 3	CASO IMPOSSIBILE: in startingCoord dopo ' - ' ci sarà sempre un numero maggiore o uguale a 1 (input controllato da funzione chiamante)
10	SEPARATOR = ' - '	CASO IMPOSSIBILE:

	startingCoord = "" shipSize = 4 START_ROW_COORD = 3	StartingCoord non può essere vuoto (input controllato da funzione chiamante)
11	START_ROW_COORD = 3 SEPARATOR = ' - ' startingCoord = "A1" shipSize = 4 START_ROW_COORD = 3	CASO IMPOSSIBILE: StartingCoord non contiene un formato valido (input controllato da funzione chiamante)

concatCoordinates**INPUT**

Nome	Descrizione	Tipo	Vincoli
startCell	Prima coordinata della nave	Stringa	Formato: riga-colonna
endCell	Ultima coordinata della nave	Stringa	Formato: riga-colonna
COORD_SEPARATOR	Separatore di coordinate	Carattere	' - '

OUTPUT

Nome	Descrizione	Tipo	Vincoli
dest	Concatenazione delle coordinate	Stringa	Formato: startCell_endCell

Descrizione: Concatena startCell ed endCell separandole da COORD_SEPARATOR

Numero di test	INPUT	OUTPUT
1	COORD_SEPARATOR = ' - ' startCell = "A-10" endCell = "A_11"	Dest = "A-10_A-11"
2	COORD_SEPARATOR = ' - ' startCell = "" endCell = "A-11"	CASO IMPOSSIBILE: startCell non può essere una stringa vuota (input controllato da funzione chiamante)
3	COORD_SEPARATOR = ' - ' startCell = "A-10" endCell = ""	CASO IMPOSSIBILE: endCell non può essere una stringa vuota

		(input controllato da funzione chiamante)
4	COORD_SEPARATOR = ' _ ' startCell="A-10" endCell="A-16"	CASO IMPOSSIBILE: il range indicato da startCell e endCell è di un numero di celle superiore a 5 (input controllato da funzione chiamante)
5	COORD_SEPARATOR = ' _ ' startCell="A-14" endCell="A-10"	CASO IMPOSSIBILE: startCell e endCell non sono coordinate valide tra loro (input controllato da funzione chiamante)
6	COORD_SEPARATOR = ' _ ' startCell="V-10" endCell="A-10"	CASO IMPOSSIBILE: startCell deve contenere una coordinata valida (la colonna deve essere minore o uguale di 'P') (input controllato da funzione chiamante)
7	COORD_SEPARATOR = ' _ ' startCell="O-10" endCell="A-10"	CASO IMPOSSIBILE: startCell deve contenere una coordinata valida (la colonna deve essere maggiore o uguale di 'A') (input controllato da funzione chiamante)
8	COORD_SEPARATOR = ' _ ' startCell="A-17" endCell="A-10"	CASO IMPOSSIBILE: startCell deve contenere una coordinata valida (la riga deve essere minore o uguale di 16) (input controllato da funzione chiamante)
9	COORD_SEPARATOR = ' _ ' startCell="A-0" endCell="A-10"	CASO IMPOSSIBILE: startCell deve contenere una coordinata valida (la riga deve essere maggiore o uguale di 1) (input controllato da funzione chiamante)

10	COORD_SEPARATOR = ' _ ' startCell="A10" endCell="A-10"	CASO IMPOSSIBILE: startCell deve contenere una coordinata valida (formato non valido) (input controllato da funzione chiamante)
11	COORD_SEPARATOR = ' _ ' startCell="A-10" endCell="V-10"	CASO IMPOSSIBILE: endCell deve contenere una coordinata valida (la colonna deve essere minore o uguale di 'P') (input controllato da funzione chiamante)
12	COORD_SEPARATOR = ' _ ' startCell="A-10" endCell="O-10"	CASO IMPOSSIBILE: endCell deve contenere una coordinata valida (la colonna deve essere maggiore o uguale di 'A') (input controllato da funzione chiamante)
13	COORD_SEPARATOR = ' _ ' startCell="A-10" endCell="A-17"	CASO IMPOSSIBILE: endCell deve contenere una coordinata valida (la riga deve essere minore o uguale di 16) (input controllato da funzione chiamante)
14	COORD_SEPARATOR = ' _ ' startCell="A-10" endCell="A-0"	CASO IMPOSSIBILE: endCell deve contenere una coordinata valida (la riga deve essere maggiore o uguale di 1) (input controllato da funzione chiamante)
15	COORD_SEPARATOR = ' _ ' startCell="A-10" endCell="A10"	CASO IMPOSSIBILE: endCel deve contenere una coordinata valida (formato non valido) (input controllato da funzione chiamante)

getFirstCell**INPUT:**

Nome	Descrizione	Tipo	Vincoli
coords	Range di coordinate di una nave	Stringa	Formato: primaCella_ultimaCella
COORD_SEPARATOR	Separatore di coordinate	Carattere	'_'

