Sommario Pseudocodice

Torna al sommario

Torna al sommario

MODULO: gameLoop

INTERFACCIA:

- lanciareColpo()
- laciareColpoClassico()
- laciareBombardamentoAereo()
- lanciareRadar()
- lanciareLargoRaggio()
- switchGiocatori()
- scrivereColpo()
- aumentareCoordinate()
- calcolareScansione()
- leggereTipoColpo()
- leggereLinea()
- finePartita()

CORPO:

- gameLoop()

TUTTE LE FUNZIONI:

gameLoop()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita	DatiPartita	//
turniTotali	Turni totali dellapartita	Intero	> 0
tipoColpo	Valore che indica iltipo di colpo che l'utente vuole lanciare	Intero	1 <= tipoColpo <= 4
inizioTurno	Valore che l'utente sceglie di inserire quando si trova nelmenu' di inizio turno	Intero	0 <= inizioTurno <= 1
fineTurno	Valore che l'utente sceglie di inserire quando si trova nel menu' di fine turno	Intero	0 <= fineTurno <= 2
BACK	Valore che identifica il ritorno al menu principale	Intero	Variabile globale
ESCI	Valore che identifica l'uscita dal gioco	Intero	Variabile globale

Torna al sommario

SALVARE	Valore che identifica il salvataggio della partita e l'uscita	Intero	Variabile globale
endGame	Valore di controllo che indica lo stato della partita	Intero	CONTINUA, BACK, SALVARE, ESCI

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura che contiene i dati della partita con i campi modificati	DatiPartita	//

ALGORITMO

```
endGame = 0
ESEGUI
    turnITotali = leggereTurnitotali (struttura)
    inizioTurno = confermareInizioTurno (leggereDatiGiocatore1 (struttura))
    SE (inizioTurno = 0)
        ALLORA
        struttura = scrivereEndGame (struttura, BACK)
        endGame = 1
        ALTRIMENTI
            stampareInizioTurno (leggereDatiGiocatore1 (struttura), turniTotali)
            tipoColpo = leggereTipoColpo (leggereDatiGiocatore1 (struttura),
            turniTotali)
            SE (tipoColpo = 1)
                ALLORA
                struttura = lanciareColpoClassico (struttura)
                ALTRIMENTI SE (tipoColpo = 2)
                    ALLORA
                    struttura = lanciareLargoRaggio (struttura)
                FINE
                ALTRIMENTI SE (tipoColpo = 3)
```

Torna al sommario

ALLORA

struttura = lanciareRadar (struttura)

FINE

ALTRIMENTI

struttura = lanciareBombardamentoAereo (struttura)

FINE

struttura = switchGiocatori (struttura)

fineTurno = confermareFineTurno (leggereDatiGiocatore1 (struttura))

SE (fineTurno = 0)

ALLORA

struttura = scrivereEndGame (struttura, BACK)

endGame = 1

ALTRIMENTI SE (fineTurno = 1)

ALLORA

endGame = finePartita (struttura)

SE (endGame = 1)

ALLORA

struttura = scrivereEdnGame (struttura, ESCI)

FINE

FINE

ALTRIMENTI

struttura = scrivereEndGame (struttura, SALVARE)

endGame = 1

FINE

FINE

FINE

FINCHE' (endGame = 0)

Torna al sommario

lanciareColpo()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Colpo	Coordinate del colpo da lanciare	Coordinate	//
struttura	Struttura che contiene i dati della partita	DatiPartita	//
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
Avversario	Struttura che contiene i dati del giocatore in attesa del turno	Player	//
esitoPresenaNave	esito della verifica Nave affondata	Intero	0 <= esitoRpesenzaNave <= 1
esitoNaveAffondata	esito della verifica Nave affondata	Intero	0 <= esitoNAveAffondata <= 1
esitoPresenza	Esito della verifica presenza elemento	Intero	0 <= esitoPresenza <= 1
ESITO_COLPO_OK	id esito colpo riuscito	Intero	Variabile globale
ESITO_COLPO_ERR	id colpo già lanciato in coordinate colpo	Intero	Variabile globale
SEGNA_COLPO	id scrivere colpo inmappa giocatore	Intero	Variabile globale
SEGNA_NAVE	id scrivere colpo inmappa avversaria	Intero	Variabile globale
SEGNA_COLPO_MANCATO	id scrivere colpo mancato in mappagiocatore	Intero	Variabile globale
AFFONDATO	id nave affondata	Carattere	Variabile globale
COLPITO	id nave colpita	Carattere	Variabile globale

Torna al sommario

ACQUA	id acqua in coordinatecolpo	Carattere	Variabile globale
SAS	id errore “Colpo giàlanciato”	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita con i campi modificati	DatiPartita	//

ALGORITMO

giocatore = leggereDatiGiocatore1 (struttura)

avversario = leggereDatiGiocatore2 (struttura)

struttura = scrivereEsitoColpi (struttura, ESITO_COLPO_OK)

esitoPresenzaNave = verificaPresenzaNave (avversario, colpo)

SE (esitoPresenzaNave = 1)

ALLORA

giocatore = scrivereColpo (giocatore, colpo, SEGNA_COLPO)

avversario = scrivere Colpo (avversario, colpo, SEGNA_NAVE)

esitoNaveAffondata = verificareNaveAffondata (avversario, colpo)

SE (esitoNaveAffondata = 1)

ALLORA
giocatore = incrementareNaviAffondate (giocatore)

stampareEsitoColpo (colpo, AFFONDATO)

ALTRIMENTI

stampareEsitoColpo (colpo, COLPITO)

FINE

ALTRIMENTI

esitoPresenza = verificarePresenza (avversario, colpo, ACQUA)

SE (esitoPresenza = 1)

ALLORA

Torna al sommario

giocatore = scrivereColpo (giocatore,colpo,SEGNA_COLPO_MANCATO)

avversario = scrivereColpo (avversario,colpo,SEGNA_NAVE_MANCATO)

stampareEsitoColpo (colpo, ACQUA)

ALTRIMENTI

stampareERRORE (SAS) //colpo già lanciato

struttura = scrivereEsitoColpi (struttura,ESITO_COLPO_ERR)

FINE

FINE

struttura = scrivereDatiGiocatore1 (struttura, giocatore)

struttura = scrivereDatiGiocatore2 (struttura, avversario)

lanciareColpoClassico()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Colpo	Coordinate del colpoda lanciare	Coordinate	//
struttura	Struttura che contiene i dati dellapartita	datiPartita	//
esitoColpo	Esito del colpo lanciato	intero	ESITO_COLPO_OK, ESITO_COLPO_ERR
ESITO_COLPO_OK	Id esito del colpo riuscito	intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita con i campi modificati	datiPartita	//

ALGORITMO

ESEGUI

colpo = ottenereCoordinate (colpo)

struttura = lanciareColpo (struttura, colpo)

esitoColpo = leggereEsitoColpi (struttura)

FINCHE' (esitoColpo <> ESITO_COLPO_OK)

Torna al sommario

lanciareBombardamentoAereo()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita	datiPartita	//
colpo	Coordinate del colpo da lanciare	coordinate	//
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
linea	Orientamento bombardamento	intero	0 <= linea <= 15
direzione	Direzione del colpo	intero	VERTICALE <= direzione <= ORIZZONTALE
esitoColpo	Esito del colpo lanciato	intero	ESITO_COLPO_OK, ESITO_COLPO_ERR
SRC_DIREZIONE	Percorso del file menu direzione	vettore di caratteri	Variabile globale
ESITO_COLPO_ERR	Id colpo già lanciato in coordinate colpo	intero	Variabile globale
RIGHE	Numero di righe della mappaNavi	intero	Variabile globale
ECS	id dell'errore "tutte le coordinate sono già state colpite"	intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
errColpo	variabile che indica il numero di colpi lanciati dal giocatore	intero	0 <= errColpo <= 9
colpiLanciati	variabile contatore che indica i colpi che sono stati lanciati dal bombardamento aereo	intero	0 <= colpiLanciati <= 9

Torna al sommario

k	variabile contatore per il lancio dei colpi	intero	0 <= k <= RIGHE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita con i campi modificati	datiPartita	//

ALGORITMO

```
errColpo = 0
colpilanciati = 0
ESEGUI
        stampareFile (SRC_DIREZIONE)
        direzione = leggereRispostaValida (1, 2)
        linea = leggereLinea (direzione)
        k = 0
        ESEGUI
                colpo = aumentareCoordinate (colpo, direzione, linea, k)
                struttura = lanciareColpo(struttura, colpo)
                colpiLanciati = colpiLanciati +1
                esitoColpo = leggereEsitoColpo(struttura)
                SE (esitoColpo = ESITO_COLPO_ERR)
                        ALLORA
                        errColpo = errColpo +1
                FINE
                k = k + 1
        FINCHE' (k < RIGHE)
        SE (errColpo = colpilanciati)
                ALLORA
                stampareErrore (ECS)                    //tutte le coordinate sono già state colpite
        FINE
FINCHE' (errColpo = colpiLanciati)
```

Torna al sommario

giocatore = leggereDatiGiocatore1 (struttura)

giocatore = incrementareBombardamentoAereo (giocatore)

struttura = scrivereDatGiocatore1 (struttura, giocatore)

lanciareRadar()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita	datiPartita	//
giocatore	Struttura che contiene i dati del giocatore corrente	player	//
avversario	Struttura che contiene i dati del giocatore in attesa del turno	palyer	//
posizioniScansionate	vettore di posizioni calcolate per il radar	vettore di coordinate	Dim CELLE_SCANSIONARE
esitoPresenzaNave	esito del controllo PresenzaNave	intero	0 <= esitoPresenzaNave <= 1
CELLE_RADAR	Valore che indica l'area o il numero di celle massimo che il radar può scansionare	intero	Variabile globale
SEGNA_RADAR_NAVE	id scrivi nave trovata	intero	Variabile globale
SEGNA_RADAR_VUOTO	id scrivi no nave trovata	intero	Variabile globale
RADAR_NAVE	Id esito del colpo radar quando viene trovata una nave	Carattere	Variabile globale
RADAR_VUOTO	Id esito del colpo radar quando non trova una nave	Carattere	Variabile globale

Torna al sommario

ACQUA	id acqua di MappaNavi	Carattere	Variabile globale
EAL	id errore “Tutte le coordinate sono state già scansionate”	intero	Variabile globale
esitoVerificaScansione	Esito del controllo verificaScansione	Intero	0 <= esitoVerificaScansione <= 1
esitoPresenza	Esito del controllo presenza elemento	Intero	0 <= esitoPresenza <= 1

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
errColpo	Contatore che viene incrementato ogni volta che una cella del radar non contiene acqua o una nave	intero	0 <= errColpo <= 9
colpiLanciati	Contatore che viene incrementato ogni volta che viene scansionata una cella	intero	0 <= colpiLanciati <= 9
k	variabile contatore che cicla le posizioni di posizioniScansionate	intero	0 <= k <= CELLE_RADAR

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita con i campi modificati	datiPartita	//

ALGORITMO

giocatore = leggereDAtiGiocatore1(struttura)

avversario = leggereDatiGiocatore2(struttura)

errColpo = 0

Torna al sommario

```
colpiLanciati = 0

ESEGUI
        colpo = ottenereCoordinate (colpo)

        posizioniScansionate = inizializzarePosScansione (posizioneScansionate)

        posizioniScansionate = calcolareScansione (colpo, posizioniScansionate)

        k = 0

        MENTRE (k < CELLE_RADAR)        //=9

                esitoVerificaScansione = verificareScansione (elemento in posizione k di
                posizioniScansionate)

                SE (esitoVerificaScansione = 1)

                        ALLORA

                        colpiLanciati = colpiLanciati + 1
                        esitoPresenzaNave = verificarePresenzaNave (avversario, elemento in posizione k
                        di posizioniScansionate)

                        SE (esitoPresenzaNave = 1)

                                ALLORA

                                giocatore = scrivereColpo (giocatore, elemento in posizione k di
                                posizioniScansionate,SEGNA_RADAR_NAVE)

                                stampareEsitoColpo (elemento in posizione k di posizoniScansionate,
                                RADAR_NAVE)

                                ALTRIMENTI

                                        esitoPresenza = verificarePresenza (avversario,
                                        elemento in posizione k di posizioniScansionate, ACQUA)

                                        SE (esitoPresenza = 1)

                                                ALLORA
                                                giocatore = scrivereColpo (giocatore, elemento in posizione
                                                k diposizioniScansionate, SEGNA_RADAR_VUOTO)


                                                stampareEsitoColpo (elemento in posizione k di
                                                posizioniScansionate, RADAR_VUOTO)
                                        ALTRIMENTI

                                                errColpo = errColpo +1
                                FINE
                        FINE

                FINE

                k = k + 1
        FINE
```

