

**MAIN**

PSEUDOCODICE

menu\_principale()

**COSTANTI**

NOME	DESCRIZIONE	VALORE
ID_PRIMA_SEZIONE	Numero identificativo del primo sottocampo	0
ID_SECONDA_SEZIONE	Numero identificativo del secondo sottocampo	1
ID_TERZA_SEZIONE	Numero identificativo del terzo sottocampo	2
ID_QUARTA_SEZIONE	Numero identificativo del quarto sottocampo	3
PRIMA_RIGA_SEZ_0	Numero identificativo della prima riga del primo sottocampo	0
PRIMA_COLONNA_SEZ_0	Numero identificativo della prima colonna del primo sottocampo	0
PRIMA_RIGA_SEZ_1	Numero identificativo della prima riga del secondo sottocampo	0
PRIMA_COLONNA_SEZ_1	Numero identificativo della prima colonna del secondo sottocampo	3
PRIMA_RIGA_SEZ_2	Numero identificativo della prima riga del terzo sottocampo	3
PRIMA_COLONNA_SEZ_2	Numero identificativo della prima colonna del terzo sottocampo	0
PRIMA_RIGA_SEZ_3	Numero identificativo della prima riga del quarto sottocampo	3
PRIMA_COLONNA_SEZ_3	Numero identificativo della prima colonna del quarto sottocampo	3
RIPETIZIONI_NECESSARIE_ROTAZIONE	Numero di ripetizioni delle operazioni da effettuare per ruotare correttamente un campo di 90 gradi	2
DISTANZA_1_PRIMA_POSIZIONE	Distanza dalla prima casella del sottocampo di 1 riga/colonna	1
DISTANZA_2_PRIMA_POSIZIONE	Distanza dalla prima casella del sottocampo di 2 riga/colonna	2
LUNGHEZZA_STRINGA	Lunghezza massima di una stringa	50
DIMENSIONE_CAMPO	Numero di colonne/righe del campo	6
ID_PRIMA_RIGA_COLONNA	Numero identificativo della prima riga/prima colonna	0
ID_SECONDA_RIGA_COLONNA	Numero identificativo della seconda riga/prima colonna	1
ID_TERZA_RIGA_COLONNA	Numero identificativo della terza riga/prima colonna	2
ID_QUARTA_RIGA_COLONNA	Numero identificativo della quarta riga/prima colonna	3
ID_QUINTA_RIGA_COLONNA	Numero identificativo della quinta riga/prima colonna	4
ID_SESTA_RIGA_COLONNA	Numero identificativo della sesta riga/prima colonna	5
FRECCIA_SU	Numero che indica il carattere corrispondente alla freccia su	72
FRECCIA_GIU	Numero che indica il carattere corrispondente alla freccia giù	80

FRECCIA_DESTRA	Numero che indica il carattere corrispondente alla freccia a destra	77
FRECCIA_SINISTRA	Numero che indica il carattere corrispondente alla freccia a sinistra	75
ESC	Numero che indica il carattere corrispondente alla tasto ESC	27
INVIO	Numero che indica il carattere corrispondente alla tasto INVIO	13
ERRORE	Numero che indica un errore	-1
NUM_SOTTOCAMPI	Numero che indica il numero dei sottocampi	4
ID_RUOTA_SX	Numero che indica che l'utente ha scelto di ruotare un campo a SX	4
ID_RUOTA_DX	Numero che indica che l'utente ha scelto di ruotare un campo a DX	5
ID_PRIMO_SLOT	Numero identificativo che indica il primo slot di salvataggio	0
ID_SECONDO_SLOT	Numero identificativo che indica il secondo slot di salvataggio	1
ID_TERZO_SLOT	Numero identificativo che indica il terzo slot di salvataggio	2
ID_QUARTO_SLOT	Numero identificativo che indica il quarto slot di salvataggio	3
ID_QUINTO_SLOT	Numero identificativo che indica il quinto slot di salvataggio	4
ID_SESTO_SLOT	Numero identificativo che indica il sesto slot di salvataggio	5
ID_SETTIMO_SLOT	Numero identificativo che indica il settimo slot di salvataggio	6
ID_OTTAVO_SLOT	Numero identificativo che indica il ottavo slot di salvataggio	7
ID_NONO_SLOT	Numero identificativo che indica il nono slot di salvataggio	8
ID_DECIMO_SLOT	Numero identificativo che indica il decimo slot di salvataggio	9
NUM_SLOT	Numero che indica il numero di slot di salvataggio	10
NUM_POSSIBILI_ROTAZIONI	Numero che indica il numero di possibili rotazioni effettuabili	8
PRIMO_DX	Numero identificativo che indica la rotazione del primo campo a destra	0
SECONDO_DX	Numero identificativo che indica la rotazione del secondo campo a destra	1
TERZO_DX	Numero identificativo che indica la rotazione del terzo campo a destra	2
QUARTO_DX	Numero identificativo che indica la rotazione del quarto campo a destra	3
PRIMO_SX	Numero identificativo che indica la rotazione del primo campo a sinistra	4
SECONDO_SX	Numero identificativo che indica la rotazione del secondo campo a sinistra	5

TERZO_SX	Numero identificativo che indica la rotazione del terzo campo a sinistra	6
QUARTO_SX	Numero identificativo che indica la rotazione del quarto campo a sinistra	7
CINQUE_PEDINE	Numero che indica 5 pedine	5
QUATTRO_PEDINE	Numero che indica 4 pedine	4
TRE_PEDINE	Numero che indica 3 pedine	3
DUE_PEDINE	Numero che indica 2 pedine	2
UNA_PEDINA	Numero che indica 1 pedina	1
ZERO_PEDINE	Numero che indica 0 pedine	0
NUM_CASELLE	Numero che indica il numero di caselle del campo	36
ID_PRIMA_COPPIA	Numero identificativo che indica la prima coppia di pedine di tutte le coppie di pedine presenti sul campo	0
DISTANZA_PRIMA_PEDINA_COPPIA	Numero che indica la distanza nella direzione corrente della prima casella dalla prima pedina della coppia attuale	2
DISTANZA_PRIMA_PEDINA_COPPIA_INV	Numero che indica la distanza nella direzione inversa a quella corrente della prima casella dalla prima pedina della coppia attuale	1
NUM_VOCI_MENU	Numero che indica il numero di voci del menù	5
ID_PRIMA_VOCE_MENU	Numero identificativo che indica la prima voce del menu	0
ID_SECONDA_VOCE_MENU	Numero identificativo che indica la seconda voce del menu	1
ID_TERZA_VOCE_MENU	Numero identificativo che indica la terza voce del menu	2
ID_QUARTA_VOCE_MENU	Numero identificativo che indica la quarta voce del menu	3
ID_QUINTA_VOCE_MENU	Numero identificativo che indica la quinta voce del menu	4
ID_SCHERMATA_REGOLE	Numero identificativo che indica la schermata delle regole	0
ID_SCHERMATA_ISTRUZIONI	Numero identificativo che indica la schermata delle istruzioni	1

## MODULO “tiputente”

Booleano	VERO,FALSO
Casella	PEDINA_GIOCATORE,PEDINA_PC,VUOTA
Campo	(DIMENSIONE_CAMPOxDIMENSIONE_CAMPO)-ple di Caselle
Giocatore	GIOCATORE,PC
Mossa	RUOTA,POSIZIONA
Turno	Struttura con Giocatore(giocatore) e Mossa(mossa)
Stringa	LUNGHEZZA_STRINGA-ple di caratteri
Coordinata	Struttura di due interi (x,y)
Verso_rotazione	DX,SX
Mossa_rotazione	Struttura di intero(num_campo) e Verso_rotazione(verso)
Insieme_campi_ruotati	NUM_POSSIBILI_ROTAZIONI-ple di Campi

Verso\_serie\_pedina    ORIZZONTALE,VERTICALE,DESTRA,SINISTRA  
Pedina\_verso            Struttura di coordinata(coordinata) e verso\_serie\_pedina(verso)

---

---

## MODULO “file”

### FUNZIONE “scrivere\_imp\_file”

INPUT

NOME	DESCRIZIONE	TIPO
nome_file	Stringa che indica il nome del file su cui scrivere	Stringa
valore_da_scrivere	Stringa che indica il valore da scrivere su file	Stringa

OUTPUT

NOME	DESCRIZIONE	TIPO
FILE	File modificato con scritto sopra il valore di valore_da_scrivere	File

PSEUDOCODICE

```
SE (valore_da_scrivere="PC" OR valore_da_scrivere="GIOCATORE")
    ALLORA
        SE (Il file nome_file non esiste)
            ALLORA
                Stampa il messaggio “Errore apertura file”
            ALTRIMENTI
                Scrivi su file valore_da_scrivere
        FINE
    FINE
```

FINE

---

### FUNZIONE “leggere\_imp\_file”

INPUT

NOME	DESCRIZIONE	TIPO
nome_file	Stringa che indica il nome del file da cui leggere le informazioni	Stringa

OUTPUT

NOME	DESCRIZIONE	TIPO
valore_da_leggere	Stringa che indica il valore letto da file	Stringa

PSEUDOCODICE

```
SE (Il file nome_file non esiste)
    ALLORA
        Stampa il messaggio “Errore apertura file”
        valore_da_leggere=""
    ALTRIMENTI
        valore_da_leggere=Leggi da file il valore
```

FINE

---

### FUNZIONE “creare\_file”

INPUT

NOME	DESCRIZIONE	TIPO
nome_file	Stringa che indica il nome del file da creare	Stringa

OUTPUT

NOME	DESCRIZIONE	TIPO
------	-------------	------

FILE	Nuovo file vuoto creato	File
------	-------------------------	------

PSEUDOCODICE

SE (Il file nome\_file non esiste)

ALLORA

Crea il file

FINE

### FUNZIONE “verificare\_esistenza\_file”

INPUT

NOME	DESCRIZIONE	TIPO
nome_file	Stringa che indica il nome del file di cui verificare l’esistenza	Stringa

OUTPUT

NOME	DESCRIZIONE	TIPO
verifica_esistenza	Booleano che indica se il file nome_file esiste o no	Booleano

PSEUDOCODICE

SE (Il file nome\_file non esiste)

ALLORA

verifica\_esistenza=FALSE

ALTRIMENTI

verifica\_esistenza=VERO

FINE

### FUNZIONE “leggere\_data\_partita”

INPUT

NOME	DESCRIZIONE	TIPO
nome_file	Stringa che indica il nome del file da cui leggere la data e l’ora della partita	Stringa

LAVORO

NOME	DESCRIZIONE	TIPO
giorno_g	Stringa che indica il nome del giorno in cui è stata salvata la partita	Stringa
mese	Stringa che indica il mese in cui è stata salvata la partita	Stringa
giorno_num	Stringa che indica il numero del giorno in cui è stata salvata la partita	Stringa
ora	Stringa che indica l’ora in cui è stata salvata la partita	Stringa
anno	Stringa che indica l’anno in cui è stata salvata la partita	Stringa

OUTPUT

NOME	DESCRIZIONE	TIPO
valore_da_leggere	Stringa che indica la data/ora della partita salvata da caricare	Stringa

PSEUDOCODICE

SE (Il file nome\_file non esiste)

ALLORA

Stampa il messaggio “Errore apertura file”

valore\_da\_leggere=””

ALTRIMENTI

Leggi da file giorno\_g,mese,giorno\_num,ora, anno  
valore\_da\_leggere=giorno\_g+mese+giorno\_num+ora+anno

FINE

## FUNZIONE “scrivere\_file\_partite”

### INPUT

NOME	DESCRIZIONE	TIPO
nome_file	Stringa che indica il nome del file su cui scrivere campo_gioco	Stringa
turno_gioco	Turno attuale da scrivere su file	Turno
campo_gioco	Campo della partita attuale da scrivere su file	Campo

### LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe del campo	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
j	Contatore delle colonne del campo	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
ora_data	Ora e data attuali, corrispondenti al momento del salvataggio della partita	Ora e data

### OUTPUT

NOME	DESCRIZIONE	TIPO
File	File dove sono stati salvati ora_data campo_gioco e turno_gioco	File

### PSEUDOCODICE

```
SE (Il file nome_file non esiste)
    ALLORA
        Stampa il messaggio “Errore apertura file”
    ALTRIMENTI
        Scrivi su file ora_data
        i=ID_PRIMA_RIGA_COLONNA
        MENTRE(i<DIMENSIONE_CAMPO)
            j=ID_PRIMA_RIGA_COLONNA
            MENTRE(j<DIMENSIONE_CAMPO)
                SE(leggere(campo_gioco,i,j)=VUOTA)
                    ALLORA
                        Scrivi su file “VUOTA”
                    ALTRIMENTI SE(leggere(campo_gioco,i,j)=PEDINA_GIOCATORE)
                        ALLORA
                            Scrivi su file “PEDINA_GIOCATORE”
                        ALTRIMENTI SE(leggere(campo_gioco,i,j)=PEDINA_PC)
                            ALLORA
                                Scrivi su file “PEDINA_PC”
                FINE
            j=spostare_avanti(j)
        FINE
        i=spostare_avanti(i)
    FINE
SE(turno_gioco.giocatore=GIOCATORE)
    ALLORA
        SE(turno_gioco.mossa=POSIZIONA)
            ALLORA
```

```

        Scrivi su file "GIOCATORE"
        Scrivi su file "POSIZIONA"
    ALTRIMENTI
        Scrivi su file "GIOCATORE"
        Scrivi su file "RUOTA"
    ALTRIMENTI SE(turno_gioco.giocatore=PC)
        ALLORA
            SE(turno_gioco.mossa=POSIZIONA)
                ALLORA
                    Scrivi su file "PC"
                    Scrivi su file "POSIZIONA"
                ALTRIMENTI
                    Scrivi su file "PC"
                    Scrivi su file "RUOTA"
    FINE

```

FINE

### FUNZIONE "leggere\_campo\_gioco\_partita"

INPUT

NOME	DESCRIZIONE	TIPO
nome_file	Stringa che indica il nome del file da cui leggere il campo di gioco	Stringa

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe del campo	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
j	Contatore delle colonne del campo	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
temp	Stringa temporanea	Stringa

OUTPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco letto da file	Campo

```

campo_gioco=inizializzare()
SE(Il file nome_file non esiste)
    ALLORA
        Stampa il messaggio "Errore apertura file"
    ALTRIMENTI
        i=ID_PRIMA_RIGA_COLONNA
        j=ID_PRIMA_RIGA_COLONNA
        MENTRE(Il file non è finito)
            temp=leggi_riga_attuale_da_file
            SE(temp="VUOTA")
                ALLORA
                    scrivere(campo_gioco,i,j,VUOTA)
                    SE(j=ID_SESTA_RIGA_COLONNA)
                        ALLORA
                            i=spostare_avanti(i)
                            j=ID_PRIMA_RIGA_COLONNA
                        ALTRIMENTI
                            j=spostare_avanti(j)
                FINE
            ALTRIMENTI SE(temp="PEDINA_GIOCATORE")

```

```

        ALLORA
            scrivere(campo_gioco,i,j,PEDINA_GIOCATORE)
            SE(j=ID_SESTA_RIGA_COLONNA)
                ALLORA
                    i=spostare_avanti(i)
                    j=ID_PRIMA_RIGA_COLONNA
                ALTRIMENTI
                    j=spostare_avanti(j)
            FINE
        ALTRIMENTI SE(temp="PEDINA_PC")
            ALLORA
                scrivere(campo_gioco,i,j,PEDINA_PC)
                SE(j=ID_SESTA_RIGA_COLONNA)
                    ALLORA
                        i=spostare_avanti(i)
                        j=ID_PRIMA_RIGA_COLONNA
                    ALTRIMENTI
                        j=spostare_avanti(j)
            FINE
        FINE
    FINE

```

FINE

## FUNZIONE "leggere\_turno\_gioco\_partita"

INPUT

NOME	DESCRIZIONE	TIPO
nome_file	Stringa che indica il nome del file da cui leggere il campo di gioco	Stringa

LAVORO

NOME	DESCRIZIONE	TIPO
temp	Stringa temporanea	Stringa

OUTPUT

NOME	DESCRIZIONE	TIPO
turno_gioco	Turno di gioco letto da file	Turno

PSEUDOCODICE

```

SE(Il file nome_file non esiste)
    ALLORA
        Stampa il messaggio "Errore apertura file"
    ALTRIMENTI
        MENTRE(Il file non è finito)
            temp=leggi da file la riga attuale
            SE(temp="GIOCATORE")
                ALLORA
                    turno_gioco.giocatore=GIOCATORE
                ALTRIMENTI SE(temp="PC")
                    ALLORA
                        turno_gioco.giocatore=PC
                    ALTRIMENTI SE(temp="RUOTA")
                        ALLORA
                            turno_gioco.mossa=RUOTA
                        ALTRIMENTI SE(temp="POSIZIONA")
                            ALLORA