OUTPUT:

Nome	Descrizione	Tipo	Vincoli
cell	Prima cella della nave	Stringa	Formato: riga-colonna

DESCRIZIONE: Recupera la prima coordinata da coords

TEST

Numero di test	INPUT	OUTPUT
1	COORD_SEPARATOR='_' Coords=""	CASO IMPOSSIBILE Coords non può essere vuoto (input controllato da funzione chiamante)
2	COORD_SEPARATOR='_' Coords="_A-10"	CASO IMPOSSIBILE '_' deve essere preceduto da una coordinata (input controllato da funzione chiamante)
3	COORD_SEPARATOR = '_' Coords = "A11_A-10"	CASO IMPOSSIBILE '_' deve essere preceduto da una coordinata valida (input controllato da funzione chiamante)
4	COORD_SEPARATOR='_' Coords="A-11_A-10"	CASO IMPOSSIBILE Coords non ha coordinate valide

Casi di test

Gruppo 12 – Curri Mattia, Festa Donato, Fontana Emanuele, Labianca Stefano

		(input controllato da funzione chiamante)
5	COORD_SEPARATOR = ' _' Coords = "A-9_A-10"	cell="A-9"
6	COORD_SEPARATOR = ' _' Coords = "A-10_"	CASO IMPOSSIBILE ' _ ' deve essere seguito da una coordinata (input controllato da funzione chiamante)

getLastCell

Nome	Descrizione	Tipo	Vincoli
coords	Range di coordinate di una nave	Stringa	Formato: primaCella_ultimaCella
COORD_SEPARATOR	Separatore di coordinate	Carattere	' _ '

OUTPUT:

Nome	Descrizione	Tipo	Vincoli
cell	Ultima cella della nave	Stringa	Formato: riga-colonna

Descrizione della funzione: Recupera l'ultima coordinata da coords

Numero di test	INPUT	OUTPUT
1	COORD_SEPARATOR = ' _' Coords = ""	CASO IMPOSSIBILE Coords non può essere vuoto (input controllato da funzione chiamante)
2	COORD_SEPARATOR = ' _' Coords = "A-10_"	CASO IMPOSSIBILE ' _ ' deve essere seguito da una coordinata (input controllato da funzione chiamante)

3	COORD_SEPARATOR=' _' Coords="A-11_Z-10"	CASO IMPOSSIBILE ' _ ' deve essere seguito da una coordinata valida (input controllato da funzione chiamante)
4	COORD_SEPARATOR=' _' Coords="A-11_A-10"	CASO IMPOSSIBILE Coords non ha coordinate valide (input controllato da funzione chiamante)
5	COORD_SEPARATOR=' _' Coords="A-9_A-10"	cell="A-10"
6	COORD_SEPARATOR=' _' Coords=" _A-10"	CASO IMPOSSIBILE ' _ ' deve essere preceduto da una coordinata (input controllato da funzione chiamante)

buildPlayerNavy**INPUT**

Nome	Descrizione	Tipo	Vincoli
player	player a cui far inserire le navi nel suo playGround	Player	/
MAX_SHIP_AMOUNT	Numero intero	Numero massimo di navi in un playground	15
MAX_SHIP_SIZE	Numero intero	Dimensione massima di una nave	5

OUTPUT

Nome	Descrizione	Tipo	Vincoli
player	player con le navi inserite nel suo playGround	Player	/

Casi di test:

[illegible]

	<pre>Ships = { } }</pre>	<pre>Ships = { Ship= { Label='a' Size=5 Coords="B-1_B-5" Direction='V' lifePoints=5 } Ship= { Label='b' Size=4 Coords="D-4_G-4" Direction='O' lifePoints=4 } Ship= { Label='c' Size=4 Coords="F-7_I-7" Direction='O' lifePoints=4 } Ship= { Label='d' Size=3 Coords="C-10_E-10" Direction='O' lifePoints=3 } Ship= { Label='e' Size=3 Coords="J-10_L-10" Direction='O' lifePoints=3 } Ship= { Label='f' Size=3 Coords="J-1_L-1" Direction='O' lifePoints=3 } Ship= { Label='g' Size=2</pre>
--	---------------------------------------	---

