

Modulo Ship

Descrizione del modulo: Contiene le funzioni relative all'utilizzo delle navi

INTERFACCIA

getLabel	1
getSize	2
getLifePoints	2
getCoords	3
getDirection	3
setLabel	4
setSize.....	4
setLifePoints	5
setCoords	5
setDirection	6
getColumn	6
getShipDirection.....	7
IsValidRow	8
getRow	8
buildShipCoordinate	9
buildNextCoord	10
buildVerticalCoord	11
buildHorizontalCoord.....	12
concatCoordinates.....	13
pullColumn.....	14
pullRow	14
createShip.....	15
isImpossible	17
checkBoundaries.....	19
checkCollisions	20
getFirstCell.....	22
getLastCell.....	23

getLabel

INPUT

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

ship	Nave da cui prendere l'etichetta	ship	/
------	----------------------------------	------	---

OUTPUT

Nome	Descrizione	Tipo	Vincoli
Label	Carattere che identifica l'etichetta della nave	Carattere	Le label vanno dalla 'a' (prima nave) alla 'o' (quindicesima nave)

ALGORITMO

label := campo label di ship

getSize

Nome	Descrizione	Tipo	Vincoli
ship	Nave da cui prendere le sue dimensioni	Ship	/

OUTPUT

Nome	Descrizione	Tipo	Vincoli
Size	Dimensioni della nave	Numero intero	Size >= MIN_SHIP_SIZE AND size <= MAX_SHIP_SIZE

ALGORITMO

Size := campo size di ship

getLifePoints**INPUT**

Nome	Descrizione	Tipo	Vincoli
ship	Nave da cui estrarre i punti vita	ship	/

OUTPUT

Nome	Descrizione	Tipo	Vincoli
lifePoints	Punti vita della nave	Numero intero	lifePoints \geq 0 AND lifePoints \leq MAX_LIFE_POINTS

ALGORITMO

lifePoints := campo lifePoints di ship

getCoords

INPUT

Nome	Descrizione	Tipo	Vincoli
ship	Nave da cui prendere la cella di partenza e di arrivo	ship	/

OUTPUT

Nome	Descrizione	Tipo	Vincoli
coords	Contiene le celle della nave	Stringa	Formato: primaCella_ultimaCella

LAVORO

Nome	Descrizione	Tipo	Vincoli
i	Contatore per le posizioni all'interno delle stringhe	Numero intero	i > 0

ALGORITMO

i := 1

MENTRE (i-esimo elemento del campo coords di ship <> ENDSTRING)

 i-esimo elemento di coords := i-esimo elemento del campo coords di ship

 i := i + 1

FINE

i-esimo elemento di coords := ENDSTRING

getDirection

INPUT

Nome	Descrizione	Tipo	Vincoli
ship	Nave da cui prendere l'orientamento	ship	/

OUTPUT

Nome	Descrizione	Tipo	Vincoli
direction	Carattere che rappresenta	Carattere	V = Verticale O = Orizzontale

	l'orientamento della nave		
--	---------------------------	--	--

ALGORITMO

direction := campo direction di ship

setLabel**INPUT**

Nome	Descrizione	Tipo	Vincoli
ship	Nave a cui assegnare un'etichetta	Ship	/
Label	Etichetta da assegnare alla nave	Carattere	Le label vanno dalla 'a' (prima nave) alla 'o' (quindicesima nave)

OUTPUT

Nome	Descrizione	Tipo	Vincoli
ship	Nave aggiornata	Ship	/

ALGORITMO

campo label di ship := label

setSize

Nome	Descrizione	Tipo	Vincoli
ship	Nave a cui assegnare delle dimensioni	Ship	/
Size	Dimensioni della nave	Numero intero	Size >= MIN_SHIP_SIZE AND size <= MAX_SHIP_SIZE

OUTPUT

Nome	Descrizione	Tipo	Vincoli
ship	Nave aggiornata	Ship	/

ALGORITMO

campo size di ship := ship

setLifePoints**INPUT**

Nome	Descrizione	Tipo	Vincoli
ship	Nave a cui aggiornare i punti vita	ship	/
lifePoints	Punti vita della nave	Numero intero	lifePoints \geq 0 AND lifePoints \leq MAX_SHIP_SIZE