```



FINE

FINE

FINE

**MODULO “funzionistruttdati”****FUNZIONE “scrivere”**

INPUT

NOME	DESCRIZIONE	TIPO
campo	Campo dove scrivere valore alla riga i e colonna j	Campo
i	Riga dove posizionare valore	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
j	Colonna dove posizionare valore	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
valore	Valore della casella da scrivere alla riga i e alla colonna j	Casella

OUTPUT

NOME	DESCRIZIONE	TIPO
campo	Campo con valore scritto alla riga i e alla colonna j	Campo

PSEUDOCODICE

SE(i>=ID\_PRIMA\_RIGA\_COLONNA AND i<DIMENSIONE\_CAMPO AND j>=ID\_PRIMA\_RIGA\_COLONNA AND j<DIMENSIONE\_CAMPO)

ALLORA

campo alla riga i e alla colonna j=valore

FINE

**FUNZIONE “leggere”**

INPUT

NOME	DESCRIZIONE	TIPO
campo	Campo dove andare a leggere il valore alla riga i e alla colonna j	Campo
i	Riga dove leggere valore	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
j	Colonna dove leggere valore	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO

OUTPUT

NOME	DESCRIZIONE	TIPO
valore	Valore della casella alla riga i e alla colonna j di campo	Casella

PSEUDOCODICE

SE(i>=ID\_PRIMA\_RIGA\_COLONNA AND i<DIMENSIONE\_CAMPO AND j>=ID\_PRIMA\_RIGA\_COLONNA AND j<DIMENSIONE\_CAMPO)

ALLORA

valore=campo alla riga i e alla colonna j

ALTRIMENTI

valore=VUOTA

FINE

**FUNZIONE “cancellare”**

INPUT

NOME	DESCRIZIONE	TIPO
------	-------------	------

campo	Campo dove cancellare valore alla riga i e colonna j	Campo
i	Riga dove cancellare il valore	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
j	Colonna dove cancellare il valore	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO

OUTPUT

NOME	DESCRIZIONE	TIPO
campo	Campo con valore cancellato alla riga i e alla colonna j	Campo

PSEUDOCODICE

```

SE(i>=ID_PRIMA_RIGA_COLONNA AND i<DIMENSIONE_CAMPO AND j>=ID_PRIMA_RIGA_COLONNA AND
  j<DIMENSIONE_CAMPO)
  ALLORA
    Campo alla riga i e alla colonna j=VUOTA
  FINE

```

**FUNZIONE “inizializzare”**

OUTPUT

NOME	DESCRIZIONE	TIPO
campo_di_gioco	Campo di gioco vuoto	Campo

PSEUDOCODICE

```

azzerare(campo_di_gioco)

```

**FUNZIONE “azzerare”**

INPUT

NOME	DESCRIZIONE	TIPO
campo_di_gioco	Campo di gioco da svuotare	Campo

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
j	Contatore delle colonne	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO

OUTPUT

NOME	DESCRIZIONE	TIPO
campo_di_gioco	Campo di gioco vuoto	Campo

PSEUDOCODICE

```

i=ID_PRIMA_RIGA_COLONNA
MENTRE(i<DIMENSIONE_CAMPO)
  j=ID_PRIMA_RIGA_COLONNA
  MENTRE(j<DIMENSIONE_CAMPO)
    cancellare(campo_di_gioco,i,j)
    j=spostare_avanti(j)
  FINE
  i=spostare_avanti(i)
FINE

```

**FUNZIONE “spostare\_avanti”**

INPUT

NOME	DESCRIZIONE	TIPO
------	-------------	------

i	Valore da incrementare	Intero
---	------------------------	--------

#### OUTPUT

NOME	DESCRIZIONE	TIPO
posizione_successiva	Posizione successiva ad i (i+1)	Intero

#### PSEUDOCODICE

posizione\_successiva=i+1

#### FUNZIONE “spostare\_indietro”

##### INPUT

NOME	DESCRIZIONE	TIPO
i	Valore da decrementare	Intero

#### OUTPUT

NOME	DESCRIZIONE	TIPO
posizione_precedente	Posizione precedente ad i (i-1)	Intero

#### PSEUDOCODICE

posizione\_successiva=i-1

#### MODULO “giocatore”

#### FUNZIONE “acquisire\_tasto\_qualsiasi”

##### INPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero

#### OUTPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero

#### PSEUDOCODICE

tasto=acquisire un tasto da tastiera

#### FUNZIONE “controllare\_tasto\_regole”

##### INPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero
i	Numero per controllare i tasti premuti dall’utente	Intero>=ID_SCHERMATA_REGOLE e <=ID_SCHERMATA_ISTRUZIONI

#### OUTPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero

#### PSEUDOCODICE

SE(i>=ID\_SCHERMATA\_REGOLE AND i<=ID\_SCHERMATA\_ISTRUZIONI)

ALLORA

SE(i=ID\_SCHERMATA\_REGOLE)

ALLORA

ESEGUI

tasto=acquisire\_tasto\_qualsiasi()

FINCHE’(tasto≠FRECCIA\_DESTRA AND tasto≠ESC)

FINE

ALTRIMENTI SE(i=ID\_SCHERMATA\_ISTRUZIONI)

ALLORA

ESEGUI

```

        tasto=acquisire_tasto_qualsiasi()
    FINCHE'(tasto≠FRECCIA_SINISTRA AND tasto≠ESC)
    FINE

```

```

    FINE
ALTRIMENTI
    tasto=ERRORE

```

FINE

### FUNZIONE “controllare\_tasto\_menu”

INPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero
scelta	Numero per controllare i tasti premuti dall’utente	Intero>=ID_PRIMA_VOCE_MENU e <NUM_VOCI_MENU

OUTPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero

PSEUDOCODICE

```

SE(scelta>=ID_PRIMA_VOCE_MENU AND scelta<NUM_VOCI_MENU)
    ALLORA
        SE(scelta=ID_PRIMA_VOCE_MENU)
            ALLORA
                ESEGUI
                    tasto=acquisire_tasto_qualsiasi()
                FINCHE'(tasto≠FRECCIA_GIU AND tasto≠INVIO)
                FINE
            ALTRIMENTI SE(tasto=ID_QUINTA_VOCE_MENU)
                ALLORA
                    ESEGUI
                        tasto=acquisire_tasto_qualsiasi()
                    FINCHE'(tasto≠FRECCIA_SU AND tasto≠INVIO)
                    FINE
                ALTRIMENTI
                    ESEGUI
                        tasto=acquisire_tasto_qualsiasi()
                    FINCHE'(tasto≠FRECCIA_SU AND tasto≠FRECCIA_GIU AND tasto≠INVIO)
                    FINE
            FINE
        ALTRIMENTI
            tasto=ERRORE

```

FINE

### FUNZIONE “controllare\_tasto\_posizionamento\_pedina”

INPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero
x	Numero per controllare i tasti premuti dall’utente	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
y	Numero per controllare i tasti premuti dall’utente	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO

OUTPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero

## PSEUDOCODICE

```
SE(y>=ID_PRIMA_RIGA_COLONNA AND y<DIMENSIONE_CAMPO AND x>=ID_PRIMA_RIGA_COLONNA AND
  x<DIMENSIONE_CAMPO)
  ALLORA
    SE(y=ID_PRIMA_RIGA)
      ALLORA
        SE(x=ID_PRIMA_RIGA_COLONNA)
          ALLORA
            ESEGUI
              tasto=acquire_tasto_qualsiasi()
            FINCHE'(tasto≠ESC AND tasto≠FRECCIA_GIU AND tasto≠FRECCIA_DESTRA
              AND tasto≠INVIO)
            FINE
          ALTRIMENTI SE(x=ID_SESTA_RIGA_COLONNA)
            ALLORA
              ESEGUI
                tasto=acquire_tasto_qualsiasi()
              FINCHE'(tasto≠ESC AND tasto≠FRECCIA_SU AND
                tasto≠FRECCIA_DESTRA AND tasto≠INVIO)
              FINE
            ALTRIMENTI
              ESEGUI
                tasto=acquire_tasto_qualsiasi()
              FINCHE'(tasto≠ESC AND tasto≠FRECCIA_GIU AND
                tasto≠FRECCIA_DESTRA AND tasto≠FRECCIA_SU
                AND tasto≠INVIO)
              FINE
          FINE
        ALTRIMENTI SE(y=ID_SESTA_RIGA_COLONNA)
          ALLORA
            SE(x=ID_PRIMA_RIGA_COLONNA)
              ALLORA
                ESEGUI
                  tasto=acquire_tasto_qualsiasi()
                FINCHE'(tasto≠ESC AND tasto≠FRECCIA_GIU AND
                  tasto≠FRECCIA_SINISTRA
                  AND tasto≠INVIO)
                FINE
              ALTRIMENTI SE(x=ID_SESTA_RIGA_COLONNA)
                ALLORA
                  ESEGUI
                    tasto=acquire_tasto_qualsiasi()
                  FINCHE'(tasto≠ESC AND tasto≠FRECCIA_SU AND
                    tasto≠FRECCIA_SINISTRA AND tasto≠INVIO)
                  FINE
                ALTRIMENTI
                  ESEGUI
                    tasto=acquire_tasto_qualsiasi()
                  FINCHE'(tasto≠ESC AND tasto≠FRECCIA_GIU AND
                    tasto≠FRECCIA_SINISTRA
                    AND tasto≠FRECCIA_SU AND tasto≠INVIO)
                  FINE
              FINE
            FINE
          FINE
        FINE
      FINE
    FINE
  FINE
```

ALTRIMENTI

SE(x=ID\_PRIMA\_RIGA\_COLONNA)

ALLORA

ESEGUI

tasto=acquire\_tasto\_qualsiasi()

FINCHE'(tasto≠ESC AND tasto≠FRECCIA\_GIU

AND tasto≠FRECCIA\_SINISTRA

AND tasto≠FRECCIA\_DESTRA AND tasto≠INVIO)

FINE

ALTRIMENTI SE(x=ID\_SESTA\_RIGA\_COLONNA)

ALLORA

ESEGUI

tasto=acquire\_tasto\_qualsiasi()

FINCHE'(tasto≠ESC AND tasto≠FRECCIA\_SU

AND tasto≠FRECCIA\_SINISTRA

AND tasto≠FRECCIA\_DESTRA AND tasto≠INVIO)

FINE

ALTRIMENTI

ESEGUI

tasto=acquire\_tasto\_qualsiasi()

FINCHE'(tasto≠ESC AND tasto≠FRECCIA\_SU

AND tasto≠FRECCIA\_GIU

AND tasto≠FRECCIA\_SINISTRA

AND tasto≠FRECCIA\_DESTRA AND tasto≠INVIO)

FINE

FINE

FINE

ALTRIMENTI

tasto=ERRORE

FINE

## FUNZIONE "controllare\_tasto\_campo\_da\_ruotare"

INPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero
i	Numero per controllare i tasti premuti dall'utente	Intero>=ID_PRIMA_SEZIONE e <NUM_SOTTOCAMPI

OUTPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero

PSEUDOCODICE

SE(i>=ID\_PRIMA\_SEZIONE AND i<NUM\_SOTTOCAMPI)

ALLORA

SE(i=ID\_PRIMA\_SEZIONE)

ALLORA

ESEGUI

tasto=acquire\_tasto\_qualsiasi()

FINCHE'(tasto≠FRECCIA\_DESTRA AND tasto≠INVIO AND tasto≠ESC)

FINE

ALTRIMENTI SE(i=ID\_QUARTA\_SEZIONE)

ALLORA

ESEGUI

tasto=acquire\_tasto\_qualsiasi()

```

        FINCHE'(tasto≠FRECCIA_SINISTRA AND tast≠INVIO AND tast≠ESC)
        FINE
    ALTRIMENTI
        ESEGUI
            tast=acquire_tast_qualsiasi()
        FINCHE'(tasto≠FRECCIA_SINISTRA AND tast≠FRECCIA_DESTRA AND
            tast≠INVIO AND tast≠ESC)
        FINE
    FINE
ALTRIMENTI
    tast=ERRORE
FINE
```

**FUNZIONE “controllare\_tasto\_verso\_rotazione”**

INPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero
i	Numero per controllare i tasti premuti dall’utente	Intero>=ID_RUOTA_SX e <=ID_RUOTA_DX

OUTPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero

```

PSEUDOCODICE
SE(i>=ID_RUOTA_SX AND i<=ID_RUOTA_DX)
    ALLORA
        SE(i=ID_RUOTA_SX)
            ALLORA
                ESEGUI
                    tast=acquire_tast_qualsiasi()
                FINCHE'(tasto≠FRECCIA_DESTRA AND tast≠INVIO AND tast≠ESC)
                FINE
            ALTRIMENTI
                ESEGUI
                    tast=acquire_tast_qualsiasi()
                FINCHE'(tasto≠FRECCIA_SINISTRA AND tast≠INVIO AND tast≠ESC)
                FINE
        FINE
    ALTRIMENTI
        tast=ERRORE
FINE
```

**FUNZIONE “controllare\_tasto\_cambia\_impostazioni”**

INPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero
t	Tasto premuto dall’utente in precedenza	Intero=FRECCIA_DESTRA o =FRECCIA_SINISTRA

OUTPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero

```

PSEUDOCODICE
SE(t=FRECCIA_DESTRA)
```

```

ALLORA
    ESEGUI
        tasto=acquire_tasto_qualsiasi()
    FINCHE'(tasto≠FRECCIA_SINISTRA AND tasto≠INVIO AND tasto≠ESC)
    FINE
ALTRIMENTI SE(t=FRECCIA_SINISTRA)
    ALLORA
        ESEGUI
            tasto=acquire_tasto_qualsiasi()
        FINCHE'(tasto≠FRECCIA_DESTRA AND tasto≠INVIO AND tasto≠ESC)
        FINE
    ALTRIMENTI
        tasto=ERRORE

```

FINE

#### FUNZIONE “controllare\_tasto\_menu\_salvataggio”

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero
i	Numero per controllare i tasti premuti dall’utente	Intero>=ID_PRIMO_SLOT e <NUM_SLOT

OUTPUT

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto sulla tastiera	Intero

PSEUDOCODICE

```

SE(i>=ID_PRIMO_SLOT AND i<NUM_SLOT)
    ALLORA
        SE(i=ID_PRIMO_SLOT)
            ALLORA
                ESEGUI
                    tasto=acquire_tasto_qualsiasi()
                FINCHE'(tasto≠INVIO AND tasto≠FRECCIA_GIU AND tasto≠ESC)
                FINE
            ALTRIMENTI SE(i=ID_DECIMO_SLOT)
                ALLORA
                    ESEGUI
                        tasto=acquire_tasto_qualsiasi()
                    FINCHE'(tasto≠FRECCIA_SU AND tasto≠INVIO AND tasto≠ESC)
                    FINE
            ALTRIMENTI
                ESEGUI
                    tasto=acquire_tasto_qualsiasi()
                FINCHE'(tasto≠FRECCIA_SU AND tasto≠INVIO AND tasto≠FRECCIA_GIU AND tasto≠ESC)
                FINE
        FINE
    ALTRIMENTI
        tasto=ERRORE

```

FINE

#### FUNZIONE “confermare\_operazione)

INPUT

NOME	DESCRIZIONE	TIPO
scelta	Tasto premuto sulla tastiera	Carattere
msg	Messaggio da stampare su schermo	Stringa



## OUTPUT

NOME	DESCRIZIONE	TIPO
conferma	Booleano che indica se l'utente ha confermato l'operazione o no	Booleano

## PSEUDOCODICE

### ESEGUI

pulire\_interfaccia()

Stampa msg

scelta=acquisire carattere da tastiera

FINCHE'(scelta≠'n' AND scelta≠ 'N' AND scelta≠'s' AND scelta≠'S')

FINE

SE(scelta='s' OR scelta='S')

ALLORA

conferma=VERO

ALTRIMENTI SE(scelta='n' OR scelta='N')

ALLORA

conferma=FALSO

FINE

## MODULO "ia"

### FUNZIONE 'generare\_campi\_ruotati'

#### INPUT

NOME	DESCRIZIONE	TIPO
campo_di_gioco	Campo di gioco	Campo

#### LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore dei sottocampi	Intero>=ID_PRIMA_SEZIONE e <NUM_SOTTOCAMPI

## OUTPUT

NOME	DESCRIZIONE	TIPO
campi_ruotati	Insieme di tutte le possibilità di rotazione dei vari sottocampi	Insieme_campi_ruotati

## PSEUDOCODICE

I=ID\_PRIMA\_SEZIONE

MENTRE(i<NUM\_SOTTOCAMPI)

i-esimo elemento di campi\_ruotati=copiare\_campo(campo\_di\_gioco)