		<pre>Coords="A-14_A-15" Direction='V' lifePoints=2 } Ship= { Label='h' Size=2 Coords="C-14_C-15" Direction='V' lifePoints=2 } Ship= { Label='i' Size=2 Coords="E-14_E-15" Direction='V' lifePoints=2 } Ship= { Label='j' Size=2 Coords="P-1_P-2" Direction='V' lifePoints=2 } Ship= { Label='k' Size=1 Coords="P-4_P-4" Direction='V' lifePoints=1 } Ship= { Label='l' Size=1 Coords="P-8_P-8" Direction='V' lifePoints=1 } Ship= { Label='m' Size=1 Coords="L-12_L-12" Direction='V' lifePoints=1</pre>
--	--	---

		<pre> } Ship= { Label='n' Size=1 Coords="N-14_N-14" Direction='V' lifePoints=1 } Ship= { Label='o' Size=1 Coords="H-12_H-12" Direction='O' lifePoints=1 } } } </pre>
--	--	--

buildShipCoordinate**INPUT**

Nome	Descrizione	Tipo	Vincoli
column	Primo valore della coordinata	Carattere	Column >= 'A' AND column <= 'P'
row	Secondo valore della coordinata	Stringa	Contiene un numero compreso tra 1 e 16
START_ROW_COORD	Inizio della riga nelle coordinate	Numero intero	3
SEPARATOR	Separatore tra riga e colonna	Carattere	'-'

OUTPUT

Nome	Descrizione	Tipo	Vincoli
coord	Coordinata della nave	Stringa	Formato: Colonna-Riga

LAVORO

Nome	Descrizione	Tipo	Vincoli
i	Indice di scorrimento per le stringhe	Numero intero	i > 0
j	Indice di scorrimento per le stringhe	Numero intero	j > 0

Descrizione della funzione: la funzione riceve in input la riga e colonna della nave inserite dal giocatore e restituisce la stringa coord in formato colonna-riga.

Numero di test	Input	Output
1	column = 'A' row = 1	A-1
2	column = 'P' row = 3	P-3
3	column = 'L' row = 3	L-3
4	column = 'A' row = 17	CASO IMPOSSIBILE row non potrà mai essere un valore maggiore a 16 (input controllato da funzione chiamante)
5	column = 'A' row = 0	CASO IMPOSSIBILE row non potrà mai essere un valore minore a 1 (input controllato da funzione chiamante)
6	column = 'a' row = 1	CASO IMPOSSIBILE column non potrà mai essere un valore maggiore a 'P' (input controllato da funzione chiamante)
7	column = '2' row = 16	CASO IMPOSSIBILE column non potrà mai essere un valore minore a 'A' (input controllato da funzione chiamante)
8	column = "" row = 0	CASO IMPOSSIBILE Column non può essere vuoto (input controllato da funzione chiamante)

buildNextCoord

INPUT

Nome	Descrizione	Tipo	Vincoli
Direction	Direzione della nave	Carattere	V = Verticale O = Orizzontale
startingCoord	Coordinata di partenza della nave	Stringa	Formato: riga-colonna
shipSize	Dimensione della nave	Numero intero	shipSize >= MIN_SHIP_SIZE AND shipSize <= MAX_SHIP_SIZE

OUTPUT

Nome	Descrizione	Tipo	Vincoli
nextCoord	Coordinata successiva calcolata	Stringa	Formato: riga-colonna

Descrizione della funzione: costruisce e restituisce la successiva coordinata, prendendo in considerazione la coordinata di partenza, le dimensioni della nave e la sua direzione. La coordinata costruita verrà salvata all'interno di nextCoord.