OUTPUT

Nome	Descrizione	Tipo	Vincoli
ship	Nave aggiornata	ship	/

ALGORITMO

campo lifePoints di ship := lifePoints

setCoords**INPUT**

Nome	Descrizione	Tipo	Vincoli
ship	Nave a cui assegnare la cella di partenza e di arrivo	ship	/
coords	Contiene la cella di partenza e di arrivo della nave	Stringa	Formato: primaCella_ultimaCella

OUTPUT

Nome	Descrizione	Tipo	Vincoli
ship	Nave aggiornata	ship	/

LAVORO

Nome	Descrizione	Tipo	Vincoli
i	Contatore per le posizioni all'interno delle stringhe	Numero intero	$i > 0$

ALGORITMO

$i := 1$

MENTRE (i-esimo elemento di coords \neq ENDSTRING)

i-esimo elemento del campo coords di ship := i-esimo elemento di coords

i := i + 1

FINE

i-esimo elemento del campo coords di ship := ENDSTRING

setDirection

INPUT

Nome	Descrizione	Tipo	Vincoli
ship	Nave a cui impostare la direzione	ship	/
direction	Direzione della nave	Carattere	V = Verticale O = Orizzontale

OUTPUT

Nome	Descrizione	Tipo	Vincoli
ship	Nave modificata	ship	/

ALGORITMO

campo direction di ship := direction

getColumn

INPUT

Nome	Descrizione	Tipo	Vincoli
column	Contiene il valore della colonna inserito dall'utente	Carattere	Compreso tra 'A' e 'P'

OUTPUT

Nome	Descrizione	Tipo	Vincoli
column	Contiene il valore della colonna inserito dall'utente	Carattere	Compreso tra 'A' e 'P'

LAVORO

Nome	Descrizione	Tipo	Vincoli
error	Indica se il carattere inserito è valido o meno	Numero intero	1 = carattere inserito non valido 0 = carattere inserito valido

ALGORITMO

ESEGUI

```

error := 0
StampaAVideo("Inserisci la colonna di partenza della nave: ")
column := LeggereDaTastiera()
column := toUpperCase(column)
SE ( (column < ' A ' ) OR (column > ' P ' ) )
    ALLORA
        error := 1
        StampaAVideo("Errore: devi inserire una lettera compresa tra A e P")

```

FINE

FINCHE (error = 1)

FINE

getShipDirection**INPUT**

Nome	Descrizione	Tipo	Vincoli
direction	Orientamento della nave, preso in input	Carattere	V = Verticale O = Orizzontale

OUTPUT

Nome	Descrizione	Tipo	Vincoli
direction	Orientamento della nave, preso in input	Carattere	V = Verticale O = Orizzontale

LAVORO

Nome	Descrizione	Tipo	Vincoli
error	Indica se il carattere inserito rappresenta o meno una direzione	Intero	1 = direzione non valida 0 = direzione valida

ALGORITMO

ESEGUI

```

error := 0
StampaAVideo("Inserire V per inserire la nave in verticale, oppure O per inserire la nave in
orizzontale: ")
direction := LeggereDaTastiera()
direction := toUpperCase(direction)
SE ( ( direction <> 'O' ) AND ( direction <> ' V ' ) )
    ALLORA
        error := 1
        StampaAVideo("Errore: inserire un'opzione valida")

```

FINE

FINCHE(error = 1)

FINE

IsValidRow

INPUT

Nome	Descrizione	Tipo	Vincoli
Row	Numero della riga da controllare	Numero intero	/
TABLE_MIN	Riga/Colonna minima	Numero intero	1
TABLE_MAX	Riga/Colonna massima	Numero intero	16

OUTPUT

Nome	Descrizione	Tipo	Vincoli
isValid	Indica se row contiene un valore valido o meno. Per essere valido deve essere compreso tra TABLE_MIN e TABLE_MAX	Numero intero	1 = valore valido 0 = valore non valido