(i+NUM\_SOTTOCAMPI)-esimo elemento di campi\_ruotati=campiare\_campo(campo\_di\_gioco)

ruotare\_campo(i-esimo elemento di campi\_ruotati,DX,i)

ruotare\_campo((i+NUMSOTTOCAMPI)-esimo elemento di campi\_ruotati,SX,i)

i=spostare\_avanti(i)

FINE

### FUNZIONE "trovare\_miglior\_ruotato"

#### INPUT

NOME	DESCRIZIONE	TIPO
campo_di_gioco	Campo di gioco	Campo

#### LAVORO

NOME	DESCRIZIONE	TIPO
------	-------------	------

i	Contatore dei sottocampi ruotati	Intero>=PRIMO_DX <NUM_POSSIBILI_ROTAZIONI	e
j	Contatore delle pedine dei campi ruotati	Intero>=PRIMO_DX <NUM_POSSIBILI_ROTAZIONI	e
campi	Insieme di tutte le possibilità di rotazione dei vari sottocampi	Insieme_campi_ruotati	
random	Numero casuale	Intero>=PRIMO_DX <NUM_POSSIBILI_ROTAZIONI	e
max_campo	Intero che indica il massimo numero di pedine in fila fra tutti i campi	Intero>=ZERO_PEDINE <=CINQUE_PEDINE	e
campo	Numero del campo fra i campi ruotati da scegliere come mossa di rotazione	Intero>=PRIMO_DX <NUM_POSSIBILI_ROTAZIONI	e

## OUTPUT

NOME	DESCRIZIONE	TIPO
Mossa	Migliore rotazione da effettuare	Mossa rotazione

## PSEUDOCODICE

campi=generare\_campi\_ruotati(campo\_di\_gioco)

campo=PRIMO\_DX

i=PRIMO\_DX

MENTRE(i<NUM\_POSSIBILI\_ROTAZIONI AND contare\_num\_pedine\_serie(i-esimo elemento di campi,PC,  
CINQUE\_PEDINE)=FALSO)

i=spostare\_avanti(i)

FINE

SE(i>=NUM\_POSSIBILI\_ROTAZIONI)

ALLORA

i=PRIMO\_DX

MENTRE(i<NUM\_POSSIBILI\_ROTAZIONI AND contare\_num\_pedine\_serie(i-esimo elemento di campi,PC,  
QUATTRO\_PEDINE)=FALSO)

i=spostare\_avanti(i)

FINE

FINE

SE(i>=NUM\_POSSIBILI\_ROTAZIONI)

ALLORA

MENTRE(i<NUM\_POSSIBILI\_ROTAZIONI AND contare\_num\_pedine\_serie(i-esimo elemento di campi,  
GIOCATORE,TRE\_PEDINE)=FALSO)

i=spostare\_avanti(i)

FINE

SE(i>=NUM\_POSSIBILI\_ROTAZIONI)

ALLORA

max\_campo=ZERO\_PEDINE

i=PRIMO\_DX

MENTRE(i<NUM\_POSSIBILI\_ROTAZIONI)

j=QUATTRO\_PEDINE

MENTRE(j>ZERO\_PEDINE)

SE(contare\_num\_pedine\_serie(i-esimo elemento di campi,PC,j-1)=VERO)

ALLORA

SE(j>max\_campo)

ALLORA

max\_campo=j

campo=i

FINE

FINE

j=spostare\_indietro(j)

```

    FINE
    i=spostare_avanti(i)
  FINE
ALTRIMENTI
  max_campo=CINQUE_PEDINE
  i=PRIMO_DX
  MENTRE(i<NUM_POSSIBILI_ROTAZIONI)
    SE(contare_num_pedine_serie(i-esimo elemento di campi,GIOCATORE,
      CINQUE_PEDINE)≠VERO)
      ALLORA
        SE(contare_num_pedine_serie(i-esimo elemento di campi,
          GIOCATORE, QUATTRO_PEDINE)=VERO)
          ALLORA
            SE(i<PRIMO_SX)
              ALLORA
                campo=i+NUM_SOTTOCAMPI
                SE(contare_num_pedine_serie(
                  campo-esimo elemento di
                  campi,GIOCATORE,
                  CINQUE_PEDINE)=VERO)
                  ALLORA
                    ESEGUI
                      random=
                      numero
                      casuale
                      fra
                      PRIMO_DX e
                      QUARTO_SX
                      FINCHE'(random
                      =campo)
                      FINE
                      campo=random
                FINE
              FINE
            ALLTRIMENTI
              campo=i-NUM_SOTTOCAMPI
              SE(contare_num_pedine_serie(
                campo-esimo elemento di
                campi,GIOCATORE,
                CINQUE_PEDINE)=VERO)
                ALLORA
                  ESEGUI
                    random=
                    numero
                    casuale
                    fra
                    PRIMO_DX e
                    QUARTO_SX
                    FINCHE'(random
                    =campo)
                    FINE
                    campo=random
              FINE
            FINE
          FINE
        FINE
      FINE
    FINE
  FINE
  ALLTRIMENTI
    FINE
  FINE

```

```

j=ZERO_PEDINE
MENTRE(j<QUATTRO_PEDINE)
    SE(contare_num_pedine_serie(i-esimo
        elemento di campi,GIOCATORE,j+1)
        =FALSO)
        ALLORA
            SE(j<max_campo)
                ALLORA
                    max_campo=j
                    campo=i
            FINE
        FINE
    FINE
    j=spostare_avanti(j)
FINE
FINE
ALTRIMENTI
    SE(i<NUM_SOTTOCAMPI)
        ALLORA
            campo=i+NUM_SOTTOCAMPI
            SE(contare_num_pedine_serie(campo-esimo
                elemento di campi,GIOCATORE,
                CINQUE_PEDINE)=VERO)
                ALLORA
                    ESEGUI
                        random=numero
                        casuale fra PRIMO_DX
                        e QUARTO_SX
                    FINCHE'(random=campo)
                    FINE
                    campo=random
            FINE
        FINE
    ALTRIMENTI
        campo=i-NUM_SOTTOCAMPI
        SE(contare_num_pedine_serie(campo-esimo
            elemento di campi,GIOCATORE,
            CINQUE_PEDINE)=VERO)
            ALLORA
                ESEGUI
                    random=numero
                    casuale fra PRIMO_DX
                    e QUARTO_SX
                FINCHE'(random=campo)
                FINE
                campo=random
            FINE
        FINE
    FINE
    i=spostare_avanti(i)
FINE
FINE
ALTRIMENTI
    campo=i
FINE
FINE
SE(campo<NUM_SOTTOCAMPI)

```

ALLORA

mossa.num\_campo=campo

mossa.verso=DX

ALTRIMENTI

mossa.num\_campo=campo-NUM\_SOTTOCAMPI

mossa.verso=SX

FINE

## FUNZIONE “trovare\_coppie\_pedine\_consecutive”

INPUT

NOME	DESCRIZIONE	TIPO
campo_di_gioco	Campo di gioco	Campo
g	Giocatore attuale	Giocatore

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe	INTERO>=ID_PRIMA_RIGA_COLONNA e<DIMENSIONE_CAMPO
j	Contatore delle colonne	INTERO>=ID_PRIMA_RIGA_COLONNA e<DIMENSIONE_CAMPO
k	Contatore delle coppie di pedine	INTERO>=ID_PRIMA_COPPIA

OUTPUT

NOME	DESCRIZIONE	TIPO
coppia	Insieme delle coppie di pedine del giocatore o del pc sul campo attuale	NUM_CASELLE-ple di pedina_verso

PSEUDOCODICE

k=ID\_PRIMA\_COPPIA

MENTRE(k<NUM\_CASELLE)

k-esimo elemento di coppia.coordinata.x=ERRORE

k-esimo elemento di coppia.coordinata.y=ERRORE

k=spostare\_avanti(k)

FINE

SE(g=GIOCATORE)

ALLORA

k=ID\_PRIMA\_COPPIA

i=ID\_PRIMA\_RIGA\_COLONNA

MENTRE(i<DIMENSIONE\_CAMPO)

j=ID\_PRIMA\_RIGA\_COLONNA

MENTRE(j<DIMENSIONE\_CAMPO)

SE(leggere(campo\_di\_gioco,i,j)=PEDINA\_GIOCATORE)

ALLORA

SE(i≠ID\_SESTA\_RIGA\_COLONNA)

ALLORA

SE(j≠ID\_PRIMA\_RIGA\_COLONNA)

ALLORA

SE(leggere(campo\_di\_gioco,i,spostare\_avanti(j))=PEDINA\_GIOCATORE)

ALLORA

k-esimo elemento di coppia.coordinata.x=i

k-esimo elemento di coppia.coordinata.y=j

k-esimo elemento di coppia.verso=ORIZZONTALE

k=spostare\_avanti(k)

ALTRIMENTI

SE(leggere(campo\_di\_gioco,spostare\_avanti(i),spostare\_avanti(j)

```

    )=PEDINA_GIOCATORE)
    k-esimo elemento di coppia.coordinata.x=i
    k-esimo elemento di coppia.coordinata.y=j
    k-esimo elemento di coppia.verso=SINISTRA
    k=spostare_avanti(k)
    ALTRIMENTI SE(leggere(campo_di_gioco,spostare_avanti(i),j)=
        PEDINA_GIOCATORE)
    k-esimo elemento di coppia.coordinata.x=i
    k-esimo elemento di coppia.coordinata.y=j
    k-esimo elemento di coppia.verso=VERTICALE
    k=spostare_avanti(k)
    ALTRIMENTI
    SE(leggere(campo_di_gioco,spostare_avanti(i),spostare_indietro
        (j)=PEDINA_GIOCATORE)
    k-esimo elemento di coppia.coordinata.x=i
    k-esimo elemento di coppia.coordinata.y=j
    k-esimo elemento di coppia.verso=DESTRA
    k=spostare_avanti(k)
FINE
ALTRIMENTI SE(j=ID_PRIMA_RIGA_COLONNA)
SE(leggere(campo_di_gioco,i,spostare_avanti(j))=PEDINA_GIOCATORE)
    ALLORA
    k-esimo elemento di coppia.coordinata.x=i
    k-esimo elemento di coppia.coordinata.y=j
    k-esimo elemento di coppia.verso=ORIZZONTALE
    k=spostare_avanti(k)
    ALTRIMENTI
    SE(leggere(campo_di_gioco,spostare_avanti(i),spostare_avanti(i)
        =PEDINA_GIOCATORE)
    k-esimo elemento di coppia.coordinata.x=i
    k-esimo elemento di coppia.coordinata.y=j
    k-esimo elemento di coppia.verso=SINISTRA
    k=spostare_avanti(k)
    ALTRIMENTI SE(leggere(campo_di_gioco,spostare_avanti(i),j)=
        PEDINA_GIOCATORE)
    k-esimo elemento di coppia.coordinata.x=i
    k-esimo elemento di coppia.coordinata.y=j
    k-esimo elemento di coppia.verso=VERTICALE
    k=spostare_avanti(k)
FINE
ALTRIMENTI SE(j=ID_SESTA_RIGA_COLONNA)
SE(leggere(campo_di_gioco,spostare_avanti(i),j)=PEDINA_GIOCATORE)
    k-esimo elemento di coppia.coordinata.x=i
    k-esimo elemento di coppia.coordinata.y=j
    k-esimo elemento di coppia.verso=VERTICALE
    k=spostare_avanti(k)
ALTRIMENTI
SE(leggere(campo_di_gioco,spostare_avanti(i),spostare_indietro
    (j)=PEDINA_GIOCATORE)
    k-esimo elemento di coppia.coordinata.x=i
    k-esimo elemento di coppia.coordinata.y=j
    k-esimo elemento di coppia.verso=DESTRA
    k=spostare_avanti(k)
FINE

```

```

        FINE
    ALTRIMENTI
        SE(j≠ID_SESTA_RIGA_COLONNA)
            ALLORA
                SE(leggere(campo_di_gioco,i,spostare_avanti(j))=PEDINA_GIOCATORE)
                    k-esimo elemento di coppia.coordinata.x=i
                    k-esimo elemento di coppia.coordinata.y=j
                    k-esimo elemento di coppia.verso=ORIZZONTALE
                    k=spostare_avanti(k)
            FINE
        FINE
    FINE
    j=spostare_avanti(j)
    FINE
    i=spostare_avanti(i)
    FINE
    ALTRIMENTI SE(g==PC)
    k=ID_PRIMA_COPPIA
    i=ID_PRIMA_RIGA_COLONNA
    MENTRE(i<DIMENSIONE_CAMPO)
        j=ID_PRIMA_RIGA_COLONNA
        MENTRE(j<DIMENSIONE_CAMPO)
            SE(leggere(campo_di_gioco,i,j)=PEDINA_PC)
                ALLORA
                    SE(i≠ID_SESTA_RIGA_COLONNA)
                        ALLORA
                            SE(j≠ID_PRIMA_RIGA_COLONNA)
                                ALLORA
                                    SE(leggere(campo_di_gioco,i,spostare_avanti(j))=PEDINA_PC)
                                        ALLORA
                                            k-esimo elemento di coppia.coordinata.x=i
                                            k-esimo elemento di coppia.coordinata.y=j
                                            k-esimo elemento di coppia.verso=ORIZZONTALE
                                            k=spostare_avanti(k)
                                        ALTRIMENTI
                                            SE(leggere(campo_di_gioco,spostare_avanti(i),spostare_avanti(j))=PEDINA_PC)
                                                k-esimo elemento di coppia.coordinata.x=i
                                                k-esimo elemento di coppia.coordinata.y=j
                                                k-esimo elemento di coppia.verso=SINISTRA
                                                k=spostare_avanti(k)
                                            ALTRIMENTI SE(leggere(campo_di_gioco,spostare_avanti(i),j)=PEDINA_PC)
                                                k-esimo elemento di coppia.coordinata.x=i
                                                k-esimo elemento di coppia.coordinata.y=j
                                                k-esimo elemento di coppia.verso=VERTICALE
                                                k=spostare_avanti(k)
                                            ALTRIMENTI
                                                SE(leggere(campo_di_gioco,spostare_avanti(i),spostare_indietro(j))=PEDINA_PC)
                                                    k-esimo elemento di coppia.coordinata.x=i
                                                    k-esimo elemento di coppia.coordinata.y=j
                                                    k-esimo elemento di coppia.verso=DESTRA

```

```

        k=spostare_avanti(k)
    FINE
    ALTRIMENTI SE(j=ID_PRIMA_RIGA_COLONNA)
    SE(leggere(campo_di_gioco,i,spostare_avanti(j))=PEDINA_PC)
        ALLORA
        k-esimo elemento di coppia.coordinata.x=i
        k-esimo elemento di coppia.coordinata.y=j
        k-esimo elemento di coppia.verso=ORIZZONTALE
        k=spostare_avanti(k)
        ALTRIMENTI
        SE(leggere(campo_di_gioco,spostare_avanti(i),spostare_avanti(i)
        =PEDINA_PC)
        k-esimo elemento di coppia.coordinata.x=i
        k-esimo elemento di coppia.coordinata.y=j
        k-esimo elemento di coppia.verso=SINISTRA
        k=spostare_avanti(k)
        ALTRIMENTI SE(leggere(campo_di_gioco,spostare_avanti(i),j)=
        PEDINA_PC)
        k-esimo elemento di coppia.coordinata.x=i
        k-esimo elemento di coppia.coordinata.y=j
        k-esimo elemento di coppia.verso=VERTICALE
        k=spostare_avanti(k)
    FINE
    ALTRIMENTI SE(j=ID_SESTA_RIGA_COLONNA)
    SE(leggere(campo_di_gioco,spostare_avanti(i),j)=PEDINA_PC)
        k-esimo elemento di coppia.coordinata.x=i
        k-esimo elemento di coppia.coordinata.y=j
        k-esimo elemento di coppia.verso=VERTICALE
        k=spostare_avanti(k)
    ALTRIMENTI
    SE(leggere(campo_di_gioco,spostare_avanti(i),spostare_indietro
    (j)=PEDINA_PC)
        k-esimo elemento di coppia.coordinata.x=i
        k-esimo elemento di coppia.coordinata.y=j
        k-esimo elemento di coppia.verso=DESTRA
        k=spostare_avanti(k)
    FINE
    FINE
    ALTRIMENTI
    SE(j≠ID_SESTA_RIGA_COLONNA)
        ALLORA
        SE(leggere(campo_di_gioco,i,spostare_avanti(j))=PEDINA_PC)
            k-esimo elemento di coppia.coordinata.x=i
            k-esimo elemento di coppia.coordinata.y=j
            k-esimo elemento di coppia.verso=ORIZZONTALE
            k=spostare_avanti(k)
        FINE
    FINE
    FINE
    FINE
    j=spostare_avanti(j)
    FINE
    i=spostare_avanti(i)
    FINE

```



FINE

## FUNZIONE “trovare\_casella\_posizionabile”

### INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo
coppia	Coppia di pedine di cui controllare le posizioni nella verso corrente o in quello inverso	Pedina_verso
g	Giocatore attuale	Giocatore

### LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe/colonne per controllare i limiti del campo	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO

### OUTPUT

NOME	DESCRIZIONE	TIPO
coordinata	Coordinata che indica la casella successiva alla coppia in serie dove è possibile posizionare la pedina	Coordinata

### PSEUDOCODICE

```
SE(coppia.coordinata.x≠ERRORE AND coppia.coordinata.y≠ERRORE)
    SE(g=GIOCATORE)
        ALLORA
            SE(coppia.verso=ORIZZONTALE)
                ALLORA
                    i=DISTANZA_PRIMA_PEDINA_COPPIA
                    MENTRE(coppia.coordinata.y+i<DIMENSIONE_CAMPO AND
                        leggere(campo_gioco,coppia.coordinata.x, coppia.coordinata.y+i)
                        =PEDINA_GIOCATORE)
                        i=spostare_avanti(i)
                FINE
            SE(coppia.coordinata.y+i<DIMENSIONE_CAMPO AND leggere(campo_gioco,coppia.coordinata.x,
                coppia.coordinata.y+i)=VUOTA)
                ALLORA
                    coordinata.x=coppia.coordinata.x
                    coordinata.y=coppia.coordinata.y+i
                ALTRIMENTI
                    i=DISTANZA_PRIMA_PEDINA_COPPIA_INV
                    MENTRE(coppia.coordinata.y-i>=ID_PRIMA_RIGA_COLONNA AND leggere(campo_gioco,
                        coppia.coordinata.x,coppia.coordinata.y-i=PEDINA_GIOCATORE)
                        i=spostare_avanti(i)
                FINE
            SE(coppia.coordinata.y-i>=ID_PRIMA_RIGA_COLONNA AND leggere(campo_gioco,
                coppia.coordinata.x,coppia.coordinata.y-i=VUOTA)
                ALLORA
                    coordinata.x=coppia.coordinata.x
                    coordinata.y=coppia.coordinata.y-i
                ALTRIMENTI
                    coordinata.x=ERRORE
                    coordinata.y=ERRORE
        FINE
```

```

FINE
ALTRIMENTI SE(coppia.verso=VERTICALE)
i=DISTANZA_PRIMA_PEDINA_COPPIA
MENTRE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND
    leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y)
    =PEDINA_GIOCATORE)
i=spostare_avanti(i)
FINE
SE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND leggere(
    campo_gioco,coppia.coordinata.x+i,coppia.coordinata.y)=VUOTA)
    ALLORA
        coordinata.x=coppia.coordinata.x+i
        coordinata.y=coppia.coordinata.y
    ALTRIMENTI
        i=DISTANZA_PRIMA_PEDINA_COPPIA_INV
        MENTRE(coppia.coordinata.x-i>=ID_PRIMA_RIGA_COLONNA AND leggere(campo_gioco,
            coppia.coordinata.x-i,coppia.coordinata.y=PEDINA_GIOCATORE)
            i=spostare_avanti(i)
    FINE
    SE(coppia.coordinata.x-i>=ID_PRIMA_RIGA_COLONNA AND leggere(campo_gioco,
        coppia.coordinata.x-i,coppia.coordinata.y=VUOTA)
        ALLORA
            coordinata.x=coppia.coordinata.x-i
            coordinata.y=coppia.coordinata.y
        ALTRIMENTI
            coordinata.x=ERRORE
            coordinata.y=ERRORE
    FINE
FINE
ALTRIMENTI SE(coppia.verso=DESTRA)
i=DISTANZA_PRIMA_PEDINA_COPPIA
MENTRE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND coppia.coordinata.
    y-i>=ID_PRIMA_RIGA_COLONNA AND
    leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y-i)
    =PEDINA_GIOCATORE)
i=spostare_avanti(i)
FINE
SE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND coppia.coordinata.
    y-i>=ID_PRIMA_RIGA_COLONNA AND
    leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y-i)
    =VUOTA)
    ALLORA
        coordinata.x=coppia.coordinata.x+i
        coordinata.y=coppia.coordinata.y-i
    ALTRIMENTI
        i=DISTANZA_PRIMA_PEDINA_COPPIA_INV
        MENTRE(coppia.coordinata.x-i<DIMENSIONE_CAMPO AND coppia.coordinata.
            Y+i>=ID_PRIMA_RIGA_COLONNA AND
            leggere(campo_gioco,coppia.coordinata.x-i, coppia.coordinata.y+i)
            =PEDINA_GIOCATORE)
            i=spostare_avanti(i)
    FINE
    SE(coppia.coordinata.x-i<DIMENSIONE_CAMPO AND coppia.coordinata.
        Y+i>=ID_PRIMA_RIGA_COLONNA AND

```

```

        leggere(campo_gioco,coppia.coordinata.x-i, coppia.coordinata.y+i)
        =VUOTA)
    ALLORA
    coordinata.x=coppia.coordinata.x-i
    coordinata.y=coppia.coordinata.y+i
    ALTRIMENTI
    coordinata.x=ERRORE
    coordinata.y=ERRORE
    FINE
FINE
ALTRIMENTI SE(coppia.verso=SINISTRA)
i=DISTANZA_PRIMA_PEDINA_COPPIA
MENTRE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND coppia.coordinata.
    y-i>=ID_PRIMA_RIGA_COLONNA AND
    leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y+i)
    =PEDINA_GIOCATORE)
i=spostare_avanti(i)
FINE
SE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND coppia.coordinata.
    Y+i>=ID_PRIMA_RIGA_COLONNA AND
    leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y-i)
    =VUOTA)
    ALLORA
    coordinata.x=coppia.coordinata.x+i
    coordinata.y=coppia.coordinata.y+i
    ALTRIMENTI
    i=DISTANZA_PRIMA_PEDINA_COPPIA_INV
    MENTRE(coppia.coordinata.x-i<DIMENSIONE_CAMPO AND coppia.coordinata.
        Y-i>=ID_PRIMA_RIGA_COLONNA AND
        leggere(campo_gioco,coppia.coordinata.x-i, coppia.coordinata.y-i)
        =PEDINA_GIOCATORE)
        i=spostare_avanti(i)
    FINE
    SE(coppia.coordinata.x-i<DIMENSIONE_CAMPO AND coppia.coordinata.
        Y-i>=ID_PRIMA_RIGA_COLONNA AND
        leggere(campo_gioco,coppia.coordinata.x-i, coppia.coordinata.y-i)
        =VUOTA)
        ALLORA
        coordinata.x=coppia.coordinata.x-i
        coordinata.y=coppia.coordinata.y-i
        ALTRIMENTI
        coordinata.x=ERRORE
        coordinata.y=ERRORE
    FINE
    FINE
FINE
ALTRIMENTI SE(g=PC)
SE(coppia.verso=ORIZZONTALE)
    ALLORA
    i=DISTANZA_PRIMA_PEDINA_COPPIA
    MENTRE(coppia.coordinata.y+i<DIMENSIONE_CAMPO AND
        leggere(campo_gioco,coppia.coordinata.x, coppia.coordinata.y+i)
        =PEDINA_PC)
    i=spostare_avanti(i)

```

```

FINE
SE(coppia.coordinata.y+i<DIMENSIONE_CAMPO AND leggere(campo_gioco,coppia.coordinata.x,
coppia.coordinata.y+i)=VUOTA)
    ALLORA
        coordinata.x=coppia.coordinata.x
        coordinata.y=coppia.coordinata.y+i
    ALTRIMENTI
        i=DISTANZA_PRIMA_PEDINA_COPPIA_INV
        MENTRE(coppia.coordinata.y-i>=ID_PRIMA_RIGA_COLONNA AND leggere(campo_gioco,
            coppia.coordinata.x,coppia.coordinata.y-i=PEDINA_PC)
            i=spostare_avanti(i)
    FINE
SE(coppia.coordinata.y-i>=ID_PRIMA_RIGA_COLONNA AND leggere(campo_gioco,
    coppia.coordinata.x,coppia.coordinata.y-i=VUOTA)
    ALLORA
        coordinata.x=coppia.coordinata.x
        coordinata.y=coppia.coordinata.y-i
    ALTRIMENTI
        coordinata.x=ERRORE
        coordinata.y=ERRORE
    FINE
FINE
ALTRIMENTI SE(coppia.verso=VERTICALE)
i=DISTANZA_PRIMA_PEDINA_COPPIA
MENTRE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND
    leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y)
    =PEDINA_PC)
i=spostare_avanti(i)
FINE
SE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND leggere(
    campo_gioco,coppia.coordinata.x+i,coppia.coordinata.y)=VUOTA)
    ALLORA
        coordinata.x=coppia.coordinata.x+i
        coordinata.y=coppia.coordinata.y
    ALTRIMENTI
        i=DISTANZA_PRIMA_PEDINA_COPPIA_INV
        MENTRE(coppia.coordinata.x-i>=ID_PRIMA_RIGA_COLONNA AND leggere(campo_gioco,
            coppia.coordinata.x-i,coppia.coordinata.y=PEDINA_PC)
            i=spostare_avanti(i)
    FINE
SE(coppia.coordinata.x-i>=ID_PRIMA_RIGA_COLONNA AND leggere(campo_gioco,
    coppia.coordinata.x-i,coppia.coordinata.y=VUOTA)
    ALLORA
        coordinata.x=coppia.coordinata.x-i
        coordinata.y=coppia.coordinata.y
    ALTRIMENTI
        coordinata.x=ERRORE
        coordinata.y=ERRORE
    FINE
FINE
ALTRIMENTI SE(coppia.verso=DESTRA)
i=DISTANZA_PRIMA_PEDINA_COPPIA
MENTRE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND coppia.coordinata.
    y-i>=ID_PRIMA_RIGA_COLONNA AND

```

```

        leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y-i)
        =PEDINA_PC)
i=spostare_avanti(i)
FINE
SE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND coppia.coordinata.
    y-i>=ID_PRIMA_RIGA_COLONNA AND
    leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y-i)
    =VUOTA)
    ALLORA
        coordinata.x=coppia.coordinata.x+i
        coordinata.y=coppia.coordinata.y-i
    ALTRIMENTI
        i=DISTANZA_PRIMA_PEDINA_COPPIA_INV
    MENTRE(coppia.coordinata.x-i<DIMENSIONE_CAMPO AND coppia.coordinata.
        Y+i>=ID_PRIMA_RIGA_COLONNA AND
        leggere(campo_gioco,coppia.coordinata.x-i, coppia.coordinata.y+i)
        =PEDINA_PC)
        i=spostare_avanti(i)
    FINE
SE(coppia.coordinata.x-i<DIMENSIONE_CAMPO AND coppia.coordinata.
    Y+i>=ID_PRIMA_RIGA_COLONNA AND
    leggere(campo_gioco,coppia.coordinata.x-i, coppia.coordinata.y+i)
    =VUOTA)
    ALLORA
        coordinata.x=coppia.coordinata.x-i
        coordinata.y=coppia.coordinata.y+i
    ALTRIMENTI
        coordinata.x=ERRORE
        coordinata.y=ERRORE
    FINE
FINE
ALTRIMENTI SE(coppia.verso=SINISTRA)
i=DISTANZA_PRIMA_PEDINA_COPPIA
MENTRE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND coppia.coordinata.
    y-i>=ID_PRIMA_RIGA_COLONNA AND
    leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y+i)
    =PEDINA_PC)
i=spostare_avanti(i)
FINE
SE(coppia.coordinata.x+i<DIMENSIONE_CAMPO AND coppia.coordinata.
    Y+i>=ID_PRIMA_RIGA_COLONNA AND
    leggere(campo_gioco,coppia.coordinata.x+i, coppia.coordinata.y-i)
    =VUOTA)
    ALLORA
        coordinata.x=coppia.coordinata.x+i
        coordinata.y=coppia.coordinata.y+i
    ALTRIMENTI
        i=DISTANZA_PRIMA_PEDINA_COPPIA_INV
    MENTRE(coppia.coordinata.x-i<DIMENSIONE_CAMPO AND coppia.coordinata.
        Y-i>=ID_PRIMA_RIGA_COLONNA AND
        leggere(campo_gioco,coppia.coordinata.x-i, coppia.coordinata.y-i)
        =PEDINA_PC)
        i=spostare_avanti(i)
    FINE

```

```

SE(coppia.coordinata.x-i<DIMENSIONE_CAMPO AND coppia.coordinata.
Y-i>=ID_PRIMA_RIGA_COLONNA AND
leggere(campo_gioco,coppia.coordinata.x-i, coppia.coordinata.y-i)
=VUOTA)
ALLORA
coordinata.x=coppia.coordinata.x-i
coordinata.y=coppia.coordinata.y-i
ALTRIMENTI
coordinata.x=ERRORE
coordinata.y=ERRORE

```

FINE

FINE

FINE

ALTRIMENTI

coordinata.x=ERRORE

coordinata.y=ERRORE

FINE

## FUNZIONE “scegliere\_coppia”

INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo
g	Giocatore attuale	Giocatore

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle coppie	Intero>=ID_PRIMA_COPPIA e <NUM_CASELLE
coppie	Insieme delle coppie di pedine del giocatore o del pc sul campo attuale	NUM_CASELLE-ple di pedina_verso

OUTPUT

NOME	DESCRIZIONE	TIPO
coordinata	Migliore casella dove posizionare la pedina in base alla situazione del campo	Coordinata

PSEUDOCODICE

SE(g=GIOCATORE OR g=PC)

ALLORA

coppie=trovare\_coppie\_pedine\_consecutive(campo\_gioco,g)

i=ID\_PRIMA\_COPPIA

ESEGUI

Coordinata=trovare\_casella\_posizionabile(campo\_gioco,i-esimo elemento di coppie,g)

i=spostare\_avanti(i)

FINCHE’(i<NUM\_CASELLE AND coordinata.x==ERRORE AND coordinata.y=ERRORE)

FINE

ALTRIMENTI

coordinata.x=ERRORE

coordinata.y=ERRORE

FINE

## FUNZIONE “copiare\_campo”

INPUT

NOME	DESCRIZIONE	TIPO
campo_di_gioco	Campo di gioco da copiare	Campo

## LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
j	Contatore delle colonne	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO

## OUTPUT

NOME	DESCRIZIONE	TIPO
nuovo_campo	Campo di gioco copiato	Campo

## PSEUDOCODICE

```

nuovo_campo=inizializzare()
i=ID_PRIMA_RIGA_COLONNA
MENTRE(i<DIMENSIONE_CAMPO)
    j=ID_PRIMA_RIGA_COLONNA
    MENTRE(j<DIMENSIONE_CAMPO)
        scrivere(nuovo_campo,i,j,leggere(campo_di_gioco,i,j))
        j=spostare_avanti(j)
    FINE
    i=spostare_avanti(i)

```

FINE

**FUNZIONE “posizionare\_per\_priorita”**

## INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo

## OUTPUT

NOME	DESCRIZIONE	TIPO
coordinata	Migliore casella dove posizionare la pedina in base alla priorità delle varie caselle	Coordinata

## PSEUDOCODICE

```

SE(controllare_casella_libera(campo_gioco,ID_SECONDA_RIGA_COLONNA,ID_SECONDA_RIGA_COLONNA)=VERO)
    ALLORA
        coordinata.x=ID_SECONDA_RIGA_COLONNA
        coordinata.y=ID_SECONDA_RIGA_COLONNA
    ALTRIMENTI
        SE(controllare_casella_libera(campo_gioco,ID_QUINTA_RIGA_COLONNA,ID_QUINTA_RIGA_COLONNA)=VERO)
            coordinata.x=ID_QUINTA_RIGA_COLONNA
            coordinata.y=ID_QUINTA_RIGA_COLONNA
        ALTRIMENTI
            SE(controllare_casella_libera(campo_gioco,ID_SECONDA_RIGA_COLONNA,ID_QUINTA_RIGA_COLONNA)=VERO)
                coordinata.x=ID_SECONDA_RIGA_COLONNA
                coordinata.y=ID_QUINTA_RIGA_COLONNA
            ALTRIMENTI
                SE(controllare_casella_libera(campo_gioco,ID_QUINTA_RIGA_COLONNA,ID_SECONDA_RIGA_COLONNA)=VERO)
                    coordinata.x=ID_QUINTA_RIGA_COLONNA
                    coordinata.y=ID_SECONDA_RIGA_COLONNA
                ALTRIMENTI
                    SE(controllare_casella_libera(campo_gioco,ID_TERZA_RIGA_COLONNA,ID_TERZA_RIGA_COLONNA)=VERO)