Numero di test	Input	Output
1	startingCoord="A-1" shipSize=1 direction = 'V'	nextCoord="A-1"
2	startingCoord="A-1" shipSize=1 direction = 'O'	nextCoord="A-1"
3	startingCoord="A-1" shipSize=1 direction = ''	CASO IMPOSSIBILE direction sarà sempre un valore valido (V o O) (input controllato da funzione chiamante)
4	startingCoord="A-1" shipSize=5 direction = 'V'	nextCoord="A-5"
5	startingCoord="A-1" shipSize=5 direction = 'O'	nextCoord="E-1"
6	startingCoord="A-1" shipSize=5 direction = ''	CASO IMPOSSIBILE direction sarà sempre un valore valido (V o O). (input controllato da funzione chiamante)

7	startingCoord="A-1" shipSize=3 direction = 'V'	nextCoord="A-3"
8	startingCoord="A-1" shipSize=3 direction = 'O'	nextCoord="C-1"
9	startingCoord="A-1" shipSize=3 direction = ''	CASO IMPOSSIBILE direction sarà sempre un valore valido (V o O). (input controllato da funzione chiamante)
10	startingCoord="A-1" shipSize = 0 direction = 'V'	CASO IMPOSSIBILE shipSize non potrà mai essere un valore inferiore a 1 (input controllato da funzione chiamante)
11	startingCoord="A-1" shipSize = 0 direction = 'O'	CASO IMPOSSIBILE shipSize non potrà mai essere un valore inferiore a 1 (input controllato da funzione chiamante)
12	startingCoord = "A-1" shipSize = 10 direction = 'V'	CASO IMPOSSIBILE shipSize non potrà mai essere un valore maggiore a 5 (input controllato da funzione chiamante)
13	startingCoord="A-1" shipSize = 10 direction = 'O'	CASO IMPOSSIBILE shipSize non potrà mai essere un valore maggiore a 5 (input controllato da funzione chiamante)
14	startingCoord="Z-1" shipSize = 4 direction = 'V'	CASO IMPOSSIBILE: startingCoord non potrà mai avere come primo carattere un valore superiore a 'P'

		(valore controllato da funzione chiamante)
15	startingCoord = "Z-1" shipSize = 4 direction = 'O'	CASO IMPOSSIBILE: startingCoord non potrà mai avere come primo carattere un valore superiore a 'P' (valore controllato da funzione chiamante)
16	startingCoord="1-1" shipSize=4 direction = 'V'	CASO IMPOSSIBILE: startingCoord non potrà mai avere come primo carattere un valore inferiore a 'A' (valore controllato da funzione chiamante)
17	startingCoord="1-1" shipSize = 4 direction = 'O'	CASO IMPOSSIBILE: startingCoord non potrà mai avere come primo carattere un valore inferiore a 'A' (valore controllato da funzione chiamante)
18	startingCoord="A-17" shipSize=4 direction = 'V'	CASO IMPOSSIBILE: in startingCoord dopo '-' ci sarà sempre un numero inferiore o uguale a 16 (input controllato da funzione chiamante)
19	startingCoord="A-17" shipSize=4 direction = 'O'	CASO IMPOSSIBILE: in startingCoord dopo '-' ci sarà sempre un numero minore o uguale a 16 (input controllato da funzione chiamante)
20	startingCoord="A-0" shipSize=4 direction = 'V'	CASO IMPOSSIBILE:

		in startingCoord dopo '-' ci sarà sempre un numero maggiore o uguale a 1 (input controllato da funzione chiamante)
21	startingCoord="A-0" shipSize = 4 direction = 'O'	CASO IMPOSSIBILE: in startingCoord dopo '-' ci sarà sempre un numero maggiore o uguale a 1 (input controllato da funzione chiamante)
22	startingCoord="" shipSize=4 direction = 'V'	CASO IMPOSSIBILE: in startingCoord ci saranno valori validi (input controllato da funzione chiamante)
23	startingCoord="" shipSize=4 direction = 'O'	CASO IMPOSSIBILE: in startingCoord ci saranno valori validi (input controllato da funzione chiamante)

createShip**INPUT**

Nome	Descrizione	Tipo	Vincoli
size	Dimensione della nave	Numero intero	size >= MIN_SHIP_SIZE AND size <= MAX_SHIP_SIZE
Number	Numero della nave	Numero intero	number >= 1 AND number <= MAX_SHIP_AMOUNT
Playground	Playground nel quale deve essere caricata la nave	Array bidimensionale di caratteri	Dimensione 16x16 Caratteri consentiti: Da 'a' a 'o' per le navi '~' per il mare 'x' nave colpita '*' nave affondata

START_LOWERCASE_ASCII	Prima lettera dell'alfabeto minuscola in ASCII	Numero intero	97
-----------------------	--	---------------	----