Torna al sommario

SE (errColpo = colpiLanciati)

ALLORA

stampaErrore (EAL) //tutte le coordinate sono già state scansionate

FINE

FINCHE' (errColpo = colpiLanciati)

giocatore = incrementareRadar (giocatore)

struttura = scrivereDatiGiocatore1 (struttura, giocatore)

struttura = scrivereDatiGiocatore2 (struttura, avversario)

lanciareLargoRaggio()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Colpo	Coordinate del singolo colpo da lanciare	Coordinate	//
struttura	Struttura che contiene i dati della partita	DatiPartita	//
posizioniScansionate	vettore di posizioni calcolate per il largoRaggio	Vettore di coordinate	Dim CELLE_LARGO_RAGGIO
CELLE_LARGO_RAGGIO	Valore che indica l'area o numero di caselle massimo che il largo raggio può colpire	Intero	Variabile globale
Giocatore	struttura dati del giocatore che ha lanciato il colpo a largo raggio	Player	//
ESITO_COLPO_ERR	Valore che indica che una delle caselle colpite da largo raggio è già stata precedentemente colpita	Intero	Variabile globale

Torna al sommario

ECS	id dell'errore "Tutte le coordinate scelte sono state già colpite"	Intero	Variabile globale
esitoColpo	Esito del colpo lanciato	Intero	ESITO_COLPO_OK <= esitoColpo <= ESITO_COLPO_ERR
esitoVerificaScansione	Indica se il colpo può essere lanciato o meno in quella posizione	Intero	0 <= esitoVerificaScansione <= 1

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
k	Contatore che cicla gli elementi di posizioniScansionate	Intero	0 <= k <= CELLE_LARGO_RAGGIO

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita con i campi modificati	DatiPartita	//

ALGORITMO

errColpo = 0

colpiLanciati = 0

ESEGUI

colpo = ottenereCoordinate (colpo)

posizioniScansionate = inizializzaPosScansione (posizioniScansionate)

posizioniScansionate = calcolareScansione (colpo, posizioniScansionate)

k = 0

MENTRE (k < CELLE_LARGO_RAGGIO). //=9

esitoVerificaScansione = verificareScansione (elemento in posizione k di posizioni Scansionate)

SE (esitoVerificaScansione = 1)

Torna al sommario

ALLORA

struttura = lanciareColpo (struttura, elemento in posizione k di posizioniScansionate)

colpiLanciati = colpiLanciati + 1

esitoColpo = leggereEsitoColpi (struttura)

SE (esitoColpo = ESITO_COLPO_ERR) //=2

ALLORA

errColpo = errColpo +1

FINE

FINE

k = k + 1

FINE

giocatore = leggereDatiGiocatore1 (struttura)

SE (errColpo = colpiLanciati)

ALLORA

stampaErrore (ECS). //tutte le coordinate sono già state colpite

FINE

FINCHE' (errColpo = colpiLanciati)

giocatore = incrementareLargoRaggio (giocatore)

struttura = scrivereDatiGiocatore1 (struttura, giocatore)

switchGiocatori()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita	DatiPartita	//
giocatore	Struttura che contiene i dati del giocatore corrente	Player	//

Torna al sommario

avversario	Struttura che contiene i dati del giocatore in attesa del turno	Player	//
turnoPartita	Valore che indica il turno del giocatore 1 o giocatore 2	Intero	PLAYER_UNO <= turnoPartita <= PLAYER_DUE
PLAYER_UNO	turno del giocatore 1	Intero	Variabile globale
PLAYER_DUE	turno del giocatore 2	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita con i campi modificati	DatiPartita	//

ALGORITMO

```
giocatore = leggereDatiGiocatore1 (struttura)
avversario = leggereDatiGiocatore2 (struttura)
struttura = scrivereDatiGiocatore1 (struttura, avversario)
struttura = scrivereDatiGiocatore2 (struttura, giocatore)
turnoPartita = leggereTurnoPartita(struttura)
SE (turnoPartita = PLAYER_UNO)                    //=1
        ALLORA
        struttura = scrivereTurnoPartita (struttura, PLAYER_DUE)
        ALTRIMENTI
                struttura = scrivereTurnoPartita (struttura, PLAYER_UNO)
                struttura = incrementareTurniTotali (struttura)
FINE
```

Torna al sommario

scrivereColpo()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
giocatore	Struttura dati che contiene i dati del giocatore	Player	//
posizione	Coordinate x, y in cui scrivere il colpo	Coordinate	//
tipologia	Tipo di colpo da scrivere nella mappa del giocatore	Intero	//
SEGNA_COLPO	Id segna COLPITO in mappaColpi giocatore	Intero	Variabile globale
SEGNA_NAVE_MANCATO	Id segna MANCATO in mappaNavi giocatore	Intero	Variabile globale
SEGNA_RADAR_NAVE	Id segna RADAR_NAVE in mappaColpi giocatore	Intero	Variabile globale
SEGNA_RADAR_VUOTO	Id segna RADAR_VUOTO in mappaColpi giocatore	Intero	Variabile globale
COLPITO	Id colpito	Carattere	Variabile globale
MANCATO	Id colpo a vuoto	Carattere	Variabile globale
RADAR_NAVE	Id presenza nave durante scansione radar	Carattere	Variabile globale
RADAR_VUOTO	Id nessuna presenza nave durante scansione radar	Carattere	Variabile globale
SEGNA_NAVE	Id segna COLPITO in mappaNavi giocatore	Intero	Variabile Globale
SEGNA_COLPO_MANCATO	Id segna mancato in	Intero	Variabile globale

Torna al sommario

	mappaColpi di giocatore		

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
giocatore	Struttura dati che contiene i dati del giocatore con i campi modificati	Player	//

ALGORITMO

SE (tipologia = SEGNA_COLPO)

ALLORA

giocatore = scrivereMappaColpi (giocatori, posizione, COLPITO)

ALTRIMENTI SE (tipologia = SEGNA_NAVE)

ALLORA

giocatore = scrivereMappaNavi (giocatore, posizione, COLPITO)

FINE

ALTRIMENTI SE (tipologia = SEGNA_COLPO_MANCATO)

ALLORA

giocatore = scrivereMappaColpi (giocatore, posizione, MANCATO)

FINE

ALTRIMENTI SE (tipologia = SEGNA_NAVE_MANCATO)

ALLORA

giocatore = scrivereMappaNavi (giocatore, posizione, MANCATO)

FINE

ALTRIMENTI SE (tipologia = SEGNA_RADAR_NAVE)

ALLORA

giocatore = scrivereMappaColpi (giocatore, posizione, RADAR_NAVE)

FINE

ALTRIMENTI SE (tipologia = SEGNA_RADAR_VUOTO)

ALLORA

giocatore = scrivereMappaColpi (giocatore, posizione, RADAR_VUOTO)

Torna al sommario

FINE

FINE

aumentareCoordinate()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Colpo	Coordinate del colpo in cui scrivere linea e valore	Coordinate	//
Direzione	Valore che indica la direzione in cui lanciare il colpo, VERTICALE, ORIZZONTALE	Intero	VERTICALE <= Direzione <= ORIZZONTALE
VERTICALE	Valore che indica l'orientamento verticale	intero	Variabile globale
Valore	Valore da incrementare per variare la posizione all'interno delle righe o colonne della mappa	intero	0 <= valore <= 15
Linea	Linea scelta dall'utente	Intero	0 <= linea <= 15

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Colpo	Coordinate del colpo aumentate	Coordinate	//

ALGORITMO

SE (direzione = VERTICALE)

ALLORA

colpo = scrivereCoordinateX (colpo, valore)

colpo =scrivereCoordinateY(colpo, linea)

ALTRIMENTI

colpo = scrivereCoordinateX(colpo, linea)

colpo = scrivereCoordinateY (colpo, valore)

Torna al sommario

FINE

calcolareScansione()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Colpo	Contiene le coordinate x, y del colpo	Coordinate	//
Posizioni	Coordinate delle nove posizioni da calcolare a partire dalla coordinata colpo	Vettore di coordinate	Dim CELLE_LARGO_RAGGIO
Righe	Valore del campo x di colpo	Intero	0 <= righe <= 15
Colonne	Valore del campo y di colpo	Intero	0 <= colonne <= 15
AREA_LARGO_RAGGIO	Valore che indica l'area o numero di caselle che	Intero	Variabile globale
x	coordinate x di colpo	intero	0 <= x <= 15
y	coordinate y di colpo	intero	0<=x<=15

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
Conta	Contatore del vettore posizioni	Intero	//
i	Variabile contatore che cicla l'area del largo raggio	intero	0 <= i <= AREA_LARGO_RAGGIO
J	Variabile contatore che cicla l'area del largo raggio	intero	0 <= j <= AREA_LARGO_RAGGIO
x	Contiene il valore del campo X di Colpo	Intero	1 <= x <= 16
y	Contiene il valore del campo Y di Colpo	intero	1 <= y <= 16

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizioni	Coordinate delle nove posizioni calcolate a partire dalla coordinata colpo	Vettore di coordinate	Dim CELLE_LARGO_RAGGIO

ALGORITMO

```
Conta = 0
i = 0
MENTRE (i < AREA_LARGO_RAGGIO)      //=3
        j = 0
        MENTRE (j < AREA_LARGO_RAGGIO)
                x = leggereCoordinateX (colpo)
                y = leggereCoordinateY (colpo)
                elemento in posizione conta di posizioni = scrivereCoordinateX (elemento in posizione
                conta di posizioni, (x - 1) + i)
                elemento in posizione conta di posizioni = scrivereCoordinateY (elemento in posizione
                conta di posizioni, (y - 1) + i)
                conta = conta + 1
                j = j + 1
        FINE
        i = i + 1
FINE
```

leggereTipoColpo()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
giocatore	Struttura dati checontiene i dati del giocatore	Player	//
turno	Indica il turno delgiocatore	Intero	PLAYER_UNO <= turno <=

Torna al sommario

			PLAYER_DUE
MAX_LARGO_RAGGIO	Numero massimo di colpi a largoraggio che l'utente può lanciare	Intero	Variabile globale
MAX_RADAR	Numero massimodi colpi radar che l'utente può lanciare	Intero	Variabile globale
MAX_BOMBARDAMENTO	Numero massimodi colpo bombardamento aereo che l'utente può lanciare	Intero	Variabile globale
TURNO_BOMBARDAMENTO	Valore che indicail turno dopo il quale l'utente può lanciare il bombardamentoaereo	Intero	Variabile globale
SNA	Id dell'errore "Non puoi usare questo tipo di colpo"	Intero	Variabile globale
Scelta	Scelta utente del colpo	Intero	1 <= scelta <= 4
C_largoRaggio	Numero di colpi largoRaggio che l'utente può lanciare	Intero	0 <= c_largoRaggio <= 3
C_bombardamento	Numero di colpi bombardamento che l'utente può lanciare	Intero	0 <= c_bombardamento <= 1
C_radar	Numero di colpi radar che l'utente può lanciare	Intero	0 <= c_radar <= 3

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
errore	Indica se è possibile lanciare o meno un colpo speciale	Intero	0 <= errore <= 1

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
scelta	Tipo di colpo scelto dall'utente	Intero	1 <= scelta <= 4

ALGORITMO

c_largoRaggio = MAX_LARGO_RAGGIO – leggereLargoRaggio (giocatore)

c_bombardamento = MAX_BOMBARDAMENTO –leggereBombardamentoAereo(giocatore)

c_radar = MAX_RADAR – leggereRadar (giocatore)

Torna al sommario

ESEGUI

stampareAvideo (“Selezione tipologia di colpo”)

stampareAvideo (“1) Colpo classico”)

tampareAvideo (“2) Colpo a largo raggio”, c_largoraggio)

stampareAvideo (“3) Radar”, c_radar)

SE (turno > TURNO_BOMBARDAMENTO)

ALLORA

stampareAvideo (“4) Bombardamento aereo”, c_bombardamento)

FINE

stampareAvideo (“Scelta: ”)

scelta = leggereRispostaValida (1, 4)

SE (scelta = 1)

ALLORA

errore = 0

ALTRIMENTI SE(scelta = 2 **AND** c_largoRaggio > 0)

ALLORA

errore = 0

FINE

ALTRIMENTI SE (scelta = 3 **AND** c_radar > 0)

ALLORA

errore = 0

FINE

ALTRIMENTI SE (scelta = 4 **AND** c_bombardamento > 0 **AND** turno > TURNO_BOMBARDAMENTO)