```

coordinata.x=ID\_TERZA\_RIGA\_COLONNA  
coordinata.y=ID\_TERZA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUARTA\_RIGA\_COLONNA,ID\_QUARTA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_QUARTA\_RIGA\_COLONNA  
coordinata.y=ID\_QUARTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_TERZA\_RIGA\_COLONNA,ID\_QUARTA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_TERZA\_RIGA\_COLONNA  
coordinata.y=ID\_QUARTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUARTA\_RIGA\_COLONNA,ID\_TERZA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_QUARTA\_RIGA\_COLONNA  
coordinata.y=ID\_TERZA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_PRIMA\_RIGA\_COLONNA,ID\_PRIMA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_PRIMA\_RIGA\_COLONNA  
coordinata.y=ID\_PRIMA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_PRIMA\_RIGA\_COLONNA,ID\_SESTA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_PRIMA\_RIGA\_COLONNA  
coordinata.y=ID\_SESTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_SESTA\_RIGA\_COLONNA,ID\_PRIMA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_SESTA\_RIGA\_COLONNA  
coordinata.y=ID\_PRIMA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_SESTA\_RIGA\_COLONNA,ID\_SESTA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_SESTA\_RIGA\_COLONNA  
coordinata.y=ID\_SESTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_PRIMA\_RIGA\_COLONNA,ID\_TERZA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_PRIMA\_RIGA\_COLONNA  
coordinata.y=ID\_TERZA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_PRIMA\_RIGA\_COLONNA,ID\_QUARTA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_PRIMA\_RIGA\_COLONNA  
coordinata.y=ID\_QUARTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_SESTA\_RIGA\_COLONNA,ID\_TERZA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_SESTA\_RIGA\_COLONNA  
coordinata.y=ID\_TERZA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_SESTA\_RIGA\_COLONNA,ID\_QUARTA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_SESTA\_RIGA\_COLONNA  
coordinata.y=ID\_QUARTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_SECONDA\_RIGA\_COLONNA,ID\_TERZA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_SECONDA\_RIGA\_COLONNA  
coordinata.y=ID\_TERZA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_SECONDA\_RIGA\_COLONNA,ID\_QUARTA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_SECONDA\_RIGA\_COLONNA  
coordinata.y=ID\_QUARTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_TERZA\_RIGA\_COLONNA,ID\_SECONDA\_RIGA\_COLONNA)=VERO)  
coordinata.x=ID\_TERZA\_RIGA\_COLONNA  
coordinata.y=ID\_SECONDA\_RIGA\_COLONNA



ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_TERZA\_RIGA\_COLONNA,ID\_QUINTA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_TERZA\_RIGA\_COLONNA  
    coordinata.y=ID\_QUINTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUARTA\_RIGA\_COLONNA,ID\_SECONDA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_QUARTA\_RIGA\_COLONNA  
    coordinata.y=ID\_SECONDA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUARTA\_RIGA\_COLONNA,ID\_QUINTA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_QUARTA\_RIGA\_COLONNA  
    coordinata.y=ID\_QUINTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUINTA\_RIGA\_COLONNA,ID\_TERZA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_QUINTA\_RIGA\_COLONNA  
    coordinata.y=ID\_TERZA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUINTA\_RIGA\_COLONNA,ID\_QUARTA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_QUINTA\_RIGA\_COLONNA  
    coordinata.y=ID\_QUARTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_SECONDA\_RIGA\_COLONNA,ID\_PRIMA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_SECONDA\_RIGA\_COLONNA  
    coordinata.y=ID\_PRIMA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_TERZA\_RIGA\_COLONNA,ID\_SESTA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_TERZA\_RIGA\_COLONNA  
    coordinata.y=ID\_SESTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUARTA\_RIGA\_COLONNA,ID\_PRIMA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_QUARTA\_RIGA\_COLONNA  
    coordinata.y=ID\_PRIMA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUARTA\_RIGA\_COLONNA,ID\_SESTA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_QUARTA\_RIGA\_COLONNA  
    coordinata.y=ID\_SESTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_PRIMA\_RIGA\_COLONNA,ID\_QUINTA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_PRIMA\_RIGA\_COLONNA  
    coordinata.y=ID\_QUINTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_PRIMA\_RIGA\_COLONNA,ID\_SECONDA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_PRIMA\_RIGA\_COLONNA  
    coordinata.y=ID\_SECONDA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUINTA\_RIGA\_COLONNA,ID\_SESTA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_QUINTA\_RIGA\_COLONNA  
    coordinata.y=ID\_SESTA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_QUINTA\_RIGA\_COLONNA,ID\_PRIMA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_QUINTA\_RIGA\_COLONNA  
    coordinata.y=ID\_PRIMA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_SESTA\_RIGA\_COLONNA,ID\_SECONDA\_RIGA\_COLONNA)=VERO)  
    coordinata.x=ID\_SESTA\_RIGA\_COLONNA  
    coordinata.y=ID\_SECONDA\_RIGA\_COLONNA

ALTRIMENTI

SE(controllare\_casella\_libera(campo\_gioco,ID\_SESTA\_RIGA\_COLONNA,ID\_QUINTA\_RIGA\_COLONNA)=VERO)

```

        coordinata.x=ID_SESTA_RIGA_COLONNA
        coordinata.y=ID_QUINTA_RIGA_COLONNA
    ALTRIMENTI
    SE(controllare_casella_libera(campo_gioco,ID_TERZA_RIGA_COLONNA,ID_PRIMA_RIGA_COLONNA)=VERO)
        coordinata.x=ID_TERZA_RIGA_COLONNA
        coordinata.y=ID_PRIMA_RIGA_COLONNA
    ALTRIMENTI
    SE(controllare_casella_libera(campo_gioco,ID_SECONDA_RIGA_COLONNA,ID_SESTA_RIGA_COLONNA)=VERO)
        coordinata.x=ID_SECONDA_RIGA_COLONNA
        coordinata.y=ID_SESTA_RIGA_COLONNA
    ALTRIMENTI
        coordinata.x=ERRORE
        coordinata.y=ERRORE

```

FINE

## MODULO "impostazioni.c"

### FUNZIONE "creare\_impostazioni\_default"

LAVORO

NOME	DESCRIZIONE	TIPO
nome_file_imp_def	Nome del file delle impostazioni di default	Stringa="pentago_impostazioni_default.txt"

OUTPUT

NOME	DESCRIZIONE	TIPO
FILE	Nuovo file delle impostazioni di default creato	FILE

PSEUDOCODICE

```
nome_file_imp_def="pentago_impostazioni_default.txt"
```

```
creare_file(nome_file_imp_def)
```

```
scrivere_imp_file(nome_file_imp_def,"GIOCATORE")
```

### FUNZIONE "modificare\_imp"

INPUT

NOME	DESCRIZIONE	TIPO
turno	Giocatore che si vuole far iniziare per primo	giocatore

LAVORO

NOME	DESCRIZIONE	TIPO
nome_file_imp_mod	Nome del file delle impostazioni di modificate	Stringa="pentago_impostazioni_modificate.txt"

OUTPUT

NOME	DESCRIZIONE	TIPO
FILE	Nuovo file delle impostazioni modificate modificato	FILE

PSEUDOCODICE

```
nome_file_imp_mod="pentago_impostazioni_modificate.txt"
```

```
SE(turno==GIOCATORE OR turno==PC)
```

```
    ALLORA
```

```
        SE(verificare_esistenza_file(nome_file_imp_mod)=VERO)
```

```
            ALLORA
```

```
                SE(turno=GIOCATORE)
```

```
                    ALLORA
```

```
                        scrivere_imp_file(nome_file_imp_mod,"GIOCATORE")
```

```

        ALTRIMENTI
            scrivere_imp_file(nome_file_imp_mod,"PC")
        FINE
    ALTRIMENTI
        creare_file(nome_file_imp_mod)
        SE(turno=GIOCATORE)
            ALLORA
                scrivere_imp_file(nome_file_imp_mod,"GIOCATORE")
            ALTRIMENTI
                scrivere_imp_file(nome_file_imp_mod,"PC")
        FINE
    FINE
FINE

```

## FUNZIONE "leggere\_impostazioni"

### LAVORO

NOME	DESCRIZIONE	TIPO
nome_file_imp_def	Nome del file delle impostazioni di default	Stringa="pentago_impostazioni_default.txt"
nome_file_imp_mod	Nome del file delle impostazioni di modificate	Stringa="pentago_impostazioni_modificate.txt"
turno_l	Stringa usata per leggere dal file delle impostazioni il giocatore attuale ad iniziare per primo	Stringa

### OUTPUT

NOME	DESCRIZIONE	TIPO
turno	Giocatore attuale che inizia per primo la partita letto dal file delle impostazioni	Giocatore

### PSEUDOCODICE

```

nome_file_imp_def="pentago_impostazioni_default.txt"
nome_file_imp_mod="pentago_impostazioni_modificate.txt"
SE(verificare_esistenza_file(nome_file_imp_mod)=VERO)
    ALLORA
        turno_l=leggere_imp_file(nome_file_imp_mod)
        SE(turno_l="GIOCATORE")
            ALLORA
                turno=GIOCATORE
            ALTRIMENTI
                turno=PC
        FINE
    ALTRIMENTI
        SE(verificare_esistenza_file(nome_file_imp_def)≠VERO)
            ALLORA
                creare_impostazioni_default()
            FINE
        turno_l=leggere_imp_file(nome_file_imp_def)
        SE(turno_l="GIOCATORE")
            ALLORA
                turno=GIOCATORE
            ALTRIMENTI
                turno=PC
        FINE
    FINE
FINE

```

---

---

## MODULO “interfacciagrafica”

### FUNZIONE “stampare\_menu”

INPUT

NOME	DESCRIZIONE	TIPO
scelta	Scelta effettuata dal giocatore	Intero>=ID_PRIMA_VOCE_MENU e <NUM_VOCI_MENU

OUTPUT

NOME	DESCRIZIONE	TIPO
MENU	Interfaccia per scegliere una voce del menu	TESTO

SE(scelta>=ID\_PRIMA\_VOCE\_MENU AND scelta<NUM\_VOCI\_MENU)

    ALLORA

        pulire\_interfaccia()

        Stampa “MENU’ PRINCIPALE”

        SE(scelta=ID\_PRIMA\_VOCE\_MENU)

            ALLORA

                Stampa “» Nuova partita «”

            ALTRIMENTI

                Stampa “Nuova partita”

        FINE

        SE(scelta=ID\_SECONDA\_VOCE\_MENU)

            ALLORA

                Stampa “» Carica partita «”

            ALTRIMENTI

                Stampa “Carica partita”

        FINE

        SE(scelta=ID\_TERZA\_VOCE\_MENU)

            ALLORA

                Stampa “» Modifica impostazioni «”

            ALTRIMENTI

                Stampa “Modifica impostazioni”

        FINE

        SE(scelta=ID\_QUARTA\_VOCE\_MENU)

            ALLORA

                Stampa “» Regole del gioco «”

            ALTRIMENTI

                Stampa “Regole del gioco”

        FINE

        SE(scelta=ID\_QUINTA\_VOCE\_MENU)

            ALLORA

                Stampa “» Fine «”

            ALTRIMENTI

                Stampa “Fine”

        FINE

        Stampa “SU/GIU’: Spostati nel menù    INVIO: Effettua scelta”

FINE

---

### FUNZIONE “pulire\_interfaccia”

OUTPUT

NOME	DESCRIZIONE	TIPO
SCHERMATA_VUOTA	Schermata pulita da tutto il testo precedente	TESTO

PSEUDOCODICE

Cancella tutte le stampe effettuate sullo schermo

stampare\_logo()

**FUNZIONE “stampare\_impostazioni”**

INPUT

NOME	DESCRIZIONE	TIPO
turno	Attuale primo giocatore impostato	Giocatore

OUTPUT

NOME	DESCRIZIONE	TIPO
IMPOSTAZIONI	Interfaccia per modificare le impostaizoni	TESTO

PSEUDOCODICE

pulire\_interfaccia()

Stampa “IMPOSTAZIONI”

Stampa “Primo turno:”

SE(turno=PC)

    ALLORA

        Stampa “GIOCATORE           » PC «”

    ALTRIMENTI SE(turno=GIOCATORE)

        ALLORA

            Stampa “» GIOCATORE «           PC”

FINE

**FUNZIONE “stampare\_logo”**

OUTPUT

NOME	DESCRIZIONE	TIPO
LOGO	Logo del gioco	TESTO

PSEUDOCODICE

Stampa “\*\*\*\*\*”

Stampa ”GRUPPO 9 – PENTAGO”

Stampa “\*\*\*\*\*”

**FUNZIONE “stampare\_campo”**

INPUT

NOME	DESCRIZIONE	TIPO
Campo	Campo di gioco	campo
x	Riga attuale dove si è posizionato il giocatore	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO o =ERRORE
Y	Colonna attuale dove si è posizionato il giocatore	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO o =ERRORE

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO
J	Contatore delle colonne	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE_CAMPO

OUTPUT

NOME	DESCRIZIONE	TIPO
------	-------------	------

Campo	Interfaccia per vedere la situazione del campo e scegliere la casella dove posizionare la pedina	TESTO
-------	--	-------

PSEUDOCODICE

```

SE((x=ERRORE AND y=ERRORE) OR (x>=ID_PRIMA_RIGA_COLONNA AND x<DIMENSIONE_CAMPO AND
  y>=ID_PRIMA_RIGA_COLONA AND y<DIMENSIONE_CAMPO))
  ALLORA
    pulire_interfaccia()
    i=0
    MENTRE(i<DIMENSIONE_CAMPO
      j=0
      MENTRE(j<DIMENSIONE_CAMPO
        SE(i=x AND j=y)
          ALLORA
            SE(leggere(campo,i,j)=VUOTA)
              ALLORA
                Stampa "» «"
              ALTRIMENTI SE(leggere(campo,i,j)=PEDINA_PC)
                ALLORA
                  Stampa "» © «"
                ALTRIMENTI SE(leggere(campo,i,j)=
                  PEDINA_GIOCATORE)
                  ALLORA
                    Stampa "» O «"
              FINE
            ALTRIMENTI
              SE(leggere(campo,i,j)=VUOTA)
                ALLORA
                  Stampa " "
                ALTRIMENTI SE(leggere(campo,i,j)=PEDINA_PC)
                  ALLORA
                    Stampa "©"
                  ALTRIMENTI SE(leggere(campo,i,j)=
                    PEDINA_GIOCATORE)
                    ALLORA
                      Stampa "O"
                FINE
              FINE
            j=spostare_avanti(j)
          FINE
        i=spostare_avanti(i)
      FINE
    FINE
  FINE

```

FINE

FUNZIONE “stampare\_scelta\_rotazione”

INPUT		
NOME	DESCRIZIONE	TIPO
scelta	Scelta effettuata dal giocatore	Intero>=ID_PRIMA_SEZIONE e <=ID_RUOTA_DX o =ERRORE

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle scelte di sottocampo e rotazione	Intero>=ID_PRIMA_SEZIONE e <=ID_RUOTA_DX

## OUTPUT

NOME	DESCRIZIONE	TIPO
SCELTA SOTTOCAMPO E ROTAZIONE	Interfaccia per scegliere il sottocampo e la rotazione	TESTO

## PSEUDOCODICE

SE((scelta>=ID\_PRIMA\_SEZIONE AND scelta<=ID\_RUOTA\_DX) OR scelta=ERRORE)

ALLORA

i=0

MENTRE(i<=ID\_RUOTA\_DX)

SE(i=ID\_PRIMA\_SEZIONE)

ALLORA

SE(i=scelta)

ALLORA

STAMPA "» 1 «"

ALTRIMENTI

STAMPA " 1 "

FINE

FINE

SE(i=ID\_SECONDA\_SEZIONE)

ALLORA

SE(i=scelta)

ALLORA

STAMPA "» 2 «"

ALTRIMENTI

STAMPA " 2 "

FINE

FINE

SE(i=ID\_TERZA\_SEZIONE)

ALLORA

SE(i=scelta)

ALLORA

STAMPA "» 3 «"

ALTRIMENTI

STAMPA " 3 "

FINE

FINE

SE(i=ID\_QUARTA\_SEZIONE)

ALLORA

SE(i=scelta)

ALLORA

STAMPA "» 4 «"

ALTRIMENTI

STAMPA " 4 "

FINE

FINE

SE(i=ID\_RUOTA\_SX)

ALLORA

SE(i=scelta)

ALLORA

STAMPA "» SX «"

ALTRIMENTI

STAMPA " SX "