OUTPUT

Nome	Descrizione	Tipo	Vincoli
ship	Nave del giocatore	Ship	/

LAVORO

Nome	Descrizione	Tipo	Vincoli
cell	Prima coordinata della nave	Stringa	Formato: riga-colonna
nextCell	Ultima coordinata della nave	Stringa	Formato: riga-colonna
coords	Range di coordinate della nave	Stringa	Formato: primaCella_ultimaCella
column	Indica la colonna della prima coordinata della nave	Carattere	Column >= 'A' AND column <= 'P'
row	Contiene il valore della riga della prima coordinata della nave	Stringa	Contiene un numero tra 1 e 16
direction	Orientamento della nave	Stringa	V = Verticale O = Orizzontale
isImpossibleToDo	Indica l'impossibilità di inserire una nave nel range di coordinate scelto	Numero intero	1 = Impossibile inserire nave 0 = La nave può essere inserita

Descrizione della funzione: Costruisce e restituisce una nave. La funzione verifica anche se è possibile inserirla all'interno della mappa di gioco del giocatore.

Numero di test	Input	Lavoro	Output
1	size = 5 number = 1 playground = ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~	direction = 'V' (acquisito da tastiera dalla funzione getShipDirection()) column = 'A' (acquisito da tastiera dalla	Ship = { label = 'a' lifePoints = 5 size = 5 coords = "A-1_A-5" direction = "V" }

	<pre> ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ START_LOWERCASE_ASCII = 97 </pre>	<p>funzione getColumn())</p> <p>row = 1 (acquisito da tastiera dalla funzione getRow())</p>	
2	<pre> size = 1 number = 15 playground = ~~~~~ START_LOWERCASE_ASCII = 97 </pre>	<pre> direction = 'V' (acquisito da tastiera) column = 'A' (acquisito da tastiera) row = 1 (acquisito da tastiera) </pre>	<pre> Ship = { label = 'o' lifePoints = 1 size = 1 coords = "A-1_A-1" direction = "V" } </pre>
3	<pre> size = 3 number = 6 playground = ~~~~~ START_LOWERCASE_ASCII = 97 </pre>	<pre> direction = 'O' (acquisito da tastiera) column = 'F' (acquisito da tastiera) row = 4 (acquisito da tastiera) </pre>	<pre> Ship = { label = 'f' lifePoints = 3 size = 3 coords = "F-4_F-6" direction = 'O' } </pre>

	<pre> ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ START_LOWERCASE_ASCII = 97 </pre>		
4	<pre> size = 0 number = 1 playground = ~~~~~ START_LOWERCASE_ASCII = 97 </pre>	<pre> direction = 'V' (acquisito da tastiera) column = 'A' (acquisito da tastiera) row = 1 (acquisito da tastiera) </pre>	<p>CASO IMPOSSIBILE: size non potrà mai essere minore di 1</p> <p>(input controllato da funzione chiamante)</p>
5	<pre> size = 6 number = 1 playground = ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ </pre>	<pre> direction = 'V' (acquisito da tastiera) column = 'A' (acquisito da tastiera) row = 1 (acquisito da tastiera) </pre>	<p>CASO IMPOSSIBILE size non potrà mai essere maggiore di 5</p> <p>(input controllato da funzione chiamante)</p>

	<pre> ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ START_LOWERCASE_ASCII = 97 </pre>		
6	<pre> size = 1 number = 0 playground = ~~~~~ START_LOWERCASE_ASCII = 97 </pre>	<pre> direction = 'V' (acquisito da tastiera) column = 'A' (acquisito da tastiera) row = 1 (acquisito da tastiera) </pre>	<p>CASO IMPOSSIBILE number non potrà mai essere minore di 1</p> <p>(input controllato da funzione chiamante)</p>
7	<pre> size = 1 number = 16 playground = ~~~~~ START_LOWERCASE_ASCII = 97 </pre>	<pre> direction = 'V' (acquisito da tastiera) column = 'A' (acquisito da tastiera) row = 1 (acquisito da tastiera) </pre>	<p>CASO IMPOSSIBILE number non potrà mai essere maggiore di 15</p> <p>(input controllato da funzione chiamante)</p>

Casi di test

Gruppo 12 – Curri Mattia, Festa Donato, Fontana Emanuele, Labianca Stefano

	~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ START_LOWERCASE_ASCII = 97		
--	----------------------------------------------------------------------------------------------------	--	--

### toUpperCase

#### INPUT:

Nome	Descrizione	Tipo	Vincoli
letter	Lettera da trasformare	carattere	Compresa tra 'a' e 'z'
LOWER_TO_UPPER	Conversione delle parole da minuscolo a maiuscolo o viceversa	Numero intero	32