ALLORA

errore= 0

FINE

ALTRIMENTI

errore = 1

FINE

SE (errore = 1)

Torna al sommario

ALLORA

stampaErrore (SN) // non puoi usare questo tipo di colpo

FINE

FINCHE' (errore <> 0)

LeggereLinea()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
direzione	Valore che indica se la direzione indicata è orizzontale o verticale	Intero	1 <= direzione <= 2
lettura	Valore della colonna che l'utente sceglie in caso di direzione verticale	Carattere	A <= lettura <= P
VERTICALE	Id orientamento verticale	Intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
errore	Indica se la lettera che indentifica la colonna è stata inserita correttamente	Intero	0 <= errore <= 1

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
linea	Numero di riga o colonna letta	intero	0 <= linea <= 15

ALGORITMO

linea = 0

SE (direzione = VERTICALE)

ALLORA

Torna al sommario

```
ESEGUI
        errore = 0
        stampareAvideo (“Inserisci lettera colonna: ”)
        lettura = leggereDaTastiera( )
        SE (lettura >= A AND lettura <= P)
                ALLORA
                linea = (lettura – 64) - 1
                ALTRIMENTI SE (lettura >= a AND lettura <= p)
                        ALLORA
                        linea = (lettura – 96) - 1
                FINE
                ALTRIMENTI
                        errore = 1
        FINE
FINCHE’ (errore = 1)
ALTRIMENTI
stampareAVideo (“Inserisci numero riga: ”)
linea = leggereRispostaValida (1, 16)
linea = linea – 1
FINE
```

Torna al sommario

finePartita()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
struttura	Struttura che contiene i dati della partita	datiPartita	//
Giocatore	struttura dati del giocatore 1	player	//
avversario	dati del giocatore 2 di struttura	player	//
naviAffondate	Valore che indica le navi in quel momento affondate dal giocatore	intero	0 <= naviAffondate <= 15
TOTALE_NAVI	numero totale di navi possibili	intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
finePartita	esito della verifica finePartita	intero	0 <= finePartita <= 1

ALGORITMO

finePartita = 1

giocatore = leggereDatiGiocatore1 (struttura)

avversario = leggereDatiGiocatore2 (struttura)

naviAffondate = leggereNaviAffondate (giocatore)

SE (naviAffondate >= TOTALE_NAVI)

ALLORA

StampareAVideo (“HAI VINTO: ”, leggereNomePlayer (giocatore))

ALTRIMENTI

naviAffondate = leggereNaviAffondate (avversario)

SE (leggereNaviAffondate >= TOTALE_NAVI)

Torna al sommario

ALLORA

StampareAVideo (“HAI VINTO: “, leggereNomePlayer (avversario))

ALTRIMENTI

finePartita = 0

FINE

FINE

Torna al sommario

MODULO: globali

INTERFACCIA:

- leggereRispostaValida()
- ottenereCoordinate()
- convertiPosizioneInLettera()

leggereRispostaValida()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
min	Valore minimo ammesso come risposta valida	Intero	//
max	Valore massimo ammesso come risposta valida	Intero	//
ANC	Id errore "risposta non corretta"	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Risposta	Dato letto da tastiera che rappresenta la scelta dell'utente	Intero	min <= risposta <= max

ALGORITMO

ESEGUI

risposta = leggereDaTastiera( )

SE (risposta < min OR risposta > max)

ALLORA

stampareErrore (ANC) //risposta non corretta

FINE

FINCHE' (risposta < min OR risposta > max)

Torna al sommario

ottenereCoordinate()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura usata per contenere i dati del colpo da lanciare	Coordinate	//
stringaCoordinate	Valore delle coordinate che il giocatore sceglie, lette da tastiera	Vettore di caratteri	Dim = MAX_DIM_STRINGA_POS
isValid	Esito della verifica della correttezza di stringaCoordinate	Intero	0 <= isValid <= 1
MAX_DIM_STRINGA_POS	Dimensione massima della stringa posizione	Intero	Variabile globale
SOM	Id errore “colpo fuori mappa”	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura che contiene le coordinate del colpo in una posizione valida che si vuole lanciare	Coordinate	//

ALGORITMO

ESEGUI

stampareAVideo (“Inserisci coordinate nel formato An [es: C5]”)

stringaCoordinate = leggereDaTastiera ( )

posizione = verificareCoordinate (stringaCoordinate, posizione)

isValid = leggereCoordinateIsValid (posizione)

SE (isValid = 0)

ALLORA

stampareErrore (SOM) // colpo fuori mappa

FINE

FINCHE’ (isValid = 0)

Torna al sommario

convertiPosizioneInLettera()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
num	Valore intero da convertire in lettera	Intero	0 <= num <= 15

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Lettera	Valore convertito in carattere	Carattere	A <= lettera <= P

ALGORITMO

lettera = (num +64)+1

Torna al sommario

MODULO: verifiche

INTERFACCIA:

- verificareSingolaAffondata()
- verificareAffondataVerticale()
- verificareAffondataOrizzontale()
- verificareDirezioneNave()

CORPO:

- verificarePresenzaNave()
- verificareNave()
- verificareNaveAffondata()
- verificareScansione()
- verificarePresenza()
- verificareCoordinate()

TUTTE LE FUNZIONI:

verificarePresenzaNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
Posizione	Struttura che contiene le coordinate in cui verificare la presenza della nave	Coordinate	//
ID_PORTAEREI	Id della nave di len5	carattere	Variabile globale
ID_CORAZZATA	Id della nave di len4	carattere	Variabile globale
ID_INCROCIATORE	Id della nave di len3	carattere	Variabile globale
ID_CACCIATORPEDINIERE	Id della nave di len2	carattere	Variabile globale
ID_NAVESUPPORTO	Id della nave di len1	carattere	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Presenza	Esito della verifica presenzaNave 0 = non è presente 1 = presente	Intero	0 <= presenza <= 1

ALGORITMO

SE (verificarePresenza (giocatore, posizione, ID_PORTAEREI)) //=A

ALLORA

Torna al sommario

presenza = 1

ALTRIMENTI SE ((verificarePresenza (giocatore, posizione, ID_CORAZZATA)) //=B

ALLORA

presenza = 1

FINE

ALTRIMENTI SE ((verificarePresenza (giocatore, posizione, ID_INCROCIATORE))//=C

ALLORA

presenza = 1

FINE

ALTRIMENTI SE ((verificarePresenza (giocatore, posizione, ID_CACCIATORPEDINIERE))//=D

ALLORA

presenza = 1

FINE

ALTRIMENTI SE ((verificarePresenza (giocatore, posizione, ID_NAVESUPPORTO))//=E

ALLORA

presenza = 1

FINE

ALTRIMENTI

presenza = 0

FINE

verificareNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
Posizione	Struttura che contiene le coordinate della nave da posizionare	Coordinate	//
Direzione	Indica la direzione della nave da posizionare 1= VERTICALE 2= ORIZZONTALE	Intero	1 <= direzione <= 2

Torna al sommario

lunghezzaNave	Lunghezza della nave da posizionare	Intero	1 <= lunghezzaNave <= 5
Lettura	Copia di posizione su cui effettuare calcoli	Coordinate	= posizione
esitoLetturaX	Variabile che contiene il valore di x di posizione	Intero	0 <= esitoLetturaX <= RIGHE
esitoLetturaY	Variabile che contiene il valore di y di posizione	Intero	0 <= esitoLetturaY <= COLONNE
esitoPresenza	Esito della verifica: verificaPresenza	Intero	0 <= esitoPresenza <= 1
DISTANZA_NAVI	Valore della distanza tra le navi	Intero	Variabile globale
ACQUA	Id Acqua delle Mappe	Intero	Variabile globale
RIGHE	Valore che corrisponde al numero di righe della Mappa	Intero	Variabile globale
COLONNE	Valore che corrisponde al numero di colonne della Mappa	Intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
i	Contatore posizione righe	Intero	0 <= i <= r
j	Contatore posizione colonne	Intero	0 <= j <= c
r	Valore di supporto per il controllo del posizionamento della nave	Intero	= DISTANZA_NAVI = lunghezza nave + 2
c	Valore di supporto per il controllo del posizionamento della nave	Intero	= DISTANZA_NAVI = lunghezza nave + 2

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Errore	Esito della verifica verificareNave	Intero	0 <= errore <= 2

Torna al sommario

ALGORITMO

Lettura = posizione

errore = 0

i = 0

lettura = decrementareCoordinateX (lettura)

lettura = decrementareCoordinateY (lettura)

SE (direzione = 1)

ALLORA

r = lunghezzaNave+2

c = DISTANZA_NAVI

ALTRIMENTI

r = DISTANZA_NAVI

c = lunghezzaNave+2

FINE

MENTRE (i < r **AND** errore = 0)

j = 0

MENTRE (j < c **AND** errore = 0)

esitoLetturaX = leggereCoordinateX (lettura)

esitoLetturaY = leggereCoordinateY (lettura)

SE ((esitoLetturaX >= 0 **AND** esitoLetturaX < RIGHE) **AND** (esitoLetturaY >=0 **AND** esitoLetturaY < COLONNE))

ALLORA

esitoPresenza = verificarePresenza (giocatore, lettura, ACQUA)

SE (esitoPresenza = 0)

ALLORA

errore = 2 //distanza nave non rispettata

FINE

ALTRIMENTI SE ((esitoLetturaX < -1 **OR** esitoLetturaX > RIGHE) **OR** (esitoLetturaY < -1 **OR** esitoLetturaY > COLONNE))

ALLORA

errore = 1. //nave non rientra nella mappa

FINE

Torna al sommario

FINE

lettura = IncrementareCoordinateY (lettura)

j = j + 1

FINE

lettura = ScrivereCoordinateY (lettura, leggereCoordinateY (posizione)-1)

lettura = IncrementareCoordinateX (lettura)

i = i + 1

FINE

VerificareNaveAffondata()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
Colpo	Struttura che contiene le coordinate della nave colpita	Coordinate	//
Direzione	Valore che indica la direzione della nave -1 = nave singola 1 = VERTICALE 2 = ORIZZONTALE	Intero	= -1, = 1, =2
VERTICALE	Valore che corrisponde alla posizione verticale	Intero	Variabile globale
ORIZZONTALE	Valore che corrisponde alla posizione orizzontale	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Affondato	Esito della verifica naveAffondata 0 = non affondata 1 = affondata	Intero	0 <= affondato <= 1

Torna al sommario

ALGORITMO

affondato = 1

direzione = verificareDirezioneNave (giocatore, colpo)

SE (direzione = VERTICALE)

ALLORA

affondato = verificareAffondatoVerticale (giocatore, colpo)

ALTRIMENTI SE (direzione = ORIZZONTALE)

ALLORA

affondato = verificareAffondatoOrizzontale (giocatore, colpo)

FINE

ALTRIMENTI

affondato = verificareSingolaAffondata (giocatore, colpo)

FINE

verificareScansione()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura che contiene le coordinate inserite da scansionare	Coordinate	//
r	Numero di riga su cui effettuare la scansione	Intero	0 <= r < RIGHE
c	Numero di colonna su cui effettuare la scansione	Intero	0 <= c < COLONNE
RIGHE	Valore che corrisponde al numero di righe della Mappa	Intero	Variabile globale
COLONNE	Valore che corrisponde al numero di colonne della Mappa	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
esito	Esito della verifica scansione = 0 ERR = 1 OK	Intero	0 <= esito <= 1

Torna al sommario

ALGORITMO

esito = 0

r = leggereCoordinateX (posizione)

c = leggereCoordinateY (posizione)

SE (r >= 0 **AND** r < RIGHE)

ALLORA

SE (c >= 0 **AND** c < COLONNE)

ALLORA

esito = 1

FINE

FINE

verificarePresenza()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
Posizione	Struttura che contiene le coordinate in cui verificare se è presente l'elemento "verificare"	Coordinate	//
Verificare	Elemento di cui verificare la presenza in mappaNavi del giocatore	Carattere	A <= verificare <= E
letturaMappaNavi	Elemento presente su mappaNavi nella posizione inserita	Carattere	A <= verificare <= E

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
errore	Esito della verifica Presenza = 1 elemento presente = 0 elemento non presente	Intero	0 <= errore <= 1

Torna al sommario

ALGORITMO

errore = 0

letturaMappaNavi = leggereMappaNavi (giocatore, posizione)

SE (letturaMappaNavi = verificare)

ALLORA

errore = 1. //elemento presente

ALTRIMENTI

errore = 0. //elemento non presente

FINE

verificareCoordinate()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
stringaCoordinate	Stringa che contiene le coordinate da verificare	Vettore di caratteri	Dim = MAX_DIM_STRINGA_POS
Posizione	Struttura che contiene le coordinate da verificare	Coordinate	//
Verificare	Elemento da verificare in mappaNavi del giocatore	Carattere	A <= verificare <= E