ALGORITMO

isValid := 1

SE ((row < TABLE_MIN) OR (row > TABLE_MAX))

ALLORA

isValid := 0

StampareAVideo("Errore: devi inserire un numero compreso tra 1 e 16")

FINE

getRow

INPUT

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

numericRow	Contiene il valore della riga inserito dall'utente	Numero intero	numericRow >= TABLE_MIN AND numericRow <= TABLE_MAX
------------	--	---------------	---

OUTPUT

Nome	Descrizione	Tipo	Vincoli
row	Contiene il valore della riga inserito dall'utente	Stringa	/

LAVORO

Nome	Descrizione	Tipo	Vincoli
isValid	Indica se il valore inserito dall'utente è valido o meno	Numero intero	0 = valore non valido 1 = valore valido

ALGORITMO**ESEGUI**

StampaAVideo("Inserisci la riga di partenza della nave: ")

numericRow := LeggereDaTastiera()

isValid := isValidRow(numericRow)

FINCHE(isValid = 0)

FINE

row := numberToString(numericRow)

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	$j > 0$

ALGORITMO

```

i := 1
j := START_ROW_COORD
i-esimo valore di coord := column
(i + 1)-esimo valore di coord := SEPARATOR
MENTRE (i-esimo valore di row <> ENDSTRING)
    j-esimo valore di coord := i-esimo valore di row
    i := i + 1
    j := j + 1
FINE
j-esimo valore di coord := ENDSTRING

```

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

ALGORITMO

```

SE (direction = 'V')
    ALLORA
        nextCoord := buildVerticalCoord(startingCoord, shipSize)

```

ALTRIMENTI

SE (direction='O')

ALLORA

nextCoord := buildHorizontalCoord(startingCoord, shipSize)

FINE

FINE

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

LAVORO

Nome	Descrizione	Tipo	Vincoli
numericRow	Valore della riga della coordinata finale	Numero intero	numericRow >= TABLE_MIN AND numericRow <= TABLE_MAX
nextRow	Valore della riga della coordinata finale	Stringa	Contiene il valore di numericRow
i	Indice di scorrimento delle stringhe	Numero intero	i > 0
len	Lunghezza di startingCoord	Intero	Lunghezza compresa tra 3 e 4 (ENDSTRING escluso)

ALGORITMO

i := START_ROW_COORD

```

len := getLength(startingCoord)
numericRow := stringToNumber(startingCoord, len) + shipSize - 1
nextRow := numberToString(numericRow)
primo elemento di nextCoord := primo elemento di startingCoord
secondo elemento di nextCoord := SEPARATOR
MENTRE ((i - 2)-esimo elemento di nextRow <> ENDSTRING)
    i-esimo elemento di nextCoord := (i - 2)-esimo elemento di nextRow
    i := i + 1
FINE
i-esimo elemento di nextCoord := ENDSTRING

```

buildHorizontalCoord**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

LAVORO

Nome	Descrizione	Tipo	Vincoli
numericColumn	Valore della colonna della coordinata finale	Numero intero	numericColumn >= TABLE_MIN AND numericColumn <= TABLE_MAX
startingColumn	Colonna della prima coordinata	Carattere	startingColumn >= 'A' AND startingColumn <= 'P'
nextColumn	Colonna dell'ultima coordinata	Carattere	nextColumn >= 'A' AND nextColumn <= 'P'

i	Indice di scorrimento per le stringhe	Numero intero	i > 0
---	---------------------------------------	---------------	-------