[illegible]



```

                                ALTRIMENTI
                                    Stampa
                                    "VUOTO"
                                FINE
                            FINE
                        FINE
                    SE(j=ID_SECONDO_SLOT)
                        ALLORA
                            SE(j=i)
                                ALLORA
                                    SE(verificare_esistenza_file("slot2.txt")=VERO)
                                        ALLORA
                                            Stampa
                                            "»" leggere_data_partita("slot2.txt") "«"
                                        ALTRIMENTI
                                            Stampa
                                            "» VUOTO «"
                                        FINE
                                    ALTRIMENTI
                                        SE(verificare_esistenza_file("slot2.txt")=VERO)
                                            ALLORA
                                                Stampa il valore di
                                                leggere_data_partita("slot2.txt")
                                            ALTRIMENTI
                                                Stampa
                                                "VUOTO"
                                            FINE
                                        FINE
                                    FINE
                                FINE
                            FINE
                        SE(j=ID_TERZO_SLOT)
                            ALLORA
                                SE(j=i)
                                    ALLORA
                                        SE(verificare_esistenza_file("slot3.txt")=VERO)
                                            ALLORA
                                                Stampa
                                                "»" leggere_data_partita("slot3.txt") "«"
                                            ALTRIMENTI
                                                Stampa
                                                "» VUOTO «"
                                            FINE
                                        ALTRIMENTI
                                            SE(verificare_esistenza_file("slot3.txt")=VERO)
                                                ALLORA
                                                    Stampa il valore di
                                                    leggere_data_partita("slot3.txt")
                                                ALTRIMENTI
                                                    Stampa
                                                    "VUOTO"
                                                FINE
                                            FINE
                                        FINE
                                    FINE
                                FINE
                            FINE
                        SE(j=ID_QUARTO_SLOT)
                            ALLORA

```

```

SE(j=i)
    ALLORA
        SE(verificare_esistenza_file("slot4.txt")=VERO)
            ALLORA
                Stampa
                "»" leggere_data_partita("slot4.txt") "«"
            ALTRIMENTI
                Stampa
                "» VUOTO «"
        FINE
    ALTRIMENTI
        SE(verificare_esistenza_file("slot4.txt")=VERO)
            ALLORA
                Stampa il valore di
                leggere_data_partita("slot4.txt")
            ALTRIMENTI
                Stampa
                "VUOTO"
        FINE
    FINE
SE(j=ID_QUINTO_SLOT)
    ALLORA
        SE(j=i)
            ALLORA
                SE(verificare_esistenza_file("slot5.txt")=VERO)
                    ALLORA
                        Stampa
                        "»" leggere_data_partita("slot5.txt") "«"
                    ALTRIMENTI
                        Stampa
                        "» VUOTO «"
                FINE
            ALTRIMENTI
                SE(verificare_esistenza_file("slot5.txt")=VERO)
                    ALLORA
                        Stampa il valore di
                        leggere_data_partita("slot5.txt")
                    ALTRIMENTI
                        Stampa
                        "VUOTO"
                FINE
        FINE
SE(j=ID_SESTO_SLOT)
    ALLORA
        SE(j=i)
            ALLORA
                SE(verificare_esistenza_file("slot6.txt")=VERO)
                    ALLORA
                        Stampa
                        "»" leggere_data_partita("slot6.txt") "«"
                    ALTRIMENTI
                        Stampa

```

[illegible]

```

                                ALTRIMENTI
                                    Stampa
                                    "VUOTO"
                                FINE
                            FINE
                        FINE
                    SE(j=ID_NONO_SLOT)
                        ALLORA
                            SE(j=i)
                                ALLORA
                                    SE(verificare_esistenza_file("slot9.txt")=VERO)
                                        ALLORA
                                            Stampa
                                            "»" leggere_data_partita("slot9.txt") "«"
                                        ALTRIMENTI
                                            Stampa
                                            "» VUOTO «"
                                        FINE
                                    ALTRIMENTI
                                        SE(verificare_esistenza_file("slot9.txt")=VERO)
                                            ALLORA
                                                Stampa il valore di
                                                leggere_data_partita("slot9.txt")
                                            ALTRIMENTI
                                                Stampa
                                                "VUOTO"
                                        FINE
                                    FINE
                                FINE
                            FINE
                        SE(j=ID_DECIMO_SLOT)
                            ALLORA
                                SE(j=i)
                                    ALLORA
                                        SE(verificare_esistenza_file("slot10.txt")=VERO)
                                            ALLORA
                                                Stampa
                                                "»" leggere_data_partita("slot10.txt")
                                                "«"
                                            ALTRIMENTI
                                                Stampa
                                                "» VUOTO «"
                                            FINE
                                        FINE
                                    ALTRIMENTI
                                        SE(verificare_esistenza_file("slot3.txt")=VERO)
                                            ALLORA
                                                Stampa il valore di
                                                leggere_data_partita("slot10.txt")
                                            ALTRIMENTI
                                                Stampa
                                                "VUOTO"
                                        FINE
                                    FINE
                                FINE
                            FINE
                        j=spostare_avanti(j)
                    FINE
                FINE
            FINE
        FINE
    FINE

```

FINE

FINE

---

---

## MODULO “menu”

### FUNZIONE “menu\_principale”

LAVORO

NOME	DESCRIZIONE	TIPO
scelta	Intero che indica la voce attuale su cui si trova l’utente	Intero>=ID_PRIMA_VOCE_MENU e <NUM_VOCE_MENU
inserimento_menu	Tasto premuto dall’utente	Intero
esci_gioco	Booleano che indica se l’utente ha deciso di uscire dal gioco	Booleano

PSEUDOCODICE

esci\_gioco=FALSO

scelta=ID\_PRIMA\_VOCE\_MENU

ESEGUI

stampare\_menu(scelta)

inserimento\_menu=controllare\_tasto\_menu(scelta)

SE(inserimento\_menu=FRECCIA\_SU)

ALLORA

scelta=spostare\_indietro(scelta)

ALTRIMENTI SE(inserimento\_menu=FRECCIA\_GIU)

ALLORA

Scelta=spostare\_avanti(scelta)

ALTRIMENTI SE(inserimento\_menu=INVIO)

ALLORA

SE(scelta=ID\_PRIMA\_VOCE\_MENU)

ALLORA

Pulire\_interfaccia()

nuova\_partita()

ALTRIMENTI SE(scelta=ID\_SECONDA\_VOCE\_MENU)

ALLORA

pulire\_interfaccia()

Carica\_partita()

ALTRIMENTI SE(scelta=ID\_TERZA\_VOCE\_MENU)

ALLORA

Pulire\_interfaccia()

Modifica\_impostazioni()

ALTRIMENTI SE(scelta=ID\_QUARTA\_VOCE\_MENU)

ALLORA

Pulire\_interfaccia()

Regole\_gioco()

ALTRIMENTI

SE(scelta=ID\_QUINTA\_VOCE\_MENU)

ALLORA

pulire\_interfaccia()

Esci\_gioco=fine\_gioco()

FINE

FINE

FINCHE’(esci\_gioco==FALSO)

FINE

---

## FUNZIONE “fine\_gioco”

OUTPUT

NOME	DESCRIZIONE	TIPO
esci_gioco	Booleano che indica se l'utente ha deciso di uscire dal gioco	Booleano

PSEUDOCODICE  
esci\_gioco=VERO

**FUNZIONE “regole\_gioco”**

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle schermate	Intero>=ID_SCHERMATA_REGOLE e <ID_SCHERMATA_ISTRUZIONI
cambia_schermata	Tasto premuto dall’utente	Intero

PSEUDOCODICE  
i=ID\_SCHERMATA\_REGOLE  
ESEGUI
 SE(i=ID\_SCHERMATA\_REGOLE)
 ALLORA
 stampare\_regole\_gioco()
 cambia\_schermata=controllare\_tasto\_regole(i)
 SE(cambia\_schermata=FRECCIA\_DESTRA)
 ALLORA
 i=spostare\_avanti(i)
 FINE
 ALTRIMENTI SE(i=ID\_SCHERMATA\_ISTRUZIONI)
 ALLORA
 stampare\_istruzioni\_gioco()
 cambia\_schermata=controllare\_tasto\_regole(i)
 SE(cambia\_schermata=FRECCIA\_SINISTRA)
 ALLORA
 i=spostare\_indietro(i)
 FINE
 FINE
 FINE
 FINCHE’(cambia\_schermata≠ESC)
 FINE

**MODIFICA\_IMPOSTAZIONI**

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle schermate	Intero>=ID_SCHERMATA_REGOLE e <ID_SCHERMATA_ISTRUZIONI
cambia_schermata	Tasto premuto dall’utente	Intero
turno	Giocatore che inizia la partita	Giocatore
Conferma	Booleano che indica se l'utente ha confermato le modifiche	Booleano

PSEUDOCODICE  
turno=leggere\_imp()  
ESEGUI
 stampare\_impostazioni(turno)
 SE(turno=GIOCATORE)
 ALLORA
 Cambia\_schermata=FRECCIA\_SINISTRA
 Cambia\_schermata=controllare\_tasto\_cambia\_impostazioni(cambia\_schermata)
 SE(cambia\_schermata=FRECCIA\_DESTRA)

```

        ALLORA
            turno=PC
        ALTRIMENTI SE(cambia_schermata=INVIO)
            ALLORA
                conferma=confermare_operazione("Vuoi salvare le impostazioni?(s/n)")
                SE(conferma=VERO)
                    ALLORA
                        Modificare_imp(turno)
                    FINE
            FINE
        FINE
    ALTRIMENTI
        cambia_schermata=FRECCIA_DESTRA
        cambia_schermata=controllare_tasto_cambia_impostazioni(cambia_schermata)
        SE(cambia_schermata=FRECCIA_SINISTRA)
            ALLORA
                turno=GIOCATORE
            ALTRIMENTI SE(cambia_schermata=INVIO)
                ALLORA
                    conferma=confermare_operazione("Vuoi salvare le impostazioni?(s/n)")
                    SE(conferma=VERO)
                        ALLORA
                            Modificare_imp(turno)
                        FINE
                FINE
            FINE
        FINE
    FINCHE'(cambia_schermata≠INVIO AND cambia_schermata≠ESC)
    FINE

```

### **FUNZIONE "nuova\_partita"**

LAVORO

NOME	DESCRIZIONE	TIPO
turno_gioco	Giocatore e mossa che deve effettuare	Turno
campo_gioco	Campo di gioco	Campo

PSEUDOCODICE

```

campo_gioco=inizializzare()
turno_gioco.giocatore=leggere_imp()
turno_gioco.mossa=POSIZIONA
giocare_partita(campi_gioco,turno_gioco)

```

### **FUNZIONE "carica\_partita"**

LAVORO

NOME	DESCRIZIONE	TIPO
turno_gioco	Giocatore e mossa che deve effettuare	Turno
campo_gioco	Campo di gioco	Campo
Conferma	Booleano che indica se l'utente ha confermato l'operazione	Booleano
i	Contatore degli slot	Intero>=ID_PRIMO_SLOT e <NUM_SLOT
tasto	Tasto premuto dall'utente	Intero

PSEUDOCODICE

```

campo_gioco=inizializzare()
turno_gioco.giocatore=GIOCATORE

```

```

turno_gioco.mossa=POSIZIONA
i=ID_PRIMO_SLOT
SE(verificare_esistenza_file("slot1.txt")=FALSO AND
  verificare_esistenza_file("slot2.txt")=FALSO AND
  verificare_esistenza_file("slot3.txt")=FALSO AND
  verificare_esistenza_file("slot4.txt")=FALSO AND
  verificare_esistenza_file("slot5.txt")=FALSO AND
  verificare_esistenza_file("slot6.txt")=FALSO AND
  verificare_esistenza_file("slot7.txt")=FALSO AND
  verificare_esistenza_file("slot8.txt")=FALSO AND
  verificare_esistenza_file("slot9.txt")=FALSO AND
  verificare_esistenza_file("slot10.txt")=FALSO)
  ALLORA
    pulire_interfaccia()
    Stampa "Non ci sono partite salvate attualmente. Premi un tasto qualsiasi per tornare al menù"
    tasto=acquisire_tasto_qualsiasi()
  ALTRIMENTI
    ESEGUI
      ESEGUI
        Stampare_slot_partite(i)
        Tasto=controllare_tasto_menu_salvataggio(i)
        SE(tasto=FRECCIA_SU)
          ALLORA
            i=spostare_indietro(i)
          ALTRIMENTI SE(tasto=FRECCIA_GIU)
            i=spostare_avanti(i)
          ALTRIMENTI SE(tasto=ESC)
            Conferma=VERO
        FINE
      FINCHE'(tasto≠INVIO AND tasto≠ESC)
      FINE
    SE(tasto=INVIO)
      ALLORA
        SE(i=ID_PRIMO_SLOT)
          SE(verificare_esistenza_file("slot1.txt")=VERO)
            ALLORA
              SE(confermare_operazione("Sei sicuro di voler caricare
                questa partita? (s/n)")=VERO)
                ALLORA
                  campo_gioco=leggere_campo_gioco_partita
                    ("slot1.txt")
                  turno_gioco=leggere_turno_gioco_partita
                    ("slot1.txt")
                  conferma=VERO
                ALTRIMENTI
                  conferma=FALSO
              FINE
            FINE
          ALTRIMENTI SE(i=ID_SECONDO_SLOT)
            SE(verificare_esistenza_file("slot2.txt")=VERO)
              ALLORA
                SE(confermare_operazione("Sei sicuro di voler caricare
                  questa partita? (s/n)")=VERO)
                  ALLORA

```



```

                                campo_gioco=leggere_campo_gioco_partita
                                    ("slot2.txt")
                                turno_gioco=leggere_turno_gioco_partita
                                    ("slot2.txt")
                                conferma=VERO
                                ALTRIMENTI
                                conferma=FALSO
                            FINE
                        FINE
                    ALTRIMENTI SE(i=ID_TERZO_SLOT)
                        SE(verificare_esistenza_file("slot3.txt")=VERO)
                            ALLORA
                                SE(confermare_operazione("Sei sicuro di voler caricare
                                    questa partita? (s/n)")=VERO)
                                    ALLORA
                                        campo_gioco=leggere_campo_gioco_partita
                                            ("slot3.txt")
                                        turno_gioco=leggere_turno_gioco_partita
                                            ("slot3.txt")
                                        conferma=VERO
                                        ALTRIMENTI
                                        conferma=FALSO
                                    FINE
                                FINE
                            FINE
                        FINE
                    ALTRIMENTI SE(i=ID_QUARTO_SLOT)
                        SE(verificare_esistenza_file("slot4.txt")=VERO)
                            ALLORA
                                SE(confermare_operazione("Sei sicuro di voler caricare
                                    questa partita? (s/n)")=VERO)
                                    ALLORA
                                        campo_gioco=leggere_campo_gioco_partita
                                            ("slot4.txt")
                                        turno_gioco=leggere_turno_gioco_partita
                                            ("slot4.txt")
                                        conferma=VERO
                                        ALTRIMENTI
                                        conferma=FALSO
                                    FINE
                                FINE
                            FINE
                        FINE
                    ALTRIMENTI SE(i=ID_QUUINTO_SLOT)
                        SE(verificare_esistenza_file("slot5.txt")=VERO)
                            ALLORA
                                SE(confermare_operazione("Sei sicuro di voler caricare
                                    questa partita? (s/n)")=VERO)
                                    ALLORA
                                        campo_gioco=leggere_campo_gioco_partita
                                            ("slot5.txt")
                                        turno_gioco=leggere_turno_gioco_partita
                                            ("slot5.txt")
                                        conferma=VERO
                                        ALTRIMENTI
                                        conferma=FALSO
                                    FINE
                                FINE
                            FINE
                        FINE
                    FINE
                FINE
            FINE
        FINE
    FINE

```

```

ALTRIMENTI SE(i=ID_SESTO_SLOT)
    SE(verificare_esistenza_file("slot6.txt")=VERO)
        ALLORA
            SE(confermare_operazione("Sei sicuro di voler caricare
            questa partita? (s/n)")=VERO)
                ALLORA
                    campo_gioco=leggere_campo_gioco_partita
                    ("slot6.txt")
                    turno_gioco=leggere_turno_gioco_partita
                    ("slot6.txt")
                    conferma=VERO
                ALTRIMENTI
                    conferma=FALSO
            FINE
        FINE
    FINE
ALTRIMENTI SE(i=ID_SETTIMO_SLOT)
    SE(verificare_esistenza_file("slot7.txt")=VERO)
        ALLORA
            SE(confermare_operazione("Sei sicuro di voler caricare
            questa partita? (s/n)")=VERO)
                ALLORA
                    campo_gioco=leggere_campo_gioco_partita
                    ("slot7.txt")
                    turno_gioco=leggere_turno_gioco_partita
                    ("slot7.txt")
                    conferma=VERO
                ALTRIMENTI
                    conferma=FALSO
            FINE
        FINE
    FINE
ALTRIMENTI SE(i=ID_OTTAVO_SLOT)
    SE(verificare_esistenza_file("slot8.txt")=VERO)
        ALLORA
            SE(confermare_operazione("Sei sicuro di voler caricare
            questa partita? (s/n)")=VERO)
                ALLORA
                    campo_gioco=leggere_campo_gioco_partita
                    ("slot8.txt")
                    turno_gioco=leggere_turno_gioco_partita
                    ("slot8.txt")
                    conferma=VERO
                ALTRIMENTI
                    conferma=FALSO
            FINE
        FINE
    FINE
ALTRIMENTI SE(i=ID_NONO_SLOT)
    SE(verificare_esistenza_file("slot9.txt")=VERO)
        ALLORA
            SE(confermare_operazione("Sei sicuro di voler caricare
            questa partita? (s/n)")=VERO)
                ALLORA
                    campo_gioco=leggere_campo_gioco_partita
                    ("slot9.txt")
                    turno_gioco=leggere_turno_gioco_partita