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
lenStringa	Contiene la lunghezza di stringaCoordinate	Intero	1 <= lenStringa <= MAX_DIM_STRINGA_POS

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura che contiene le coordinate verificate	Coordinate	//

ALGORITMO

lenStringa = leggereLunghezzaStringa (stringaCoordinate)

SE (lenStringa >= 2 **AND** lenStringa <= MAX_DIM_STRINGA_POS)

Torna al sommario

ALLORA

SE (elemento in posizione 0 di stringaCoordinate >= A **AND** elemento in posizione 0 distringaCoordinate <= P)

ALLORA

posizione = scrivereCoordinateY (posizione, elemento in posizione 0 di stringaCoordinate – 65)

posizione = scrivereCoordinateIsValid (posizione, 1)

ALTRIMENTI SE (elemento in posizione 0 di stringaCoordinate >= a **AND** elemento in posizione 0 di stringaCoordinate <= p)

ALLORA

posizione = scrivereCoordinateY (posizione, elemento in posizione 0 di stringaCoordinate - 97)

posizione = scrivereCoordinateIsValid (posizione, 1)

FINE

ALTRIMENTI

posizione = ScrivereCoordinateIsValid (posizione, 0)

FINE

SE (leggereCoordinateIsValid (posizione) = 1)

ALLORA

SE (elemento in posizione 1 di stringaCoordinate = 1 **AND** (elemento in posizione 2 di stringaCoordinate >= 0 **AND** elemento in posizione 2 di stringaCoordinate <= 6) **AND** elemento in posizione 3 di stringaCoordinate = carattere di fine stringa)

ALLORA

posizione = scrivereCoordinateX (posizione, (10 + elemento in posizione 2 di stringaCoordinate -49))

posizione = scrivereCoordinateIsValid (posizione, 1)

ALTRIMENTI SE (elemento in posizione 1 di stringaCoordinate >= 1 **AND** elemento in posizione 1 di stringaCoordinate <= 9 **AND** elemento in posizione 2 di stringaCoordinate = simbolo fine stringa)

ALLORA

posizione = scrivereCoordinateX (posizione, (elemento in posizione 1 di posizione = stringaCoordinate – 49))

posizione = scrivereCoordinateIsValid (posizione, 1)

FINE

ALTRIMENTI

Torna al sommario

posizione = scrivereCoordinateIsValid (posizione, 0)

FINE

FINE

ALTRIMENTI

posizione = scrivereCoordinateIsValid (posizione, 0)

FINE

verificareSingolaAffondata()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
Colpo	Struttura che contiene le coordinate della nave colpita	Coordinate	//
colpoBackup	Copia di colpo per effettuare calcoli	Coordinate	//
letturaMappaIncrementata	Valore mappaNavi in posizione x, y+1 di colpo	Carattere	Tutti i caratteri ammessi nella mappaNavi
letturaMappaDecrementata	Valore mappaNavi in posizione x, y-1 di colpo	Carattere	Tutti i caratteri ammessi nella mappa navi
Pos	Valore del campo x di colpoBackup	Intero	0 <= Pos <= 15

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Affondato	Esito della verifica naveAffondata 0 = non affondata 1 = affondata	Intero	0 <= affondato <= 1

ALGORITMO

affondato = 1

colpoBackup = colpo

colpoBackup = incrementareCoordinateX (colpoBackup)

Torna al sommario

pos = leggereCoordinateX (colpoBackup)

letturaMappaIncrementata = leggereMappaNavi (giocatore, colpoBackup)

colpoBackup = scrivereCoordinateX (colpoBackup, pos - 2)

letturaMappaDecrementata = leggereMappaNavi (giocatore, colpoBackup)

SE ((letturaMappaIncrementata >= A **AND** letturaMappaIncrementata <= E) **OR**
(letturaMappaDecrementata >= A **AND** letturaMappaDecrementata <= E))

ALLORA

affondato = 0

ALTRIMENTI

colpoBackup = colpo

colpoBackup = incrementareCoordinateY (colpoBackup)

pos = leggereCoordinateY (colpoBackup)

letturaMappaIncrementata = leggereMappaNavi (giocatore, colpoBackup)

colpoBackup = scrivereCoordinateY (colpoBackup, pos-2)

letturaMappaDecrementata = leggereMappaNavi (giocatore, colpoBackup)

SE ((letturaMappaIncrementata >= A **AND** letturaMappaIncrementata <= E) **OR**
(letturaMappaDecrementata >= A **AND** letturaMappaDecrementata <= E))

ALLORA

affondato = 0

FINE

FINE

verificareAffondatoVerticale()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
Colpo	Struttura che contiene le coordinate della nave colpita	Coordinate	//
letturaMappa	Contiene il valore presente su mappaNavi nella posizione colpita	Carattere	Tutti i caratteri ammessi nella mappaNavi

Torna al sommario

r	Numero di riga in cui si trova la parte di nave colpita	intero	0 <= r < RIGHE
ACQUA	Id dell'Acqua in Mappa	Carattere	Variabile globale
RIGHE	Valore che corrisponde al numero di righe della Mappa	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Affondato	Esito della verifica naveAffondata 0 = non affondata 1 = affondata	Intero	0 <= affondato <= 1

ALGORITMO

```
affondato = 1
letturaMappa = leggereMappaNavi (giocatore, colpo)
r = leggereCoordinateX (colpo)
MENTRE (letturaMappa <> ACQUA AND (r >= 0 AND r < RIGHE))
        r = r + 1
        colpo = scrivereCoordinateX (colpo, r)
        letturaMappa = leggereMappaNavi (giocatore, colpo)
FINE
r = r - 1
colpo = scrivereCoordinateX (colpo, r)
letturaMappa = leggereMappaNavi (giocatore, colpo)
MENTRE (letturaMappa <> ACQUA AND affondato = 1 AND (r >= 0 AND r < RIGHE))
        SE (letturaMappa >= 'A' AND letturaMappa <= 'E')
                ALLORA
                affondato = 0
                ALTRIMENTI
                        r = r - 1
                        colpo = scrivereCoordinateX (colpo, r)
                        letturaMappa = leggereMappaNavi (giocatore, colpo)
```

Torna al sommario

FINE

FINE

verificareAffondatoOrizzontale()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
Colpo	Struttura che contiene le coordinate della nave colpita	Coordinate	//
letturaMappa	Contiene il valore presente su mappaNavi nella posizione colpita	Carattere	Tutti i caratteri ammessi nella mappaNavi
c	Numero di colonna in cui si trova la parte di nave colpita	intero	0 <= c < COLONNE
ACQUA	Id dell'acqua in mappa	Carattere	Variabile globale
COLONNE	Valore che corrisponde al numero di colonne della Mappa	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Affondato	Esito della verifica naveAffondata 0 = non affondata 1 = affondata	Intero	0 <= affondato <= 1

ALGORITMO

affondato = 1

letturaMappa = leggereMappaNavi (giocatore, colpo)

c = leggereCoordinateY (colpo)

MENTRE (letturaMappa <> ACQUA **AND** (c >= 0 **AND** c < COLONNE))

c = c + 1

colpo = scrivereCoordinateY (colpo, c)

Torna al sommario

letturaMappa = leggereMappaNavi (giocatore, colpo)

FINE

c = c - 1

colpo = scrivereCoordinateY (colpo, c)

letturaMappa = leggereMappaNavi (giocatore, colpo)

MENTRE (letturaMappa <> ACQUA **AND** affondato = 1 **AND** (c >= 0 AND c < COLONNE))

SE (letturaMappa >= 'A' **AND** letturaMappa <= 'E')

ALLORA

affondato = 0

ALTRIMENTI

c = c - 1

colpo = scrivereCoordinateY (colpo, c)

letturaMappa = leggereMappaNavi (giocatore, colpo)

FINE

FINE

verificareDirezioneNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore corrente	Player	//
Posizione	Struttura che contiene le coordinate della nave colpita	Coordinate	//
posizioneBackup	Copia di posizione su cui effettuare i calcoli	Coordinate	//
letturaMappa	Contiene il valore presente su mappaNavi nella posizione colpita	Carattere	Tutti i caratteri ammessi nella mappa navi
r	Numero di riga in cui si trova la parte di nave colpita	intero	0 <= r < RIGHE
c	Numero di colonna in cui si trova la parte di nave colpita	intero	0 <= c < COLONNE

Torna al sommario

RIGHE	Valore che corrisponde al numero di righe della Mappa	Intero	Variabile globale
COLONNE	Valore che corrisponde al numero di colonne della Mappa	Intero	Variabile globale
VERTICALE	Valore che corrisponde alla posizione verticale	Intero	Variabile globale
ORIZZONTALE	Valore che corrisponde alla posizione orizzontale	Intero	Variabile globale
COLPITO	Valore che indica una parte di nave colpita	Carattere	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Direzione	Esito della verifica direzioneNave	Intero	= ERR = VERTICALE = ORIZZONTALE

ALGORITMO

```
direzione = ERR.   //-1
posizioneBackup = posizione
r = leggereCoordinateX (posizioneBackup)
SE (r-1 >= 0 OR r+1 < RIGHE)
        ALLORA
        posizioneBackup = ScrivereCoordinateX (posizioneBackup, r+1)
        letturaMappa = leggereMappaNavi (giocatore, posizioneBackup)
        SE (letturaMappa = COLPITO)
                ALLORA
                direzione = VERTICALE
                ALTRIMENTI
                        posizioneBackup = scrivereCoordinateX (posizioneBackup, r-1)
                        letturaMappa = leggereMappaNavi (giocatore, posizioneBackup)
                        SE (letturaMappa = COLPITO)
```

Torna al sommario

```
                    ALLORA
                    direzione = VERTICALE
                FINE
        FINE
FINE

SE (direzione <> VERTICALE)
        ALLORA
        posizioneBackup = posizione
        c = leggereCoordinateY (posizioneBackup)
        SE (c - 1 >= 0 OR c + 1 < COLONNE)
                ALLORA
                posizioneBackup = scrivereCoordinateY (posizioneBackup, c + 1)
                letturaMappa = leggereMappaNavi (giocatore, posizioneBackup)
                SE (letturaMappa = COLPITO)
                        ALLORA
                        direzione = ORIZZONTALE
                        ALTRIMENTI
                                posizioneBackup = scrivereCoordinateY (posizioneBackup, c - 1)
                                letturaMappa = leggereMappaNavi (giocatore, posizioneBackup)
                                SE (letturaMappa = COLPITO)
                                        ALLORA
                                        direzione = ORIZZONTALE
                                FINE
                FINE
        FINE
FINE
```

Torna al sommario

MODULO: stampe

CORPO:

- stampareFile()
- stampareMappa()
- stampareMappe()
- stampareErrore()
- stampareEsitoColpo()
- stampareInizioTurno()
- confermaInizioTurno()
- confermaFineTurno()

stampareFile()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Contiene la posizione del file in cui scrivere	Vettore di caratteri	Tutte le variabili globali che corrispondono al percorso dei file
File	File di cui stampare il contenuto	FILE	//
OFF	Id dell'errore "impossibile aprire il file"	Intero	Variabile globale

ALGORITMO

```
file = aprireFile(posizione, lettura)

SE (file <> INESISTENTE)
        ALLORA
        MENTRE ((text = leggiCarattere(file) <> Carattere di fine file)
                stampareAVideo(text)
        FINE
        ALTRIMENTI
                stampareErrore(OFF).  //impossibile aprire file
        FINE
FINE

chiudereFile(file)
```

Torna al sommario

stampareMappaNavi()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore	Player	//
letturaMappa	Valore in coordinate posizione	Carattere	Tutti i caratteri ammessi nella mappaNavi
RIGHE	Numero di righe della mappa	Intero	Variabile globale
COLONNE	Numero di colonne della mappa	Intero	Variabile globale
MAPPA_NAVI	Id della mappa navi	Intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
r	cicla le righe della mappa	intero	0 <= r <= 15
c	cicla le colonne della mappa	Intero	0 <= c <= 15

ALGORITMO

posizione = scrivereCoordinateX(posizione, r)

stampareAVideo(“MAPPA_NAVI”)

stampareAVideo(“A B C D E F G H I J K L M N O P”)

r = 0

MENTRE(r < RIGHE)

c = 0

posizione = scrivereCoordinateY(posizione, c)

SE(r < 9)

ALLORA

stampareAVideo (“ r+1 “)

ALTRIMENTI

stampareAVideo (“r+1”)