ALGORITMO

i := START_ROW_COORD

startingColumn := **primo** elemento di startingCoord

numericColumn := getIntegerColumn(startingColumn) + shipSize - 1

nextColumn := getCharColumn(numericColumn)

primo elemento di nextCoord := nextColumn

secondo elemento di nextCoord := SEPARATOR

MENTRE (i-esimo elemento di startingCoord <> ENDSTRING)

 i-esimo elemento di nextCoord := i-esimo elemento di startingCoord

 i := i + 1

FINE

i-esimo elemento di nextCoord := ENDSTRING

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

LAVORO

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

ALGORITMO

i := 1

MENTRE (i-esimo elemento di startCell <> ENDSTRING)

i-esimo elemento di dest := i-esimo elemento di startCell

i := i + 1

FINE

i-esimo elemento di dest := COORD_SEPARATOR

j := i + 1

i := 1

MENTRE (i-esimo elemento di endCell <> ENDSTRING)

j-esimo elemento di dest := i-esimo elemento di endCell

i := i + 1

j := j + 1

FINE

j-esimo elemento di dest := ENDSTRING

pullColumn

INPUT

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

OUTPUT

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

LAVORO

Nome	Descrizione	Tipo	Vincoli
range	Range di coordinate della nave	Stringa	Formato: primaCella_ultimaCella

ALGORITMO

range := getCoords(ship)

integerColumn := getIntegerColumn(primo elemento di range)

pullRow

INPUT

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

Ship	Nave da cui estrarre il valore della riga	Ship	/
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

LAVORO

Nome	Descrizione	Tipo	Vincoli
range	Range di coordinate della nave	Stringa	Formato: primaCella_ultimaCella
row	Valore della riga	Stringa	Contiene un numero compreso tra 1 e 16
rowLen	Lunghezza della stringa row	Numero intero	Lunghezza compresa tra 1 e 2 (ENDSTRING escluso)
I	Indice di scorrimento per le stringhe	Numero intero	i > 0

ALGORITMO

```

i := START_ROW_COORD
range := getCoords(ship)
MENTRE(i-esimo elemento di range <> COORD_SEPARATOR)
    (i - 2)-esimo elemento di row := i-esimo elemento di range
    i := i + 1
FINE
(i - 2)-esimo elemento di row := ENDSTRING
rowLen := getLength(row)
integerRow := stringToNumber(row, rowLen)

```

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
isImpossibileToDo	Indica l'impossibilità di inserire una nave nel range di coordinate scelto	Numero intero	1 = Impossibile inserire nave 0 = La nave può essere inserita

ALGORITMO

StampaAVideo("IL TUO PLAYGROUND FIN ORA")

showMap(playground)

ESEGUI

column := getColumn()

row := getRow()

direction := getShipDirection()

cell := buildShipCoordinate(column, row)

nextCell := buildNextCoord(direction, cell, size)

coords := concatCoordinates(cell, nextCell)

isImpossibleToDo := isImpossible(cell, direction, coords, size, playground)

FINCHE (**isImpossibleToDo** = 1)

FINE

ship := setLabel(ship, (number + START_LOWERCASE_ASCII -1))

ship := setLifePoints(ship, size)

ship := setSize(ship, size)

ship := setCoords(ship, coords)

ship := setDirection(ship, direction)

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
playground	Mappa di gioco del 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

OUTPUT:

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

error	Valore numerico pari a 1 o 0 che rappresenta l'esito del controllo	Intero	1 = inserimento impossibile 0 = inserimento possibile
-------	--	--------	--

LAVORO:

Nome	Descrizione	Tipo	Vincoli
row	Riga della cella	intero	row >= TABLE_MIN AND row <= TABLE_MAX
column	Colonna della cella	intero	column >= TABLE_MIN AND column <= TABLE_MAX
Len	Lunghezza della stringa cella	Intero	len compreso tra 3 e 4 (carattere di ENDSTRING escluso)
checkMapBoundaries	Indica se si stanno violando i confini della mappa	Intero	1 = confini violati 0 = confini non violati
checkShipCollisions	Indica se si è in contatto con un'altra nave	Intero	1 = si è in contatto con un'altra nave 0 = non si è in contatto con un'altra nave

ALGORITMO:

error := 0

len := getLength(cell)

row := stringToNumber(cell, len)

column := getIntegerColumn(primo carattere di cell)

checkMapBoundaries := checkBoundaries(row, column, size, direction)

SE (checkMapBoundaries = 0)

ALLORA

checkShipCollisions := checkCollisions(playground, coords)

SE (checkShipCollisions = 1)