```

```

("slot9.txt")
conferma=VERO
ALTRIMENTI
conferma=FALSO
FINE
FINE
ALTRIMENTI SE(i=ID_DECIMO_SLOT)
SE(verificare_esistenza_file("slot10.txt")=VERO)
ALLORA
SE(confermare_operazione("Sei sicuro di voler caricare
questa partita? (s/n)")=VERO)
ALLORA
campo_gioco=leggere_campo_gioco_partita
("slot10.txt")
turno_gioco=leggere_turno_gioco_partita
("slot10.txt")
conferma=VERO
ALTRIMENTI
conferma=FALSO
FINE
FINE
FINE
FINE
FINCHE'(conferma=FALSO)
FINE
SE(tasto=INVIO)
ALLORA
giocare_partita(campo_gioco,turno_gioco)
FINE
FINE

```

FINE

## MODULO "partita"

### FUNZIONE "controllare\_fine\_partita"

INPUT

NOME	DESCRIZIONE	TIPO
Campo_gioco	Campo di gioco	Campo
g	Giocatore di cui controllare le pedine	Giocatore

OUTPUT

NOME	DESCRIZIONE	TIPO
Fine	Booleano che indica se la partita è finita	Booleano

PSEUDOCODICE

```

fine=FALSO
SE(contare_num_pedine_serie(campo_gioco,g,CINQUE_PEDINE)=VERO)
ALLORA
fine=VERO

```

FINE

### FUNZIONE "contare\_num\_pedine\_serie\_orizzontali"

INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo
g	Giocatore di cui controllare le pedine	Giocatore
contare_num_pedine_consecutive	Numero di pedine da contare	Intero>ZERO_PEDINE <=CINQUE_PEDINE e

## LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO
j	Contatore delle colonne	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO
conta_pedine	Numero di pedine trovate	Intero>=ZERO_PEDINE e <=contare_num_pedine_consecutive
controllo_casella	Indica il tipo di pedine da cercare in base al giocatore	Casella

## OUTPUT

NOME	DESCRIZIONE	TIPO
pedine_consecutive	Booleano che indica se sono presenti le pedine consecutive	Booleano

## PSEUDOCODICE

pedine\_consecutive=FALSO

SE(contare\_num\_pedine\_consecutive>ZERO AND contare\_num\_pedine\_consecutive<=CINQUE\_PEDINE)

ALLORA

SE(g=GIOCATORE OR g=PC)

ALLORA

SE(g=GIOCATORE)

ALLORA

controllo\_casella=PEDINA\_GIOCATORE

ALTRIMENTI SE (g=PC)

ALLORA

controllo\_casella=PEDINA\_PC

FINE

conta\_pedine=ZERO\_PEDINE

i=ID\_PRIMA\_RIGA\_COLONNA

MENTRE(i<DIMENSIONE\_CAMPO AND pedine\_consecutive=FALSO)

j=ID\_PRIMA\_RIGA\_COLONNA

MENTRE(j<DIMENSIONE\_CAMPO AND pedine\_consecutive=FALSO)

SE(leggere(campo\_gioco,i,j)=controllo\_casella)

ALLORA

conta\_pedine=spostare\_avanti(conta\_pedine)

ALTRIMENTI

conta\_pedine=ZERO\_PEDINE

FINE

SE(conta\_pedine=contare\_num\_pedine\_consecutive)

ALLORA

pedine\_consecutive=VERO

FINE

j=spostare\_avanti(j)

FINE

conta\_pedine=ZERO\_PEDINE

i=spostare\_avanti(i)

FINE

FINE

FINE

## FUNZIONE "contare\_num\_pedine\_serie\_verticali"

INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo
g	Giocatore di cui controllare le pedine	Giocatore
contare_num_pedine_consecutive	Numero di pedine da contare	Intero>ZERO_PEDINE e <=CINQUE_PEDINE

#### LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO
j	Contatore delle colonne	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO
conta_pedine	Numero di pedine trovate	Intero>=ZERO_PEDINE e <=contare_num_pedine_consecutive
controllo_casella	Indica il tipo di pedine da cercare in base al giocatore	Casella

#### OUTPUT

NOME	DESCRIZIONE	TIPO
pedine_consecutive	Booleano che indica se sono presenti le pedine consecutive	Booleano

#### PSEUDOCODICE

pedine\_consecutive=FALSO

SE(contare\_num\_pedine\_consecutive>ZERO AND contare\_num\_pedine\_consecutive<=CINQUE\_PEDINE)

ALLORA

SE(g=GIOCATORE OR g=PC)

ALLORA

SE(g=GIOCATORE)

ALLORA

controllo\_casella=PEDINA\_GIOCATORE

ALTRIMENTI SE (g=PC)

ALLORA

controllo\_casella=PEDINA\_PC

FINE

conta\_pedine=ZERO\_PEDINE

i=ID\_PRIMA\_RIGA\_COLONNA

MENTRE(i<DIMENSIONE\_CAMPO AND pedine\_consecutive=FALSO)

j=ID\_PRIMA\_RIGA\_COLONNA

MENTRE(j<DIMENSIONE\_CAMPO AND pedine\_consecutive=FALSO)

SE(leggere(campo\_gioco,j,i)=controllo\_casella)

ALLORA

conta\_pedine=spostare\_avanti(conta\_pedine)

ALTRIMENTI

conta\_pedine=ZERO\_PEDINE

FINE

SE(conta\_pedine=contare\_num\_pedine\_consecutive)

ALLORA

pedine\_consecutive=VERO

FINE

j=spostare\_avanti(j)

FINE

conta\_pedine=ZERO\_PEDINE

i=spostare\_avanti(i)

FINE

FINE

FINE

## FUNZIONE “contare\_num\_pedine\_serie\_obliquo\_destra”

INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo
g	Giocatore di cui controllare le pedine	Giocatore
contare_num_pedine_consecutive	Numero di pedine da contare	Intero>ZERO_PEDINE e <=CINQUE_PEDINE

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO
j	Contatore delle colonne	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO
k	Contatore lunghezza campo	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO
conta_pedine	Numero di pedine trovate	Intero>=ZERO_PEDINE e <=contare_num_pedine_consecutive
controllo_casella	Indica il tipo di pedine da cercare in base al giocatore	Casella

OUTPUT

NOME	DESCRIZIONE	TIPO
pedine_consecutive	Booleano che indica se sono presenti le pedine consecutive	Booleano

PSEUDOCODICE

pedine\_consecutive=FALSO

SE(contare\_num\_pedine\_consecutive>ZERO AND contare\_num\_pedine\_consecutive<=CINQUE\_PEDINE)

ALLORA

SE(g=GIOCATORE OR g=PC)

ALLORA

SE(g=GIOCATORE)

ALLORA

controllo\_casella=PEDINA\_GIOCATORE

ALTRIMENTI SE (g=PC)

ALLORA

controllo\_casella=PEDINA\_PC

FINE

conta\_pedine=ZERO\_PEDINE

k=ID\_PRIMA\_RIGA\_COLONNA

MENTRE(k<DIMENSIONE\_CAMPO AND pedine\_consecutive=FALSO)

i=ID\_PRIMA\_RIGA\_COLONNA

j=k

MENTRE(i<spostare\_avanti(k) AND pedine\_consecutive=FALSO)

SE(leggere(campo\_gioco,i,j)=controllo\_casella)

ALLORA

conta\_pedine=spostare\_avanti(conta\_pedine)

ALTRIMENTI

conta\_pedine=ZERO\_PEDINE

FINE

SE(conta\_pedine=contare\_num\_pedine\_consecutive)

ALLORA

FINE

## INPUT

LAVORO

## OUTPUT

NOME	DESCRIZIONE	TIPO
------	-------------	------

pedine_consecutive	Booleano che indica se sono presenti le pedine consecutive	Booleano
--------------------	--	----------

# PSEUDOCODICE

pedine\_consecutive=FALSO

SE(contare\_num\_pedine\_consecutive>ZERO AND contare\_num\_pedine\_consecutive<=CINQUE\_PEDINE)

ALLORA

SE(g=GIOCATORE OR g=PC)

ALLORA

SE(g=GIOCATORE)

ALLORA

controllo\_casella=PEDINA\_GIOCATORE

ALTRIMENTI SE (g=PC)

ALLORA

controllo\_casella=PEDINA\_PC

FINE

conta\_pedine=ZERO\_PEDINE

k=ID\_PRIMA\_RIGA\_COLONNA

MENTRE(k<DIMENSIONE\_CAMPO AND pedine\_consecutive=FALSO)

i=ID\_PRIMA\_RIGA\_COLONNA

j=ID\_SESTA\_RIGA\_COLONNA-k

MENTRE(i<spostare\_avanti(k) AND pedine\_consecutive=FALSO)

SE(leggere(campo\_gioco,i,j)=controllo\_casella)

ALLORA

conta\_pedine=spostare\_avanti(conta\_pedine)

ALTRIMENTI

conta\_pedine=ZERO\_PEDINE

FINE

SE(conta\_pedine=contare\_num\_pedine\_consecutive)

ALLORA

pedine\_consecutive=VERO

FINE

i=spostare\_avanti(i)

j=spostare\_avanti(j)

FINE

conta\_pedine=ZERO\_PEDINE

k=spostare\_avanti(k)

FINE

conta\_pedine=ZERO\_PEDINE

k=ID\_SECONDA\_RIGA\_COLONNA

MENTRE(k<DIMENSIONE\_CAMPO AND pedine\_consecutive=FALSO)

i=ID\_PRIMA\_RIGA\_COLONNA+k

j=ID\_PRIMA\_RIGA\_COLONNA

MENTRE(i<DIMENSIONE\_CAMPO AND pedine\_consecutive=FALSO)

SE(leggere(campo\_gioco,i,j)=controllo\_casella)

ALLORA

conta\_pedine=spostare\_avanti(conta\_pedine)

ALTRIMENTI

conta\_pedine=ZERO\_PEDINE

FINE

SE(conta\_pedine=contare\_num\_pedine\_consecutive)

ALLORA

pedine\_consecutive=VERO

FINE



i=spostare\_avanti(i)

j=spostare\_avanti(j)

FINE

conta\_pedine=ZERO\_PEDINE

k=spostare\_avanti(k)

FINE

FINE

FINE

### FUNZIONE “contare\_num\_pedine\_serie”

INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo
g	Giocatore di cui controllare le pedine	Giocatore
contare_num_pedine_consecutive	Numero di pedine da contare	Intero>ZERO_PEDINE e <=CINQUE_PEDINE

OUTPUT

NOME	DESCRIZIONE	TIPO
pedine_consecutive	Booleano che indica se sono presenti le pedine consecutive	Booleano

PSEUDOCODICE

pedine\_consecutive=FALSO

SE(contare\_num\_pedine\_consecutive>ZERO\_PEDINE AND contare\_num\_pedine\_consecutive<=CINQUE\_PEDINE)

ALLORA

SE(g=PC OR g=GIOCATORE)

ALLORA

SE

((contare\_num\_pedine\_serie\_orizzontali(campo\_gioco,g,contare\_num\_pedine\_consecutive)=VERO)OR

(contare\_num\_pedine\_serie\_verticali(campo\_gioco,g,contare\_num\_pedine\_consecutive)=VERO)OR

(contare\_num\_pedine\_serie\_obliquo\_destra(campo\_gioco,g,contare\_num\_pedine\_consecutive)=VERO)OR

(contare\_num\_pedine\_serie\_obliquo\_sinistra(campo\_gioco,g,contare\_num\_pedine\_consecutive)=VERO))

ALLORA

pedine\_consecutive=VERO

FINE

FINE

FINE

### FUNZIONE “giocare\_partita”

INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo
turno_gioco	Turno di gioco	Turno

LAVORO

NOME	DESCRIZIONE	TIPO
tasto	Tasto premuto dall’utente	Intero
notifica	Messaggio che viene mostrato in fondo alla schermata di gioco	Stringa
campo_temporaneo	Campo temporaneo uguale al campo di gioco utilizzato dal PC per decidere quale campo ruotare	Campo
mossa	Mossa di rotazione campo del PC	Mossa_rotazione
coordinata	Casella dove posizionare la pedina del PC/UTENTE	Coordinata



```

turno_gioco.mossa=RUOTA
FINE
FINE
FINE
ALTRIMENTI
stampare_campo(campo_gioco,ERRORE,ERRORE)
stampare_scelta_rotazione(z)
SE(z>=ID_PRIMA_SEZIONE AND z<NUM_SOTTOCAMPI)
    tasto=controllare_tasto_campo_da_ruotare(z)
    SE(tasto≠ESC)
        SE(tasto=FRECCIA_DESTRA)
            z=spostare_avanti(z)
        ALTRIMENTI SE(tasto=FRECCIA_SINISTRA)
            z=spostare_indietro(z)
        ALTRIMENTI SE(tasto=INVIO)
            campo_da_ruotare=z
            z=ID_RUOTA_SX
    FINE
    FINE
    ALTRIMENTI SE(z>=ID_RUOTA_SX AND z<=ID_RUOTA_DX)
        tasto=controllare_tasto_verso_rotazione(z)
        SE(tasto=FRECCIA_DESTRA)
            z=spostare_avanti(z)
        ALTRIMENTI SE(tasto=FRECCIA_SINISTRA)
            z=spostare_indietro(z)
        ALTRIMENTI SE(tasto=INVIO)
            SE(z=ID_RUOTA_SX)
                verso=SX
            ALTRIMENTI SE(z=ID_RUOTA_DX)
                verso=DX
        FINE
        ruotare_campo(campo_gioco,verso,
                        campo_da_ruotare)
        z=ID_PRIMA_SEZIONE
        turno_gioco.giocatore=PC
        turno_gioco.mossa=POSIZIONA
    FINE
    FINE
    FINE
    ALTRIMENTI SE(turno_gioco.giocatore=PC)
        SE(turno_gioco.mossa=POSIZIONA)
            campo_temporaneo=copiare_campo(campo_gioco)
            stampare_campo(campo_temporaneo,ERRORE,ERRORE)
            stampare_scelta_rotazione(ERRORE)
            mossa=trovare_miglior_ruotato(campo_temporaneo)
            ruotare_campo(campo_temporaneo,mossa.verso,mossa.num_campo)
            coordinata=scegliere_coppia(campo_temporaneo,GIOCATORE)
            SE(coordinata.x≠ERRORE AND coordinata.y≠ERRORE)
                Posizionare_pedina(campo_temporaneo,coordinata.x,
                                    coordinata.y,turno_gioco.giocatore)
        ALTRIMENTI
            coordinata=scegliere_coppia(campo_temporaneo,PC)
            SE(coordinata.x≠ERRORE AND coordinata.y≠ERRORE)
                Posizionare_pedina(campo_temporaneo,coordinata.x,

```

```

                                coordinata.y,turno_gioco.giocatore)
                                ALTRIMENTI
                                coordinata=posizionare_per_priorità(campo_gioco)
                                posizionare_pedina(campo_temporaneo,coordinata.x
                                coordinata.y,turno_gioco.giocatore)
                                FINE
                                FINE
                                turno_gioco.mossa=RUOTA
                                ALTRIMENTI
                                SE(mossa.verso=DX)
                                ruotare_campo(campo_temporaneo,SX,mossa.num_campo)
                                ALTRIMENTI
                                ruotare_campo(campo_temporaneo,DX,mossa.num_campo)
                                FINE
                                stampare_campo(campo_temporaneo,ERRORE,ERRORE)
                                stampare_scelta_rotazione(ERRORE)
                                ruotare_campo(campo_temporaneo,mossa.verso,mossa.num_campo)
                                campo_gioco=copiare_campo(campo_temporaneo)
                                turno_gioco.giocatore=GIOCATORE
                                turno_gioco.mossa=POSIZIONA
                                FINE
                                FINE
                                FINCHE'(controllare_fine_partita(campo_gioco,GIOCATORE)=FALSO AND
                                controllare_fine_partita(campo_gioco,PC)=FALSO AND
                                controllare_campo_pieno(campo_gioco)=FALSO AND tasto≠ESC)
                                FINE
                                SE(controllare_campo_pieno(campo_gioco)=VERO OR
                                (controllare_fine_partita(campo_gioco,GIOCATORE)=VERO AND
                                controllare_fine_partita(campo_gioco,PC)=VERO))
                                Stampa "La partita è finita in pareggio, premi un tasto per tornare al menu"
                                tasto=acquisire_tasto_qualsiasi()
                                ALTRIMENTI SE(controllare_fine_partita(campo_gioco,PC)=VERO)
                                Stampa "Il PC ha vinto la partita, premi un tasto per tornare al menu"
                                tasto=acquisire_tasto_qualsiasi()
                                ALTRIMENTI SE(controllare_fine_partita(campo_gioco,GIOCATORE)=VERO)
                                Stampa "Hai vinto la partita, premi un tasto per tornare al menu"
                                ALTRIMENTI SE(tasto=ESC)
                                SE(confermare_operazione("Vuoi salvare la partita? (s/n)")=VERO)
                                Salvare_partita(campo_gioco,turno_gioco)
                                FINE
                                FINE
                                FINE