#### OUTPUT:

Nome	Descrizione	Tipo	Vincoli
letter	Lettera trasformata	carattere	Compresa tra 'A' e 'Z'

**Descrizione della funzione:** la funzione riceve in input un carattere minuscolo da trasformare in un carattere maiuscolo

Numero di test	Input	Output
1	letter = 'a' LOWER_TO_UPPER = 32	letter = 'A'
2	letter = 'A' LOWER_TO_UPPER = 32	letter = 'A'
3	letter = 'f' LOWER_TO_UPPER = 32	letter = 'F'
4	letter = 'p' LOWER_TO_UPPER = 32	letter = 'P'
5	letter = "" LOWER_TO_UPPER = 32	letter = ""
6	letter = 'z' LOWER_TO_UPPER = 32	letter = 'Z'

### getLength

#### INPUT:

Casi di test

Gruppo 12 – Curri Mattia, Festa Donato, Fontana Emanuele, Labianca Stefano

Nome	Descrizione	Tipo	Vincoli
str	Stringa a cui calcolare la lunghezza	Stringa	\

**OUTPUT:**

Nome	Descrizione	Tipo	Vincoli
i	Lunghezza della stringa	intero	i >= 0

**DESCRIZIONE DELLA FUNZIONE:** Restituisce la lunghezza di una stringa

Numero di test	INPUT	OUTPUT
1	str = ""	0
2	str = "4"	1
3	str = "A-5"	3
4	str = "A-13"	4
5	str = "A-7_A-10"	8

**numberToString**

**INPUT:**

Nome	Descrizione	Tipo	Vincoli
number	Numero da convertire	Intero	\
MIN_DIGIT	0 in codice ASCII	Numero intero	48
START_ROW_COORD	Inizio della riga nelle coordinate	Numero intero	3

**OUTPUT:**

Nome	Descrizione	Tipo	Vincoli
buffer	Stringa che contiene il valore di number	Stringa	\

**DESCRIZIONE DELLA FUNZIONE:** Converte un numero naturale in una stringa.

Numero di test	INPUT	OUTPUT
1	Number = 10 MIN_DIGIT = 48 START_ROW_COORD = 3	Buffer = "10"
2	Number = 5 MIN_DIGIT = 48 START_ROW_COORD = 3	Buffer = "5"
3	Number = -1 MIN_DIGIT = 48 START_ROW_COORD = 3	CASO IMPOSSIBILE Number non potrà mai essere un valore minore di 1

		(input controllato da funzione chiamante)
4	Number = 20 MIN_DIGIT = 48 START_ROW_COORD = 3	CASO IMPOSSIBILE Number non potrà mai essere un valore maggiore di 16 (input controllato da funzione chiamante)

**remainder****INPUT:**

Nome	Descrizione	Tipo	Vincoli
dividend	Dividendo dell'operazione	intero	\
divider	Divisore dell'operazione	Intero	divider $\neq$ 0

**OUTPUT:**

Nome	Descrizione	Tipo	Vincoli
dividend	Resto della divisione	intero	\

**DESCRIZIONE DELLA FUNZIONE:** la funzione restituisce il resto di una divisione

Numero di test	INPUT	OUTPUT
1	dividend = 4 divider = 2	dividend = 0
2	dividend = 3 divider = 2	dividend = 1
3	dividend = -1 divider = 2	dividend = -1
4	dividend = 3 divider = 0	CASO IMPOSSIBILE: divider deve essere diverso da 0

**stringToNumber****INPUT:**

Nome	Descrizione	Tipo	Vincoli
str	Stringa da convertire in un numero intero	Stringa	\
len	Lunghezza della stringa da convertire	Intero	\
MIN_DIGIT	0 in codice ASCII	Numero intero	48
MAX_DIGIT	9 in codice ASCII	Numero intero	57

**OUTPUT:**

Nome	Descrizione	Tipo	Vincoli
number	Numero convertito	Intero	\

**DESCRIZIONE DELLA FUNZIONE:** Converte una stringa in un numero naturale, andando ad escludere quei caratteri che non fanno parte del range in codice ASCII delle cifre decimali.