ALLORA

StampaAVideo("Coordinate non valide: sei in contatto con un'altra nave")

error := 1

FINE

ALTRIMENTI

StampaAVideo("Coordinate non valide: stai violando i confini della mappa")

error := 1

FINE

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

ALGORITMO:

error := 0

SE (direction = 'V')

ALLORA

SE((row - 1 + size) >= TABLE_MAX)

ALLORA

error := 1

FINE

ALTRIMENTI

SE(direction = 'O')

ALLORA

SE ((column - 1 + size) >= TABLE_MAX)

ALLORA

error := 1

FINE

FINE

FINE

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

LAVORO:

Nome	Descrizione	Tipo	Vincoli
firstCell	Prima cella della nave	Stringa	Formato: riga-colonna
lastCoord	Ultima cella della nave	Stringa	Formato: riga-colonna
pivotRow	Riga di inizio del controllo	Numero intero	pivotRow >= TABLE_MIN AND

			pivotRow <= TABLE_MAX
pivotColumn	Colonna di inizio del controllo	Numero intero	pivotColumn >= TABLE_MIN AND pivotColumn <= TABLE_MAX
endRow	Riga di fine del controllo	Numero intero	endRow >= TABLE_MIN AND endRow <= TABLE_MAX
endColumn	Colonna di fine del controllo	Numero intero	endColumn >= TABLE_MIN AND endColumn <= TABLE_MAX
i	Contatore delle colonne	Numero intero	i > 0

ALGORITMO:

error := 0

firstCell := getFirstCell(coords, firstCell)

firstCellLength := getLength(firstCell)

lastCoord := getLastCell(coords, lastCoord)

lastCoordLength := getLength(lastCoord)

pivotColumn := getIntegerColumn(**primo Elemento di firstCell**) - 1

pivotRow := stringToNumber(firstCell, firstCellLength) - 1

SE (pivotColumn < TABLE_MIN)

ALLORA

pivotColumn := TABLE_MIN

FINE

SE (pivotRow < TABLE_MIN)

ALLORA

pivotRow := TABLE_MIN

FINE

endColumn := getIntegerColumn(**primo element di lastCoord**) + 1

endRow := stringToNumber(lastCoord, lastCoordLength) + 1

SE (endColumn > TABLE_MAX)

ALLORA

endColumn := TABLE_MAX

FINE

SE (endRow > TABLE_MAX)

ALLORA

endRow := TABLE_MAX

FINE

MENTRE (pivotRow <= endRow AND error = 0)

 i := pivotColumn

 MENTRE (i <= endColumn AND error = 0)

 SE (elemento in riga pivotRow e colonna i di playground <> WATER)

 ALLORA

 SE ((elemento in riga pivotRow e colonna i di playground >= 'a') AND
 (elemento in riga pivotRow e colonna i di playground >= 'p'))

 ALLORA

 error := 1

 FINE

 FINE

 i := i + 1

 FINE

 pivotRow := pivotRow + 1

FINE

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

LAVORO:

Nome	Descrizione	Tipo	Vincoli
i	Contatore per le posizioni delle stringhe	intero	i > 0

ALGORITMO:

i := 1

MENTRE (carattere in posizione i di coords <> COORD_SEPARATOR)

 carattere in posizione i di cell := carattere in posizione i di coords

 i := i + 1

FINE

carattere in posizione i di cell := ENDSTRING

getLastCell**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	Ultima cella della nave	Stringa	Formato: riga-colonna

LAVORO:

Nome	Descrizione	Tipo	Vincoli
i	Contatore per le posizioni delle stringhe	intero	i > 0
j	Contatore per le posizioni delle stringhe	intero	j > 0

ALGORITMO:

i := 1

j := 1

MENTRE (carattere in posizione i di coords <> COORD_SEPARATOR)

 i := i + 1

FINE

i := i + 1

MENTRE (carattere in posizione i di coords <> ENDSTRING)

 carattere in posizione j di cell := carattere in posizione i di coords

 j := j + 1

 i := i + 1

FINE

carattere in posizione j di cell := ENDSTRING