FINE

MENTRE (c < COLONNE)

letturaMappa = leggereMappaNavi (giocatore, posizione)

stampareAVideo (letturaMappa)

Torna al sommario

posizione = incrementareCoordinateY(posizione)

c = c + 1

FINE

posizione = incrementareCoordinateX(posizione)

r = r + 1

FINE

stampareMappe()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Dati del giocatore corrente	Player	//
mappaNavi	Coordinate per la mappaNavi	Coordinate	//
mappaColpi	Coordinate per la mappaColpi	Coordinate	//
letturaMappa	Valore in coordinate mappaNavi/mappaColpi	Carattere	//
stampaRighe	Valore del numero di riga di mappaNavi/mappaColpi	Intero	0 <= stampareRighe <= 15
RIGHE	Numero di righe della mappa	Intero	Variabile globale
COLONNE	Numero di colonne della mappa	Intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
r	Cicla le righe della mappa	Intero	0 <= r <= 15
c	Cicla le colonne della mappa	Intero	0 <= c <= 15

ALGORITMO

mappaNavi = scrivereCoordinateX(mappaNavi, r)

mappaColpi = scrivereCoordinateY(mappaColpi, r)

stampareAVideo(“ MAPPA NAVI MAPPA COLPI”)

stampareAVideo (“A B C D E F G H I J K L M N O P A B C D E F G H I J K L M N O P”)

r = 0

Torna al sommario

MENTRE (r < RIGHE)

c = 0

mappaNavi = scrivereCoordinateX(mappaNavi, c)

mappaColpi = scrivereCoordinateY(mappaColpi, c)

stampaRighe = leggereCoordinateX(mappaNavi)

SE (stampaRighe < 9)

ALLORA

stampareAVideo(stampaRighe + 1)

ALTRIMENTI

stampareAVideo(stampaRighe + 1)

FINE

c = 0

MENTRE (c < COLONNE)

letturaMappa = leggereMappaNavi(giocatore, mappaNavi)

stampareAVideo(letturaMappa)

mappaNavi = incrementareCoordinateY(mappaNavi)

c = c+1

FINE

stampaRighe = leggereCoordinateX(mappaColpi)

SE(stampaRighe < 9)

ALLORA

stampareAVideo(stampaRighe + 1)

ALTRIMENTI

stampareAVideo(stampaRighe + 1)

FINE

c = 0

MENTRE (c < COLONNE)

letturaMappa = leggereMappaColpi(giocatore, mappaColpi)

stampareAVideo(letturaMappa)

mappaColpi = incrementareCoordinateY(mappaColpi)

c = c +1

Torna al sommario

FINE

mappaNavi = incrementareCoordinateX(mappaNavi)

mappaColpi = incrementareCoordinateY(mappaColpi)

r = r + 1

FINE

stampareErrore()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Id	Valore che identifica il tipo di errore	Intero	0 <= id <= 12
MAX_DIM_STRINGA_NOME	Dimensione massima della stringa nomePlayer di giocatore	Vettore di Caratteri	Variabile globale

ALGORITMO

stampareAVideo ("[ERRORE] ")

SE (id = OFF)

ALLORA stampareAVideo ("Impossibile aprire file")

ALTRIMENTI SE (id = WFF)

ALLORA

stampareAVideo ("Impossibile salvare la partita”)

FINE

ALTRIMENTI SE (id = FNE)

ALLORA

stampareAVideo ("Non esiste una partita salvata")

FINE

ALTRIMENTI SE (id = SLO)

Torna al sommario

ALLORA

stampareAVideo ("Devi inserire un nome di lunghezza MAX: “MAX_DIM_STRINGA_NOME)

FINE

ALTRIMENTI SE (id = SOL)

ALLORA

stampareAVideo ("Nave non rientra nella mappa")

FINE

ALTRIMENTI SE (id = SDS)

ALLORA

stampareAVideo ("Distanza tra le navi non rispettata");

FINE

ALTRIMENTI SE (id = ISP)

ALLORA

stampareAVideo ("Posizione nave non valida");

FINE

ALTRIMENTI SE (id = SNA)

ALLORA

stampareAVideo ("Non puoi usare questo tipo di colpo")

FINE

ALTRIMENTI SE (id = SAS)

ALLORA

stampareAVideo ("Colpo gia' lanciato")

FINE

ALTRIMENTI SE (id = SOM)

ALLORA

stampareAVideo ("Le coordinate inserite non corrispondono a una cella della mappa")

FINE

ALTRIMENTI SE (id = ANC)

ALLORA

stampareAVideo ("Risposta non valida, riprova : ")

FINE

Torna al sommario

ALTRIMENTI SE (id = ECS)

ALLORA

stampareAVideo ("Tutte le coordinate scelte sono state gia colpite, riprova")

FINE

ALTRIMENTI SE (id = EAL)

ALLORA

stampareAVideo ("Tutte le coordinate sono state gia' scansionate, riprova")

FINE

FINE

stampareEsitoColpo()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Colpo	Coordinate del colpo lanciato dal giocatore	Coordinate	//
Esito	Valore che indica l'esito del lancio del colpo	Intero	COLPITO, AFFONDATO, MACATO
Righe	Valore righe di colpo	Intero	1 <= righe < RIGHE
Colonne	Valore colonne di colpo	Intero	1 <= colonne < COLONNE
COLPITO	Id nave colpita	Carattere	Variabile globale
AFFONDATO	Id nave affondata	Carattere	Variabile globale
RADAR_NAVE	Id nave individuata con il radar	Intero	Variabile globale
RADAR_VUOTO	Id acqua individuata con il radar	Intero	Variabile globale

ALGORITMO

righe = leggereCoordinateX (colpo) +1

colonne = convertiPosizioneInLettera (leggereCoordinateY (colpo))

SE (esito = COLPITO)

ALLORA

stampareAVideo ("COLPITO in ", colonne, righe)

ALTRIMENTI SE (esito = AFFONDATO)

ALLORA

stampareAVideo ("COLPITO E AFFONDATO in ", colonne, righe)

FINE

Torna al sommario

ALTRIMENTI SE (esito = RADAR_NAVE)

ALLORA

stampareAVideo (“NAVE in ”, colonne, righe)

FINE

ALTRIMENTI SE (esito = RADAR_VUOTO)

ALLORA

stampareAVideo (“VUOTO in ”, colonne, righe)

FINE

ALTRIMENTI

StampareAVideo (“ACQUA in ”, colonne, righe)

FINE

stampareInizioTurno()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Dati del giocatore	Player	//
turniTotali	Turni totali della partita	Intero	> 0
nomePlayer	Nome del player da stampare	Vettore di caratteri	Dim = MAX_DIM_STRINGA_NOME
naviAffondate	Numero di navi affondate dal giocatore	Intero	0 <= naviAffondate <= TOTALE_NAVI

ALGORITMO

nomePlayer = copiareStringa (nomePlayer, leggereNomePlayer (giocatore))

naviAffondate = leggereNaviAffondate (giocatore)

stampareAVideo (“Turno: ”, turniTotali)

stampareAVideo (“Giocatore: ”, nomePlayer)

stampareAVideo (“Navi affondate: ”, naviAffondate)

stampareMappa (giocatore)

Torna al sommario

confermareInizioTurno()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Dati del giocatore	Player	
nomePlayer	Nome del player da stampare	Vettore di caratteri	Dim = MAX_DIM_STRINGA_NOME
inizioTurno	Risposta dall'utente	Intero	0 <= inizioTurno <= 1
SRC_INIZIO_TURNO	Percorso del file inizioTurno da stampare	Vettore di caratteri	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
inizioTurno	Risposta dall'utente	Intero	0 <= inizioTurno <= 1

ALGORITMO

nomePlayer = copiareStringa (nomePlayer, leggereNomePlayer (giocatore))

stampareAVideo ("Giocatore: ", nomePlayer)

stampareFile (SRC_INIZIO_TURNO)

inizioTurno = leggereRispostaValida (0, 1)

confermareFineTurno()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Dati del giocatore	Player	
fineTurno	Risposta dell'utente	Intero	0 <= fineTurno <= 2
SRC_INIZIO_TURNO	Percorso del file InizioTurno da stampare	Vettore di caratteri	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
fineTurno	Risposta dell'utente	Intero	0 <= fineTurno <= 2

ALGORITMO

stampareFile (SRC_FINE_TURNO)

fineTurno = leggereRispostaValida (0, 2)

Torna al sommario

MODULO: setupGame

INTERFACCIA:

- ChiedereNomePlayer()
- PosizionareNavi()
- PosizionareNaviAutomatico()
- PosizionareNaviManuale()
- ScrivereNave()

CORPO:

- setupPartita()

TUTTE LE FUNZIONI:

SetupPartita()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura che contiene i dati della partita	datiPartita	//
rispSchermataIniziale	Risposta dell'utente al menu della schermata iniziale	Intero	0 <= risposta <= 3
rispInizioTurno	Risposta dell'utente al menu del posizionamento navi	Intero	0 <= rispInizioTurno <= 1
Supporto	Struttura dati del giocatore con le navi posizionate	Player	//
SRC_SCHERMATA_INIZIALE	Percorso del file "schermataIniziale"	Vettore di caratteri	Variabile globale
SRC_SALVATAGGIO	Percorso del file "salvataggio"	Vettore di caratteri	Variabile globale
SRC_REGOLAMENTO	Percorso del file "regolamento"	Vettore di caratteri	Variabile globale
ESCI	Corrisponde alla scelta di uscire dal gioco	Intero	Variabile globale
BACK	Corrisponde al "torna indietro" all'interno dei menu	Intero	Variabile globale
CONTINUA	Corrisponde alla scelta di continuare la partita	Intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
Loop	Valore utilizzato per continuare a rimanere o uscire dal menu setup	Intero	0 <= loop <= 1

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura che contiene i dati della partita con i dati modificati	datiPartita	//

ALGORITMO

ESEGUI

stampareFile (SRC_SCHERMATA_INIZIALE)

rispSchermataIniziale = leggereRispostaValida (0, 3)

SE (rispSchermataIniziale = 0)

ALLORA

struttura = scrivereEndGame (struttura, ESCI)

loop = 0

ALTRIMENTI SE (rispSchermataIniziale = 1)

ALLORA

struttura = inizializzaStruttura (struttura)

struttura = scrivereDatiGiocatore1(struttura,chiedereNomePlayer (leggereDatiGiocatore1(struttura), PLAYER_UNO)

struttura = scrivereDatiGiocatore2 (struttura,chiedereNomePlayer (leggereDatiGiocatore2 (struttura), PLAYER_DUE)

rispInizioTurno = confermareInizioTurno (leggereDatiGiocatore1 (struttura)

SE (rispInizioTurno = 1) //Continua

ALLORA

supporto = posizionareNavi (leggereDatiGiocatore1(struttura))

struttura = scrivereDatiGiocatore1 (struttura, supporto)

rispInizioTurno = confermareInizioTurno (leggereDatiGiocatore2 (struttura))

SE(rispInizioTurno = 1)

ALLORA

supporto = posizionareNavi (leggereDatiGiocatore2 (struttura))

struttura = scrivereDatiGiocatore2 (struttura, supporto)

struttura = scrivereEndGame (struttura, CONTINUA)

loop = 0

ALTRIMENTI

Torna al sommario

```
                                    struttura = scrivereEndGame (struttura, BACK)
                            FINE
                            ALTRIMENTI
                            struttura = scrivereEndGame (struttura, BACK)
                    FINE
            FINE
            ALTRIMENTI SE (rispSchermataIniziale = 2)
                    ALLORA
                    SE (verificareCaricamento(SRC_SALVATAGGIO) = 1)
                            ALLORA
                            struttura = caricarePartita (struttura, SRC_SALVATAGGIO)
                            struttura = scrivereEndGame (struttura, CONTINUA)
                            loop = 0
                            ALTRIMENTI
                            loop = 1
                    FINE
            FINE
            ALTRIMENTI SE (rispSchermataIniziale = 3)
                    ALLORA
                    stampareFile (SRC_REGOLAMENTO)
                    stampareAVideo ("Digita 1 per tornare indietro: ")
                    rispSchermataIniziale = leggereRispostaValida (1, 1)
                    loop = 1
            FINE
        FINE
FINCHE' (loop = 1)
```