Numero di test	INPUT	OUTPUT
1	str = "12" len = 2 MIN_DIGIT = 48 MAX_DIGIT = 57	number = 12
2	str = "45" len = 2 MIN_DIGIT = 48 MAX_DIGIT = 57	number = 45
2	str = "45" len = 3 MIN_DIGIT = 48 MAX_DIGIT = 57	CASO IMPOSSIBILE len sarà sempre un valore valido (in questo caso len di str dovrebbe essere uguale a 2)  (input controllato da funzione chiamante)
3	str = "a4l5" len = 4 MIN_DIGIT = 48 MAX_DIGIT = 57	number = 45
4	str = "a1" len = 2 MIN_DIGIT = 48 MAX_DIGIT = 57	number = 0

**isImpossible****INPUT:**

Nome	Descrizione	Tipo	Vincoli
cell	Posizione di partenza della nave	Stringa	Formato: riga-colonna
direction	Direzione della nave	carattere	V = Verticale O = Orizzontale
coords	Range di coordinate della nave	Stringa	Formato: primaCella_ultimaCella
size	Dimensioni della nave	Stringa	Size >= MIN_SHIP_SIZE AND size <= MAX_SHIP_SIZE



Gruppo 12 – Curri Mattia, Festa Donato, Fontana Emanuele, Labianca Stefano

**OUTPUT:**

**DESCRIZIONE DELLA FUNZIONE:** Verifica se è possibile inserire, all'interno del playground, una nave in una data posizione

<b>Numero di test</b>	<b>INPUT</b>	<b>OUTPUT</b>
1	cell = "A-1" direction = 'V' coords = "A-1_A-4" size = 4 playground = ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	error = 0
2	cell = "D-1" direction = 'V' coords = "D-1_D-4" size = 4 playground =	error = 1

	<pre>a a a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~</pre>	
3	<pre>cell = "A-1" direction = 'O' coords = "A-1_D-1" size = 4     playground = a a a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~</pre>	error = 1
4	<pre>cell = "D-1" direction = 'O' coords = "D-1_G-1" size = -1 playground = a a a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~</pre>	<p>CASO IMPOSSIBILE size non può essere un valore minore di 1  (input controllato da funzione chiamante)</p>

	<pre> ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ </pre>	
5	<pre> cell = "A-1" direction = 'V' coords = "G-1_A-4" size = 4 playground = ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ </pre>	<p>CASO IMPOSSIBILE          coords sarà sempre un valore valido</p> <p>(input controllato da funzione chiamante)</p>
5	<pre> cell = "D-1" direction = 'O' coords = "D-1_G-1" size = 6 playground = ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ </pre>	<p>CASO IMPOSSIBILE          size non può essere un valore maggiore di 5</p> <p>(input controllato da funzione chiamante)</p>

	<pre> ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ </pre>	
6	<pre> cell = "A-1" direction = 'L' coords = "A-1_E-1" size = 5 playground = ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ </pre>	<p>CASO IMPOSSIBILE</p> <p>direction deve avere un valore valido (V o O)</p> <p>(input controllato da funzione chiamante)</p>
7	<pre> cell = "" direction = 'V' coords = "A-1_E-1" size = 5 playground = ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ </pre>	<p>CASO IMPOSSIBILE</p> <p>cell non può essere vuoto, coords dipende da cell</p> <p>(input controllato da funzione chiamante)</p>

	<p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p>	
8	<p>cell = "A-1"</p> <p>direction = 'V'</p> <p>coords = ""</p> <p>size = 5</p> <p>playground =</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p>	<p>CASO IMPOSSIBILE</p> <p>coords non può essere vuoto</p> <p>(input controllato da funzione chiamante)</p>
9	<p>cell = "A-1"</p> <p>direction = "</p> <p>coords = "A-1_E-1"</p> <p>size = 5</p> <p>playground =</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p> <p>~ ~ ~ ~ ~</p>	<p>CASO IMPOSSIBILE</p> <p>direction non può essere vuoto</p> <p>(input controllato da funzione chiamante)</p>

### checkBoundaries

#### INPUT:

Nome	Descrizione	Tipo	Vincoli
row	Riga di partenza della nave	intero	row >= TABLE_MIN AND row <= TABLE_MAX
column	Colonna di partenza della nave	intero	column >= TABLE_MIN AND column <= TABLE_MAX
size	Dimensioni della nave	intero	Size >= MIN_SHIP_SIZE AND size <= MAX_SHIP_SIZE
direction	Direzione della nave	Carattere	V = Verticale O = Orizzontale
TABLE_MAX	Riga/Colonna massima	Numero intero	16