Torna al sommario

chiedereNomePlayer()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore	Player	//
Turno	Valore che indica il turno della partita	Intero	PLAYER_UNO <= turno PLAYER_DUE
lenStringa	Valore che indica la lunghezza della stringa	Intero	1 <= lenStringa < MAX_DIM_STRINGA_NOME
nomePlayer	Stringa letta da tastiera	Vettore di caratteri	1 <= nomePlayer < MAX_DIM_STRINGA_NOME
MAX_DIM_STRINGA_NOME	Dimensione massima per la stringa del nome del giocatore	Intero	Variabile globale
PLAYER_UNO	Identificativo del turno del giocatore 1	Intero	Variabile globale
PLAYER_DUE	Identificativo del turno del giocatore 2	Intero	Variabile globale
SLO	Errore che indica che il nome del player deve essere MAX_DIM_STRINGA	Intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
Errore	Valore che indica se il nome del giocatore è maggiore o meno di MAX_DIM_STRINGA_NOME	Intero	0 <= loop <= 1
lenStringa	Contiene la lunghezza del nome del giocatore	Intero	1 <= lenStringa <= MAX_DIM_STRINGA_NOME

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore con i campi modificati	Player	//

ALGORITMO

errore = 1

MENTRE (errore = 1)

stampareAVideo (“inserisci nome giocatore “)

SE (turno = PLAYER_UNO)

Torna al sommario

ALLORA

stampareAVideo (“giocatore 1: “)

FINE

SE (turno = PLAYER_DUE)

ALLORA

stampareAVideo (“giocatore 2: “)

FINE

nomePlayer = leggereDaTastiera ( )

lenStringa = lunghezzaStringa(nomePlayer)

SE (lenStringa <= MAX_DIM_STRINGA_NOME)

ALLORA

giocatore = copiareStringa(leggereNomePlayer(giocatore), nomePlayer)

errore = 0

ALTRIMENTI

stampareErrore (SLO) //Lunghezza stringa superata

FINE

FINE

posizionareNavi()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore	Player	//
Scelta	Valore della scelta del posizionamento navi	Intero	1 <= scelta <= 2
SRC_MOD_POS_NAVI	Percorso del file modalità posizionamento navi	Vettore di caratteri	Variabile globale
SRC_POS_AUTO_NAVI	Percorso del file posizionamento automatico navi	Vettore di caratteri	Variabile globale
MAPPA_NAVI	Identificativo della mappa navi	Intero	Variabile globale

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore con i campi modificati	Player	//

ALGORITMO

```
stampareFile (SRC_MOD_POS_NAVI)
scelta = leggereRispostaValida (1, 2)
giocatore = inizializzareMappa (giocatore, MAPPA_NAVI)
SE (scelta = 1)
        ALLORA
        MENTRE (scelta = 1)
                giocatore = posizionareNaviAutomatico (giocatore)
                stampareMappaNavi(giocatore)
                stampareFile (SRC_POS_AUTO_NAVI)
                scelta = leggereRispostaValida (1, 2)
                SE (scelta = 1)
                        ALLORA
                        giocatore = inizializzareMappa (giocatore, MAPPA_NAVI)
                FINE
        FINE
        ALTRIMENTI
        giocatore = posizionareNaviManuale (giocatore)
FINE
```

posizionareNaviAutomatico()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore	Player	//
Posizione	Struttura dati che contiene i dati delle coordinate	Coordinate	//

Torna al sommario

Navi	Contiene i dati delle navi	Vettore di nave	Dimensione = NUMERO_NAVI
Direzione	Valore che indica la direzione in cui inserire la nave nella mappa	Intero	VERTICALE <= direzione <= ORIZZONTALE
numeroNavi	Numero di navi rimaste da inserire nel vettore di navi in posizione i	Vettore di caratteri	NUM_NAVESUPPORTO <= numeroNavi <= NUM_PORTAEREI
isValid	Valore che indica l'esito della verifica sulle coordinate	Intero	0 <= isValid <= 1
esitoVerificaNave	Valore che indica l'esito della verifica del posizionamento della nave	Intero	1 <= esitoVerificaNave <= 2
NUMERO_NAVI	Valore che indica il numero massimo di navi usabili in gioco	Intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
i	Contatore che cicla il numero totale di navi	Intero	0 <= i <= NUMERO_NAVI
K	Contatore che cicla il numero di navi di uno specifico tipo	intero	0 <= k <= numeroNavi

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore con i campi modificati	Player	//

ALGORITMO

```
i = 0
navi = ottieniDatiNavi (navi)
numeroNavi = leggereNumeroNave (posizione i di navi)
MENTRE (i < NUMERO_NAVI)
        k = 0
        MENTRE (k < numeroNavi)
                isValid = 0
```

Torna al sommario

posizione = scrivereCoordinateIsValid (posizione, isValid)

MENTRE (isValid = 0)

posizione = scrivereCoordinateX (posizione, generaNumeroCasuale (1,15))

posizione = scrivereCoordinateY (posizione, generaNumeroCasuale (1,15))

direzione = generaNumeroCasuale (1,2)

esitoVerificaNave = verificareNave (giocatore, posizione, direzione, leggereLunghezzaNave (posizione i di navi))

SE (esitoVerificaNave = 0)

ALLORA

giocatore = scrivereNave (giocatore, posizione, direzione, posizione i di navi)

giocatore = scrivereCoordinateIsValid (posizione, 1)

ALTRIMENTI

posizione = scrivereCoordinateIsValid (posizione, 0)

FINE

isValid = leggereCoordinateIsValid (posizione)

FINE

k = k + 1

numeroNavi = leggereNumeroNave (posizione i di navi)

FINE

i = i + 1

FINE

posizionareNaviManuale()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore	Player	//
Posizione	struttura dati che contiene i dati delle coordinate	Coordinate	//
Navi	Contiene i dati delle navi	Vettore di nave	Dimensione NUMERO_NAVI
Direzione	Valore che indica la direzione in cui inserire la nave nella mappa	Intero	VERTICALE <= direzione <= ORIZZONTALE

Torna al sommario

numeroNavi	Numero di navi rimaste da inserire nel vettore di navi in posizione I	Vettore di caratteri	NUM_NAVESUPPORTO <= numeroNavi <= NUM_PORTAEREI
lunghezzaNave	Indica la lunghezza della nave da posizionare	Intero	1 <= lunghezzaNave <= 5
idNave	Id della nave da posizionare	Carattere	A <= idNave <= E
isValid	Valore che indica l'esito della verifica sulle coordinate	Intero	0 <= isValid <= 1
esitoVerificaNave	Valore che indica l'esito della verifica del posizionamento della nave	Intero	1 <= esitoVerificaNave <= 2
NUMERO_NAVI	Valore che indica il numero massimo di navi usabili in gioco	Intero	Variabile globale
SRC_DIREZIONE	Percorso del file direzione	Vettore di caratteri	Variabile globale
ID_NAVESUPPORTO	Lettera identificativa della nave di supporto	Carattere	Variabile globale
SOL	Numero identificativo dell'errore "nave non rientra nella mappa"	Intero	Variabile globale
SDS	Numero identificativo dell'errore "Distanza navi non rispettata"	Intero	Variabile globale
ISP	Numero identificativo dell'errore "Posizione nave non valida"	intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
i	Contatore che cicla il numero totale di navi	Intero	0 <= i <= NUMERO_NAVI
K	Contatore che cicla il numero di navi di uno specifico tipo	Intero	0 <= k <= numeroNavi
idNave	Contiene l'id o lettera della nave presente in quel momento in posizione i di navi	Carattere	A <= idNave <= E

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore con i campi modificati	datiPartita	//

ALGORITMO

```
i = 0
navi = ottieniDatiNavi (navi)
numeroNavi = leggereNumeroNave (posizione i di navi)
MENTRE (i < NUMERO_NAVI)
    k = 0
    MENTRE (k < numeroNavi)
        isValid = 0
        posizione = scrivereCoordinateIsValid (posizione, isValid)
        MENTRE (isValid = 0)
            stampareMappaNavi (giocatore)
            lunghezzaNave = leggereLunghezzaNave (posizione i di navi)
            stampareAVideo (k, lunghezzaNave, "Posizionamento nave di lunghezza: ")
            posizione = ottenereCoordinate (posizione)
            isValid = leggereCoordinateIsValid (posizione)
            SE (isValid = 1)
                ALLORA
                stampareFile (SRC_DIREZIONE)
                idNave = leggereIdNave (posizione i di navi)
                SE (idNave <> ID_NAVESUPPORTO)
                    ALLORA
                    direzione = leggereRispostaValida (1, 2)
                    ALTRIMENTI
                    direzione = VERTICALE
                FINE
                esitoVerificaNave = verificaNave (giocatore, posizione, direzione,
                lunghezzaNave)
```

Torna al sommario

SE (esitoVerificaNave = 0)

ALLORA

giocatore = scrivereNave (giocatore, posizione, direzione, posizione i di navi)

ALTRIMENTI

posizione = scrivereCoordinateIsValid (posizione, 0)

SE (esitoVerificaNave = 1)

ALLORA

stampareErrore (SOL)// nave non rientra nella mappa

FINE

SE (esitoVerificaNave = 2)

ALLORA

stampareErrore (SDS) // distanza tra le navi non rispettata

FINE

FINE

ALTRIMENTI

stampareErrore (ISP)// posizione nave non valida

FINE

isValid = leggereCoordinateIsValid (posizione)

FINE

k = k + 1

numeroNavi = leggereNumeroNave (posizione i di navi)

FINE

i = i + 1

FINE

Torna al sommario

scrivereNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore	Player	//
Posizione	struttura dati che contiene i dati delle coordinate	Coordinate	//
Nave	Contiene i dati della nave da scrivere	Nave	//
Direzione	Valore che indica la direzione in cui inserire la nave nella mappa	Intero	VERTICALE <= direzione <= ORIZZONTALE
lenNave	Lunghezza della nave da inserire nella mappa	Intero	LEN_NAVESUPPORTO <= lenNave <= LEN_PORTAEREI
VERTICALE	Valore che corrisponde alla posizione verticale	Intero	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
i	Contatore che cicla sulla lunghezza della nave da inserire	Intero	0 <= i <= lenNave

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura che contiene i dati del giocatore con i campi modificati	datiPartita	//

ALGORITMO

```
i = 0
lenNave = leggereLunghezzzaNave (nave)
MENTRE (i < lenNave)
    giocatore = scrivereMappaNavi (giocatore, posizione, leggereIdNave (nave))
    SE (direzione = VERTICALE)
        ALLORA
        posizione = incrementareCoordinateX (posizione)
```

Torna al sommario

```
            ALTRIMENTI
            posizione = incrementareCoordinateY (posizione)
        FINE
        lenNave = leggereLunghezzaNave (nave)
        i = i + 1
FINE
```

Torna al sommario

MODULO: inizializzazioni

INTERFACCIA:

- inizializzareMappa()
- inizializzarePosScansione()
- inizializzareStruttura()
- inizializzareGiocatore()

inizializzareMappa()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Dati del giocatore da inizializzare	Player	//
tipoMappa	Valore che identifica MAPPA_NAVI o MAPPA_COLPI	Intero	MAPPA_NAVI <= tipoMappa <= MAPPA_COLPI
MAPPA_NAVI	Identificativo della mappaNavi di un giocatore	Intero	Variabile globale
RIGHE	Numero massimo di righe di una mappa	Intero	Variabile globale
COLONNE	Numero massimo di colonne di una mappa	Intero	Variabile globale
ACQUA	Id dell'acqua da scrivere in coordinate posizione di mappa	Carattere	Variabile globale

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
r	Contatore che cicla le righe della mappa	Intero	0 <= r <= RIGHE
c	Contatore che cicla le colonne della mappa	Intero	0 <= c <= COLONNE
posizione	coordinate da utilizzare per l'inizializzazione della mappa	Coordinate	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Dati del giocatore con la mappa inizializzata	Player	//

Torna al sommario

ALGORITMO

posizione = scrivereCoordinateX (posizione, 0)

posizione = scrivereCoordinateY (posizione, 0)

r = 0

MENTRE (r < RIGHE)

posizione = scrivereCoordinateY (posizione, 0)

c = 0

MENTRE (c < COLONNE)

SE (tipoMappa = MAPPA_NAVI)

ALLORA

giocatore = scrivereMappaNavi (giocatore, posizione, ACQUA)

ALTRIMENTI

giocatore = scrivereMappaColpi (giocatore, posizione ACQUA)

FINE

posizione = incrementareCoordinateY (posizione)

c = c + 1

FINE

posizione = incrementareCoordinateX (posizione)

r = r + 1

FINE

inizializzarePosScansione()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizioni	Contiene le posizioni, da inizializzare a –1, per colpo a largo raggio e radar	Vettore di coordinate	Dimensione = CELLE_LARGO_RAGGIO
CELLE_LARGO_RAGGIO	Numero di elementi del vettore posizioni	Intero	Variabile globale

Torna al sommario

LAVORO

NOME	DESCRIZIONE	TIPO	VINCOLI
i	Contatore che cicla gli elementi del vettore posizioni	Intero	0 <= i <= CELLE_LARGO_RAGGIO

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizioni	Coordinate di posizioni inizializzate a -1	Vettore di coordinate	Dimensione = CELLE_LAERGO_RAGGIO

ALGORITMO

i = 0

MENTRE (i < CELLE_LARGO_RAGGIO)

elemento in posizione i di posizioni = scrivereCoordinateX (elemento in posizione i di posizioni posizioni + i, -1)

elemento in posizione i di posizioni = scrivereCoordinateY (elemento in posizione i di posizioni posizioni + i, -1)

i = i + 1

FINE

inizializzareStruttura()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Dati della partita da inizializzare	datiPartita	//
Supporto	Copia del giocatore singolo	Player	//
PLAYER_UNO	Identificativo del player uno	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Dati della partita inizializzata	datiPartita	//

Torna al sommario

ALGORITMO

struttura = scrivereTurniTotali (struttura, 1)

struttura = scrivereTurnoPartita (struttura, PLAYER_UNO)

struttura = scrivereEsitoColpi (struttura, 0)

supporto = inizializzareGiocatore (leggereDatiGiocatore1(struttura))

scrivereDatiGiocatore1 (struttura, supporto)

supporto = inizializzareGiocatore (leggereDatiGiocatore2(struttura))

struttura = scrivereDatiGiocatore2 (struttura, supporto)

inizializzareGiocatore()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Dati del giocatore da inizializzare	Player	//
MAPPA_NAVI	Id della mappaNavi di giocatore	Intero	Variabile globale
MAPPA_COLPI	Id della mappaColpi di giocatore	Intero	Variabile globale

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Dati del giocatore inizializzati	Player	//

ALGORITMO

giocatore = inizializzareMappa(giocatore, MAPPA_NAVI)

giocatore = inizializzareMappa(giocatore, MAPPA_COLPI)

giocatore = scrivereLargoRaggio(giocatore, 0)

giocatore = scrivereBombardamentoAereo(giocatore, 0)

giocatore = scrivereRadar(giocatore, 0)

giocatore = scrivereNaviAffondate(giocatore, 0)

Torna al sommario

MODULO: GestioneFile

INTERFACCIA:

- salvarePartita()
- caricarePartita()
- verificareCaricamento()

salvarePartita()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Dati	File su cui salvare i dati della partita	FILE	//
Struttura	Struttura dati da salvare nel file	datiPartita	//
Posizione	Percorso del file in cui scrivere	Vettore di caratteri	Variabili globali che contengono i percorsi ai file

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Dati	File che contiene il salvataggio	FILE	//

ALGORITMO

dati = aprireFile (posizione, “lettura”)

SE (dati <> INESISTENTE)

ALLORA

struttura = leggeredafile(dati)

ALTRIMENTI

stampareErrore (WFF)// scrittura file fallita

FINE

chiudereFile (dati)

Torna al sommario

caricarePartita()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Dati	File su cui salvare i dati della partita salvata	FILE	//
Struttura	Struttura dati in cui salvare il backup partita	datiPartita	//
Posizione	Percorso del file in cui scrivere	Vettore di caratteri	Variabili globali che contengono i percorsi ai file

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati con i dati del backup caricati	datiPartita	//

ALGORITMO

dati = aprireFile (posizione, lettura)

struttura = leggeredafile(dati)

chiudereFile (dati)

verificareCaricamento()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Dati	File su cui vengono salvati i dati della partita	FILE	//
Posizione	Percorso del file in cui scrivere	Vettore di caratteri	Variabili globali che contengono i percorsi aifile

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Verifica	Esito della verifica di esistenza del file	Intero	0 <= verifica <= 1

Torna al sommario

ALGORITMO

dati = aprireFile (posizione, “lettura”)

SE (dati = INESISTENTE)

ALLORA

stampareErrore (FNE)//file non esiste

verifica = 0

ALTRIMENTI

verifica = 1

FINE

chiudereFile (dati)

Torna al sommario

MODULO: StruttureDati

INTERFACCIA:

- ottieniDatiNavi()
- scrivereMappaNavi()
- scrivereMappaColpi()
- scrivereNaviAffondate()
- incrementareNaviAffondate()
- leggereMappaNavi()
- leggereMappaColpi()
- leggereNomePlayer()
- leggereNaviAffondate()
- scrivereColpiSpeciali()
- scrivereBombardamen(toAereo()
- scrivereRadar()
- scrivereLargoRaggio()
- incrementareBombardamentoAereo()
- incrementareRadar()
- incrementareLargoRaggio()
- leggereBombardamentoAereo()
- leggereRadar()
- leggereLargoRaggio()
- leggereColpiSpeciali()
- scrivereIdNave()
- scrivereLunghezzaNave()
- scrivereNumeroNave()
- leggereIdNave()
- leggereLunghezzaNave()
- leggereNumeroNave()
- scrivereCoordinateX()
- scrivereCoordinateY()
- scrivereCoordinateIsValid()
- incrementareCoordinateX()
- incrementareCoordinateY()
- decrementareCoordinateX()
- decrementareCoordinateY()
- leggereCoordinateX()
- leggereCoordinateY()
- leggereCoordinateIsValid()
- scrivereDatiGiocatore1()
- scrivereDatiGiocatore2()
- scrivereTurnoPartita()
- scrivereTurniTotali()
- scrivereEsitoColpi()
- scrivereEndGame()
- leggereDatiGiocatore1()

Torna al sommario

- leggereDatiGiocatore2()
- leggereTurnoPartita()
- leggereTurniTotali()
- leggereEsitoColpi()
- leggereEndGame()
- incrementareTurniTotali()

ottieniDatiNavi()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Array in cui scrivere i dati delle varie navi	Vettore di nave	Dimensione = NUMERO_NAVI
ID_PORTAEREI	Id della nave portaerei	Carattere	Variabile globale
ID_CORAZZATA	Id della nave corazzata	Carattere	Variabile globale
ID_INCROCIATORE	Id della nave incrociatore	Carattere	Variabile globale
ID_CACCIATORPEDINIERE	Id della nave cacciatorpediniere	Carattere	Variabile globale
ID_NAVESUPPORTO	Id della nave supporto	Carattere	Variabile globale
LEN_PORTAEREI	Contiene la lunghezza della nave portaerei	Intero	Variabile globale
LEN_CORAZZATA	Contiene la lunghezza della nave corazzata	Intero	Variabile globale
LEN_INCROCIATORE	Contiene la lunghezza della nave incrociatore	Intero	Variabile globale
LEN_CACCIATORPEDINIERE	Contiene la lunghezza della nave cacciatorpediniere	Intero	Variabile globale
LEN_NAVESUPPORTO	Contiene la lunghezza della nave supporto	Intero	Variabile globale
NUM_PORTAEREI	Numero massimo di navi portaerei	Intero	Variabile globale
NUM_CORAZZATA	Numero massimo di navi corazzata	Intero	Variabile globale
NUM_INCROCIATORE	Numero massimo di navi incrociatore	Intero	Variabile globale
NUM_CACCIATORPEDINIERE	Numero massimo di navi cacciatorpediniere	Intero	Variabile globale
NUM_NAVESUPPORTO	Numero massimo di navi supporto	Intero	Variabile globale

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Array che contiene i dati di tutte le navi modificato	Vettore di nave	Dimensione = NUMERO_NAVI

ALGORITMO

navi = scrivereIdNave (elemento in posizione 0 di navi, ID_PORTAEREI)

navi = scrivereIdNave (elemento in posizione 1 di navi, ID_CORAZZATA)

navi = scrivereIdNave (elemento in posizione 2 di navi, ID_INCROCIATORE)

navi = scrivereIdNave (elemento in posizione 3 di navi, ID_CACCIATORPEDINIERE)

navi = scrivereIdNave (elemento in posizione 4 di navi, ID_NAVESUPPORTO)

navi = scrivereLunghezzaNave (elemento in posizione 0 di navi, LEN_PORTAEREI)

navi = scrivereLunghezzaNave (elemento in posizione 1 di navi, LEN_CORAZZATA)

navi = scrivereLunghezzaNave (elemento in posizione 2 di navi, LEN_INCROCIATORE)

navi = scrivereLunghezzaNave (elemento in posizione 3 di navi, LEN_CACCIATORPEDINIERE)

navi = scrivereLunghezzaNave (elemento in posizione 4 di navi, LEN_NAVESUPPORTO)

navi = scrivereNumeroNave (elemento in posizione 0 di navi, NUM_PORTAEREI)

navi = scrivereNumeroNave (elemento in posizione 1 di navi, NUM_CORAZZATA)

navi = scrivereNumeroNave (elemento in posizione 2 di navi, NUM_INCROCIATORE)

navi = scrivereNumeroNave (elemento in posizione 3 di navi, NUM_CACCIATORPEDINIERE)

navi = scrivereNumeroNave (elemento in posizione 4 di navi, NUM_NAVESUPPORTO)

scrivereMappaNavi()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Posizione	Struttura dati che contiene le coordinate del valore da scrivere in mappaNavi di giocatore	Coordinate	//

Torna al sommario

Valore	Carattere da scrivere in mappaNavi	Carattere	Tutti i caratteri ammessi in mappa navi definiti come variabili globali
Righe	Contiene il numero della riga di mappaNavi in cui scrivere valore	Intero	0 <= righe < RIGHE
Colonne	Contiene il numero della colonna di mappaNavi in cui scrivere valore	Intero	0 <= colonne < COLONNE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

ALGORITMO

righe = leggereCoordinateX (posizione)

colonne = leggereCoordinateY (posizione)

campo mappaNavi in posizione righe, colonne di giocatore = valore

scrivereMappaColpi()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore	Player	//
Posizione	Struttura dati che contiene le coordinate del valore da scrivere in mappaColpi di giocatore	Coordinate	//
Valore	Valore da scrivere in mappaColpi	Carattere	Tutti i caratteri ammessi in mappa colpi definiti come variabili globali
Righe	Contiene il numero della riga di mappaColpi in cui scrivere valore	Intero	0 <= righe < RIGHE
Colonne	Contiene il numero della colonna di mappaColpi in cui scrivere valore	Intero	0 <= colonne < COLONNE

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

ALGORITMO

righe = leggereCoordinateX (posizione)

colonne = leggereCoordinateY (posizione)

campo mappaColpi in posizione righe, colonne di giocatore = valore

scrivereNaviAffondate()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Valore	Valore di navi affondate da scrivere	Intero	>= 0

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

ALGORITMO

Campo naviAffondate di giocatore = valore

Torna al sommario

incrementareNaviAffondate()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Valore	Valore naviAffondate da incrementare	intero	0 <= valore <= TOTALE_NAVI

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

ALGORITMO

valore = leggereNaviAffondate (giocatore)

valore = valore + 1

giocatore = scrivereNaviAffondate (giocatore, valore)

leggereMappaNavi()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Posizione	Struttura dati con le coordinate di mappaNavi da leggere	Coordinate	//
Righe	Contiene il numero della riga di mappaNavi in cui è presente il valore da leggere	Intero	0 <= righe < RIGHE
Colonne	Contiene il numero della colonna di mappaNavi in cui è presente il valore da leggere	Intero	0 <= colonne < COLONNE

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Valore	Valore letto in coordinate posizione di mappaNavi	Carattere	Tutti i caratteri ammessi in mappa navi definiti come variabili globali

ALGORITMO

righe = leggereCoordinateX (posizione)

colonne = leggereCoordinateY (posizone)

valore = campo mappaNavi in posizione righe, colonne di giocatore

leggereMappaColpi()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore	Player	//
Posizione	Struttura dati con le coordinate mappaColpi da leggere	Coordinate	//
Righe	Contiene il numero della riga di mappaColpi in cui è presente il valore da leggere	Intero	0 <= righe < RIGHE
Colonne	Contiene il numero della colonna di mappaColpi in cui è presente il valore da leggere	Intero	0 <= colonne < COLONNE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Valore	Valore letto in coordinate posizione di mappaColpi	Carattere	Tutti i caratteri ammessi in mappa colpi definiti come variabili globali

ALGORITMO

righe = leggereCoordinateX (posizione)

colonne = leggereCoordinateY (posizone)

valore = campo mappaColpi in posizione righe, colonne di giocatore

Torna al sommario

leggereNomePlayer()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
ptrNome	Contiene il nome del giocatore corrente da leggere	Vettore di caratteri	Dimensione = MAX_DIM_STRINGA_NOME

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
ptrNome	Nome del giocatore letto	Vettore di caratteri	Dimensione = MAX_DIM_STRINGA_NOME