#### OUTPUT:

Nome	Descrizione	Tipo	Vincoli
error	Valore numerico pari a 1 o 0 che rappresenta l'esito del controllo	intero	1 = La nave viola i limiti della mappa 0 = la nave non viola i limiti della mappa

**DESCRIZIONE DELLA FUNZIONE:** Controlla se la posizione della nave non vada oltre i limiti della mappa di gioco.

Numero di test	INPUT	OUTPUT
1	row = 1 column = 1 size = 5 direction = 'V' TABLE_MAX = 16	error = 0
2	row = 10 column = 5 size = 4 direction = 'O' TABLE_MAX = 16	error = 0
3	row = 16 column = 1 size = 5 direction = 'V' TABLE_MAX = 16	error = 1

4	row = 1 column = 16 size = 4 direction = 'O' TABLE_MAX = 16	error = 1
5	row = 13 column = 5 size = 5 direction = 'V' TABLE_MAX = 16	error = 1
6	row = 12 column = 15 size = 5 direction = 'P' TABLE_MAX = 16	CASO IMPOSSIBILE direction non potrà avere un valore diverso da 'O' e 'V'  (input controllato da funzione chiamante)
7	row = -12 column = 11 size = 5 direction = 'V' TABLE_MAX = 16	CASO IMPOSSIBILE row non può essere minore di 1  (input controllato da funzione chiamante)
8	row = 112 column = 11 size = 5 direction = 'V' TABLE_MAX = 16	CASO IMPOSSIBILE row non può essere maggiore di 16  (input controllato da funzione chiamante)
9	row = 12 column = -11 size = 5 direction = 'V' TABLE_MAX = 16	CASO IMPOSSIBILE column non può essere minore di 1  (input controllato da funzione chiamante)
10	row = 12 column = 111 size = 5 direction = 'V' TABLE_MAX = 16	CASO IMPOSSIBILE column non può essere maggiore di 16  (input controllato da funzione chiamante)
11	row = 12 column = 11 size = 0 direction = 'V' TABLE_MAX = 16	CASO IMPOSSIBILE size non può essere minore di 1  (input controllato da funzione chiamante)
12	row = 12 column = 11 size = 6 direction = 'V' TABLE_MAX = 16	CASO IMPOSSIBILE size non può essere maggiore di 5  (input controllato da funzione chiamante)

### checkCollisions

#### INPUT:

Nome	Descrizione	Tipo	Vincoli
playground	Playground di un giocatore	Array a due dimensioni di caratteri	Dimensione 16x16 Caratteri consentiti: da 'a' a 'o' per le navi '~' per il mare 'X' per nave colpita '*' per nave affondata
coords	Range di coordinate di una nave	Stringa	Formato: primaCella_secondaCella
TABLE_MIN	Riga/Colonna minima	Numero intero	1
TABLE_MAX	Riga/Colonna massima	Numero intero	16
WATER	Carattere che indica la presenza di acqua nel playground e nell'heatMap	Carattere	'~'

#### OUTPUT:

Nome	Descrizione	Tipo	Vincoli
error	Restituisce un valore numerico pari a 1 o 0 che rappresenta l'esito del controllo	intero	1 = la nave entra in collisione con altre navi 0 = nessuna collisione

**DESCRIZIONE DELLA FUNZIONE:** Verifica se una nave sta entrando in collisione con un'altra nave o se supera i confini della mappa di gioco

Numero di test	INPUT	OUTPUT
1	playground = ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	error = 0



[illegible]

## Casi di test

Gruppo 12 – Curri Mattia, Festa Donato, Fontana Emanuele, Labianca Stefano

	<pre> ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  coords = "C-1_C-5" TABLE_MIN = 1 TABLE_MAX = 16 WATER = '~' </pre>	
4	<pre> playground = ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  a a a a a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  coords = "C-1_C-18" TABLE_MIN = 1 TABLE_MAX = 16 WATER = '~' </pre>	<p>CASO IMPOSSIBILE</p> <p>coords conterrà sempre un formato valido</p> <p>(input controllato da funzione chiamante)</p>
5	<pre> playground = ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  a a a a a ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  coords = "" TABLE_MIN = 1 </pre>	<p>CASO IMPOSSIBILE</p> <p>coords non può essere vuoto</p> <p>(input controllato da funzione chiamante)</p>

	<div>TABLE_MAX = 16</div> <div>WATER = '~'</div>	
--	--------------------------------------------------	--