ALGORITMO

ptrNome = campo nomePlayer di giocatore

leggereNaviAffondate()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
naviAffondate	Contiene il numero di navi affondate del giocatore corrente	Intero	0 <= naviAffondate <= TOTALE_NAVI

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
naviAffondate	Numero di navi affondate del giocatore corrente letto	Intero	0 <= naviAffondate <= TOTALE_NAVI

ALGORITMO

naviAffondate = campo naviAffondate di giocatore

Torna al sommario

scrivereColpiSpeciali()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Struttura	Struttura che contiene i dati da scrivere nel campo colpiSS di giocatore	colpiSpeciali	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

ALGORITMO

Campo ColpiSS di giocatore = struttura

scrivereBombardamentoAereo()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Valore	Contiene il valore da scrivere nel campo bombardamentoAereo di colpiSS di giocatore	Intero	0 <= valore <= MAX_BOMBARDAMENTO
Copia	Struttura dati che contiene la copia del campo colpiSpeciali di giocatore	colpiSpeciali	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

Torna al sommario

ALGORITMO

copia = leggereColpiSpeciali (giocatore)

campo bombardamentoAereo di copia = valore

giocatore = scrivereColpiSpeciali (giocatore, copia)

scrivereRadar()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Valore	Contiene il valore da scrivere nel campo radar di colpiSS di giocatore	Intero	0 <= valore <= MAX_RADAR
Copia	Struttura dati che conterrà il campo colpiSpeciali di giocatore	colpiSpeciali	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

ALGORITMO

copia = leggereColpiSpeciali (giocatore)

campo radar di copia = valore

giocatore = scrivereColpiSpeciali (giocatore, copia)

scrivereLargoRaggio()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//

Torna al sommario

Valore	Contiene il valore da scrivere nel campo largoRaggio di colpiSS di giocatore	Intero	0 <= valore <= MAX_LARGO_RAGGIO
Copia	Struttura dati che conterrà il campo colpiSpeciali di giocatore	colpiSpeciali	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

ALGORITMO

copia = leggereColpiSpeciali (giocatore)

campo largoRaggio di copia = valore

giocatore = scrivereColpiSpeciali (giocatore, copia)

incrementareBombardamentoAereo()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Valore	Contiene il valore del campo bombardamentoAereo di colpiSS di giocatore da incrementare	Intero	0 <= valore <= MAX_BOMBARDAMENTO

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

Torna al sommario

ALGORITMO

valore = leggereBombardamentoAereo (giocatore)

valore = valore + 1

giocatore = scrivereBombardamentoAereo (giocatore, valore)

incrementareRadar()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Valore	Contiene il valore del campo radar di colpiSS di giocatore da incrementare	Intero	0 <= valore <= MAX_RADAR

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

ALGORITMO

valore = leggereRadar (giocatore)

valore = valore + 1

giocatore = scrivereRadar (giocatore, valore)

incrementareLargoRaggio()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//
Valore	Contiene il valore del campo largoRaggio di colpiSS di giocatore da incrementare	Intero	0 <= valore <= MAX_LARGO_RAGGIO

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati con i dati del giocatore modificati	Player	//

ALGORITMO

valore = leggereLargoRaggio (giocatore)

valore = valore + 1

giocatore = scrivereLargoRaggio (giocatore, valore)

leggereBombardamentoAereo()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Valore	Valore del campo bombardamentoAereo di colpiSS di giocatore	Intero	0 <= valore <= MAX_BOMBARDAMENTO

ALGORITMO

valore = elemento bombardamentoAereo di colpiSS di giocatore

leggereRadar()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Valore	Valore del campo radar di colpiSS di giocatore	Intero	0 <= valore <= MAX_RADAR

ALGORITMO

valore = Elemento radar di colpiSS di giocatore

leggereLargoRaggio()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore corrente	Player	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Valore	Valore del campo largoRaggio di ColpiSS di giocatore	Intero	0 <= valore <= MAX_LARGO_RAGGIO

ALGORITMO

valore = elemento largoRaggio di colpiSS di giocatore

leggereColpiSpeciali()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Struttura dati che contiene i dati del giocatore	Player	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Valore	Campo ColpiSS di giocatore letto	colpiSpeciali	//

Torna al sommario

ALGORITMO

valore = campo colpiSS di giocatore

scrivereIdNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Struttura che contiene i dati della nave presa in considerazione	Nave	//
ID	Contiene il valore da scrivere nel campo id di navi	Carattere	A <= ID <= E

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Struttura che contiene i dati della nave modificati	Nave	//

ALGORITMO

ID = campo id di navi

scrivereLunghezzaNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Struttura che contiene i dati della nave presa in considerazione	Nave	//
Lunghezza	Contiene il valore da scrivere nel campo lunghezza di navi	Intero	1 <= lunghezza <= LEN_PORTAEREI

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Struttura che contiene i dati della nave modificati	Nave	//

Torna al sommario

ALGORITMO

lunghezza = campo lunghezza di navi

scrivereNumeroNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Struttura che contiene i dati della nave presa in considerazione	Nave	//
Numero	Contiene il valore da scrivere nel campo numero di navi	Intero	1 <= numero <= NUM_NAVESUPPORTO

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Struttura che contiene i dati della nave modificati	Nave	//

ALGORITMO

numero = campo numero di navi

leggereIdNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Struttura che contiene i dati della nave presa in considerazione	Nave	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Id	Valore del campo id di navi letto	Carattere	A <= id <= E

ALGORITMO

id = campo id di navi

Torna al sommario

leggereLunghezzaNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Struttura che contiene i dati della nave presa in considerazione	Nave	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Lunghezza	Valore del campo lunghezza di navi letto	Intero	1 <= lunghezza <= LEN_PORTAEREI

ALGORITMO

lunghezza = campo lunghezza di navi

leggereNumeroNave()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Navi	Struttura che contiene i dati della nave presa in considerazione	Nave	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Numero	Valore del campo numero di navi letto	Intero	1 <= lunghezza <= NUM_NAVESUPPORTO

ALGORITMO

numero = campo numero di navi

Torna al sommario

scrivereCoordinateX()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da modificare	Coordinate	//
X	Valore da scrivere nel campo x di posizione	Intero	0 <= x < RIGHE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate con il campo x modificato	Coordinate	//

ALGORITMO

x = campo x di posizione

scrivereCoordinateY()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da modificare	Coordinate	//
y	Valore da scrivere nel campo y di posizione	Intero	0 <= y < COLONNE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate con il campo y modificato	Coordinate	//

ALGORITMO

y = campo y di posizione

Torna al sommario

scrivereCoordinateIsValid()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da modificare	Coordinate	//
isValid	Valore da scrivere nel campo isValid di posizione	Intero	0 <= isValid <= 1

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate con il campo isValid modificato	Coordinate	//

ALGORITMO

isValid = campo isValid id posizione

incrementareCoordinateX()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da modificare	Coordinate	//
Valore	Variabile che contiene il valore di campo x di posizione da incrementare	Intero	0 <= valore < RIGHE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati coordinate con il campo x modificato	Coordinate	//

ALGORITMO

valore = leggereCoordinateX (posizione)

Torna al sommario

valore = valore + 1

posizione = scrivereCoordinateX (posizione, valore)

incrementareCoordinateY()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da modificare	Coordinate	//
Valore	Variabile che contiene il valore del campo y di posizione da incrementare	Intero	0 <= valore < COLONNE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate con il campo y modificato	Coordinate	//

ALGORITMO

valore = leggereCoordinateY (posizione)

valore = valore + 1

posizione = scrivereCoordinateY (posizione, valore)

decrementareCoordinateX()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da modificare	Coordinate	//
Valore	Variabile che contiene il valore di campo x di posizione da decrementare	Intero	0 <= valore < RIGHE

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati coordinate con il campo y modificato	Coordinate	//

ALGORITMO

valore = leggereCoordinateX (posizione)

valore = valore - 1

posizione = scrivereCoordinateX (posizione, valore)

decrementareCoordinateY()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da modificare	Coordinate	//
Valore	Variabile che contiene il valore di campo y di posizione da decrementare	Intero	0 <= valore < COLONNE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati coordinate con il campo y modificato	Coordinate	//

ALGORITMO

valore = leggereCoordinateY (posizione)

valore = valore - 1

posizione = scrivereCoordinateY (posizione, valore)

Torna al sommario

leggereCoordinateX()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da leggere	Coordinate	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
x	Valore del campo x di posizione letto	Intero	0 <= x < RIGHE

ALGORITMO

x = campo x di posizione

leggereCoordinateY()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da leggere	Coordinate	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
y	Valore del campo y di posizione letto	Intero	0 <= y < COLONNE

ALGORITMO

y = campo y di posizione

leggereCoordinateIsValid()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Posizione	Struttura dati che contiene le coordinate da leggere	Coordinate	//

Torna al sommario

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
isValid	Valore del campo isValid di posizione letto	Intero	0 <= isValid <= 1

ALGORITMO

isValid = campo isValid di posizione

scrivereDatiGiocatore1()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da modificare	datiPartita	//
datiGiocatore_1	Struttura dati che contiene i dati da scrivere nel campo datiGiocatore_1 di struttura	Player	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita modificati	datiPartita	//

ALGORITMO

campo datiGiocatore_1 di struttura = datiGiocatore_1

scrivereDatiGiocatore2()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita	datiPartita	//
datiGiocatore_2	Struttura dati che contiene i dati da scrivere nel campo	Player	//

Torna al sommario

	datiGiocatore_2 di struttura		

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita modificati	datiPartita	//

ALGORITMO

campo datiGiocatore_2 di struttura = datiGiocatore_2

scrivereTurnoPartita()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da modificare	datiPartita	//
turnoPartita	Elemento da scrivere nel campo turnoPartita di struttura	Intero	PLAYER_UNO <= turnoPartita <= PLAYER_DUE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita modificati	datiPartita	//

ALGORITMO

campo turnoPartita di struttura = turnoPartita

Torna al sommario

scrivereTurniTotali()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da modificare	datiPartita	//
turniTotali	Elemento da scrivere nel campo turniTotali di struttura	Intero	>0

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita modificati	datiPartita	//

ALGORITMO

campo turniTotali di struttura = turniTotali

scrivereEsitoColpi()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da modificare	datiPartita	//
Esito	Elemento da scrivere nel campo esitoColpo di struttura	Intero	=ESITO_COLPO_OK =ESITO_COLPO_ERR

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita modificati	datiPartita	//

ALGORITMO

campo esitoColpo di struttura = esito

Torna al sommario

scrivereEndGame()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da modificare	datiPartita	//
endGame	Elemento da scrivere nel campo endGame di struttura	Intero	=BACK =ESCI =SALVARE

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita modificati	datiPartita	//

ALGORITMO

campo endGame di struttura = endGame

leggereDatiGiocatore1()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da leggere	datiPartita	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Valore del campo datiGiocatore_1 di struttura letto	Player	//

ALGORITMO

giocatore = campo datiGiocatore_1 di struttura

Torna al sommario

leggereDatiGiocatore2()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da leggere	datiPartita	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Giocatore	Valore del campo datiGiocatore_2 di struttura letto	Player	//

ALGORITMO

giocatore = campo datiGiocatore_2 di struttura

leggereTurnoPartita()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da leggere	datiPartita	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Turno	Valore del campo turnoPartita di struttura letto	Intero	PLAYER_UNO <= turno <= PLAYER_DUE

ALGORITMO

turno = campo turnoPartita di struttura

Torna al sommario

leggereTurniTotali()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita	datiPartita	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
turniTotali	Valore del campo turnoPartita di struttura letto	Intero	> 0

ALGORITMO

turniTotali = campo turniTotali di struttura

leggereEsitoColpi()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da leggere	datiPartita	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Esito	Valore del campo esitoColpo di struttura letto	Intero	= ESITO_COLPO_OK =ESITO_COLPO_ERR

ALGORITMO

esito = campo esitoColpo di struttura

Torna al sommario

leggereEndGame()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da leggere	datiPartita	//

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
endGame	Valore del campo endGame di struttura letto	Intero	=BACK =ESCI =SALVARE

ALGORITMO

endGame = campo endGame di struttura

incrementareTurniTotali()

INPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita da modificare	datiPartita	//
Valore	Variabile che contiene il valore di turniTotali da incrementare	Intero	>0

OUTPUT

NOME	DESCRIZIONE	TIPO	VINCOLI
Struttura	Struttura dati che contiene i dati della partita modificati	datiPartita	//

ALGORITMO

valore = leggereTurniTotali (struttura)

valore = valore + 1

scrivereTurniTotali (struttura, valore)

Torna al sommario