```

FINE

## FUNZIONE "controllare\_campo\_pieno"

INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo

LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore delle righe	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO
j	Contatore delle colonne	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO

# OUTPUT

NOME	DESCRIZIONE	TIPO
pieno	Booleano che indica se il campo è pieno	Booleano

# PSEUDOCODICE

i=ID\_PRIMA\_RIGA\_COLONNA

pieno=VERO

ESEGUI

j=ID\_PRIMA\_RIGA\_COLONNA

ESEGUI

SE(leggere(campo\_gioco,i,j)=VUOTA)

pieno=FALSO

FINE

j=spostare\_avanti(j)

FINCHE'(j<DIMENSIONE\_CAMPO AND pieno≠FALSO)

FINE

i=spostare\_avanti(i)

FINCHE'(i<DIMENSIONE\_CAMPO AND pieno≠FALSO)

FINE

# FUNZIONE “controllare\_casella\_libera”

## INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco	Campo
x	Riga del campo dove si vuole controllare la casella	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO
y	Colonna del campo dove si vuole controllare la casella	Intero>=ID_PRIMA_RIGA_COLONNA e <DIMENSIONE CAMPO

# OUTPUT

NOME	DESCRIZIONE	TIPO
vuota	Booleano che indica se la casella scelta è vuota	Booleano

# PSEUDOCODICE

vuota=VERO

SE(x>=ID\_PRIMA\_RIGA\_COLONNA AND x<DIMENSIONE\_CAMPO AND y>=ID\_PRIMA\_RIGA\_COLONNA AND y<DIMENSIONE CAMPO)

ALLORA

SE(leggere(campo\_gioco,x,y)=PEDINA\_GIOCATORE OR leggere(campo\_gioco,x,y)=PEDINA\_PC)

ALLORA

vuota=FALSO

FINE

FINE

# FUNZIONE “ruotare\_campo”

## INPUT

NOME	DESCRIZIONE	TIPO
campo_di_gioco	Campo di gioco	Campo
verso	Verso in cui ruotare il sottocampo	Verso_rotazione
num_campo	Sottocampo da ruotare	Intero>=ID_PRIMA_SEZIONE e <NUM_SOTTOCAMPI

# LAVORO

NOME	DESCRIZIONE	TIPO
------	-------------	------

i	Contatore delle ripetizioni necessarie	Intero>=0 e <RIPETIZIONI_NECESSARIE_ROTAZIONE
val_temp	Valore della prima casella del sottocampo	Casella
casella_di_partenza	Coordinata della prima casella del sottocampo	Coordinata

## OUTPUT

NOME	DESCRIZIONE	TIPO
campo_di_gioco	Campo di gioco con il sottocampo ruotato	Campo

## PSEUDOCODICE

SE(num\_campo>=ID\_PRIMA\_SEZIONE AND num\_campo<NUM\_SOTTOCAMPI)

SE(verso=DX OR verso=SX)

SE(num\_campo=ID\_PRIMA\_SEZIONE)

casella\_di\_partenza.x=PRIMA\_RIGA\_SEZ\_0

casella\_di\_partenza.y=PRIMA\_COLONNA\_SEZ\_0

ALTRIMENTI SE(num\_campo=ID\_SECONDA\_SEZIONE)

casella\_di\_partenza.x=PRIMA\_RIGA\_SEZ\_1

casella\_di\_partenza.y=PRIMA\_COLONNA\_SEZ\_1

ALTRIMENTI SE(num\_campo=ID\_TERZA\_SEZIONE)

casella\_di\_partenza.x=PRIMA\_RIGA\_SEZ\_2

casella\_di\_partenza.y=PRIMA\_COLONNA\_SEZ\_2

ALTRIMENTI

casella\_di\_partenza.x=PRIMA\_RIGA\_SEZ\_3

casella\_di\_partenza.y=PRIMA\_COLONNA\_SEZ\_3

FINE

SE(verso=SX)

i=0

MENTRE(i<RIPETIZIONI\_NECESSARIE\_ROTAZIONE)

val\_temp=leggere(campo\_di\_gioco,casella\_di\_partenza.x, casella\_di\_partenza.y)

scrivere(campo\_di\_gioco, casella\_di\_partenza.x, casella\_di\_partenza.y,

leggere(campo\_di\_gioco, casella\_di\_partenza.x,  
casella\_di\_partenza.y+DISTANZA\_1\_PRIMA\_POSIZIONE))

scrivere(campo\_di\_gioco, casella\_di\_partenza.x, casella\_di\_partenza.y+  
DISTANZA\_1\_PRIMA\_POSIZIONE,

leggere(campo\_di\_gioco, casella\_di\_partenza.x,  
casella\_di\_partenza.y+DISTANZA\_2\_PRIMA\_POSIZIONE))

scrivere(campo\_di\_gioco, casella\_di\_partenza.x, casella\_di\_partenza.y+  
DISTANZA\_2\_PRIMA\_POSIZIONE,

leggere(campo\_di\_gioco, casella\_di\_partenza.x+DISTANZA\_1\_PRIMA\_POSIZIONE,  
casella\_di\_partenza.y+DISTANZA\_2\_PRIMA\_POSIZIONE))

scrivere(campo\_di\_gioco, casella\_di\_partenza.x+DISTANZA\_1\_PRIMA\_POSIZIONE,  
casella\_di\_partenza.y+DISTANZA\_2\_PRIMA\_POSIZIONE,

leggere(campo\_di\_gioco, casella\_di\_partenza.x+DISTANZA\_2\_PRIMA\_POSIZIONE,  
casella\_di\_partenza.y+DISTANZA\_2\_PRIMA\_POSIZIONE))

scrivere(campo\_di\_gioco, casella\_di\_partenza.x+DISTANZA\_2\_PRIMA\_POSIZIONE,  
casella\_di\_partenza.y+DISTANZA\_2\_PRIMA\_POSIZIONE,

leggere(campo\_di\_gioco, casella\_di\_partenza.x+DISTANZA\_2\_PRIMA\_POSIZIONE,  
casella\_di\_partenza.y+DISTANZA\_1\_PRIMA\_POSIZIONE))

scrivere(campo\_di\_gioco, casella\_di\_partenza.x+DISTANZA\_2\_PRIMA\_POSIZIONE,  
casella\_di\_partenza.y+DISTANZA\_1PRIMA\_POSIZIONE,

leggere(campo\_di\_gioco, casella\_di\_partenza.x+DISTANZA\_2\_PRIMA\_POSIZIONE,  
casella\_di\_partenza.y))

```

scrivere(campo_di_gioco, casella_di_partenza.x+DISTANZA_2_PRIMA_POSIZIONE,
        casella_di_partenza.y,
        leggere(campo_di_gioco, casella_di_partenza.x+DISTANZA_1_PRIMA_POSIZIONE,
        casella_di_partenza.y))
scrivere(campo_di_gioco, casella_di_partenza.x+DISTANZA_1_PRIMA_POSIZIONE,
        casella_di_partenza.y,
        val_temp)
i=spostare_avanti(i)
    FINE
ALTRIMENTI
    i=0
    MENTRE(i<RIPETIZIONI_NECESSARIE_ROTAZIONE)
        val_temp=leggere(campo_di_gioco,casella_di_partenza.x, casella_di_partenza.y)
        scrivere(campo_di_gioco, casella_di_partenza.x, casella_di_partenza.y,
            leggere(campo_di_gioco, casella_di_partenza.x+
                DISTANZA_1_PRIMA_POSIZIONE,casella_di_partenza.y))
        scrivere(campo_di_gioco, casella_di_partenza.x+ DISTANZA_1_PRIMA_POSIZIONE,
            casella_di_partenza.y,
            leggere(campo_di_gioco, casella_di_partenza.x+
                DISTANZA_2PRIMA_POSIZIONE,casella_di_partenza.y))
        scrivere(campo_di_gioco, casella_di_partenza.x+DISTANZA_2_PRIMA_POSIZIONE,
            casella_di_partenza.y,
            leggere(campo_di_gioco, casella_di_partenza.x+
                DISTANZA_2_PRIMA_POSIZIONE,casella_di_partenza.y
                +DISTANZA_1_PRIMA_POSIZIONE))
        scrivere(campo_di_gioco, casella_di_partenza.x+DISTANZA_2_PRIMA_POSIZIONE,
            casella_di_partenza.y+DISTANZA_1_PRIMA_POSIZIONE,
            leggere(campo_di_gioco, casella_di_partenza.x+DISTANZA_2_PRIMA_POSIZIONE,
            casella_di_partenza.y+DISTANZA_2_PRIMA_POSIZIONE))
        scrivere(campo_di_gioco, casella_di_partenza.x+DISTANZA_2_PRIMA_POSIZIONE,
            casella_di_partenza.y+DISTANZA_2_PRIMA_POSIZIONE,
            leggere(campo_di_gioco, casella_di_partenza.x+DISTANZA_1_PRIMA_POSIZIONE,
            casella_di_partenza.y+DISTANZA_2_PRIMA_POSIZIONE))
        scrivere(campo_di_gioco, casella_di_partenza.x+DISTANZA_1_PRIMA_POSIZIONE,
            casella_di_partenza.y+DISTANZA_2_PRIMA_POSIZIONE,
            leggere(campo_di_gioco, casella_di_partenza.x,
            casella_di_partenza.y+DISTANZA_2_PRIMA_POSIZIONE))
        scrivere(campo_di_gioco, casella_di_partenza.x,
            casella_di_partenza.y+DISTANZA_2_PRIMA_POSIZIONE,
            leggere(campo_di_gioco, casella_di_partenza.x,
            casella_di_partenza.y+DISTANZA_1_PRIMA_POSIZIONE))
        scrivere(campo_di_gioco, casella_di_partenza.x,
            casella_di_partenza.y+DISTANZA_1_PRIMA_POSIZIONE,
            val_temp)
        i=spostare_avanti(i)
    FINE
    FINE
    FINE
    FINE

```

## FUNZIONE “salvare\_partita”

INPUT

NOME	DESCRIZIONE	TIPO
campo_gioco	Campo di gioco da salvare	Campo
Turno_gioco	Turno di gioco da salvare	turno

## LAVORO

NOME	DESCRIZIONE	TIPO
i	Contatore degli slot	Intero>=ID_PRIMO_SLOT <NUM_SLOT e
tasto	Tasto premuto dall'utente	Intero
Conferma	Booleano che indica che l'utente ha confermato il salvataggio	Booleano

## OUTPUT

NOME	DESCRIZIONE	TIPO
FILE	File della partita salvata	FILE

## PSEUDOCODICE

```

SE((turno_gioco.giocatore=GIOCATORE OR turno_gioco.giocatore=PC) AND (turno_gioco.mossa=RUOTA OR
turno_gioco.mossa=POSIZIONA)
    i=ID_PRIMO_SLOT
    ESEGUI
        ESEGUI
            stampare_slot_partite(i)
            tast=controllare_tasto_menu_salvataggio(i)
            SE(tasto=FRECCIA_SU)
                l=spostare_indietro(i)
            ALTRIMENTI SE(tasto=FRECCIA_GIU)
                l=spostare_avanti(i)
            ALTRIMENTI SE(tasto=ESC)
                conferma=VERO
            FINE
        FINCHE' (tasto≠INVIO AND tasto≠ESC)
        FINE
    SE(tasto=INVIO)
        SE(i=ID_PRIMO_SLOT)
            SE(verificare_esistenza_file("slot1.txt")=VERO)
                SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
                    =VERO)
                    Scrivere_file_partite(turno_gioco,campo_gioco,"slot1.txt")
                    conferma=VERO
                ALTRIMENTI
                    Conferma=FALSO
                FINE
            ALTRIMENTI
                creare_file("slot1.txt")
                Scrivere_file_partite(turno_gioco,campo_gioco,"slot1.txt")
                conferma=VERO
            FINE
        ALTRIMENTI SE(i=ID_SECONDO_SLOT)
            SE(verificare_esistenza_file("slot2.txt")=VERO)
                SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
                    =VERO)
                    Scrivere_file_partite(turno_gioco,campo_gioco,"slot2.txt")
                    conferma=VERO
                ALTRIMENTI
                    Conferma=FALSO
                FINE
            FINE

```



```

    ALTRIMENTI
        creare_file("slot2.txt")
        Scrivere_file_partite(turno_gioco,campo_gioco,"slot2.txt")
        conferma=VERO
    FINE
ALTRIMENTI SE(i=ID_TERZO_SLOT)
    SE(verificare_esistenza_file("slot3.txt")=VERO)
        SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
            =VERO)
            Scrivere_file_partite(turno_gioco,campo_gioco,"slot3.txt")
            conferma=VERO
        ALTRIMENTI
            Conferma=FALSO
        FINE
    ALTRIMENTI
        creare_file("slot3.txt")
        Scrivere_file_partite(turno_gioco,campo_gioco,"slot3.txt")
        conferma=VERO
    FINE
ALTRIMENTI SE(i=ID_QUARTO_SLOT)
    SE(verificare_esistenza_file("slot4.txt")=VERO)
        SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
            =VERO)
            Scrivere_file_partite(turno_gioco,campo_gioco,"slot4.txt")
            conferma=VERO
        ALTRIMENTI
            Conferma=FALSO
        FINE
    ALTRIMENTI
        creare_file("slot4.txt")
        Scrivere_file_partite(turno_gioco,campo_gioco,"slot4.txt")
        conferma=VERO
    FINE
ALTRIMENTI SE(i=ID_QUINTO_SLOT)
    SE(verificare_esistenza_file("slot5.txt")=VERO)
        SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
            =VERO)
            Scrivere_file_partite(turno_gioco,campo_gioco,"slot5.txt")
            conferma=VERO
        ALTRIMENTI
            Conferma=FALSO
        FINE
    ALTRIMENTI
        creare_file("slot5.txt")
        Scrivere_file_partite(turno_gioco,campo_gioco,"slot5.txt")
        conferma=VERO
    FINE
ALTRIMENTI SE(i=ID_SESTO_SLOT)
    SE(verificare_esistenza_file("slot6.txt")=VERO)
        SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
            =VERO)
            Scrivere_file_partite(turno_gioco,campo_gioco,"slot6.txt")
            conferma=VERO
        ALTRIMENTI

```

```

Conferma=FALSO
    FINE
ALTRIMENTI
    creare_file("slot6.txt")
    Scrivere_file_partite(turno_gioco,campo_gioco,"slot6.txt")
    conferma=VERO
    FINE
ALTRIMENTI SE(i=ID_SETTIMO_SLOT)
    SE(verificare_esistenza_file("slot7.txt")=VERO)
        SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
            =VERO)
            Scrivere_file_partite(turno_gioco,campo_gioco,"slot7.txt")
            conferma=VERO
        ALTRIMENTI
            Conferma=FALSO
        FINE
    ALTRIMENTI
        creare_file("slot7.txt")
        Scrivere_file_partite(turno_gioco,campo_gioco,"slot7.txt")
        conferma=VERO
    FINE
ALTRIMENTI SE(i=ID_OTTAVO_SLOT)
    SE(verificare_esistenza_file("slot8.txt")=VERO)
        SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
            =VERO)
            Scrivere_file_partite(turno_gioco,campo_gioco,"slot8.txt")
            conferma=VERO
        ALTRIMENTI
            Conferma=FALSO
        FINE
    ALTRIMENTI
        creare_file("slot8.txt")
        Scrivere_file_partite(turno_gioco,campo_gioco,"slot8.txt")
        conferma=VERO
    FINE
ALTRIMENTI SE(i=ID_NONO_SLOT)
    SE(verificare_esistenza_file("slot9.txt")=VERO)
        SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
            =VERO)
            Scrivere_file_partite(turno_gioco,campo_gioco,"slot9.txt")
            conferma=VERO
        ALTRIMENTI
            Conferma=FALSO
        FINE
    ALTRIMENTI
        creare_file("slot9.txt")
        Scrivere_file_partite(turno_gioco,campo_gioco,"slot9.txt")
        conferma=VERO
    FINE
ALTRIMENTI SE(i=ID_DECIMO_SLOT)
    SE(verificare_esistenza_file("slot10.txt")=VERO)
        SE(confermare_operazione("E' già presente una partita salvata,sovrascriverla?(s/n)")
            =VERO)
            Scrivere_file_partite(turno_gioco,campo_gioco,"slot10.txt")

```

```

                                conferma=VERO
                                ALTRIMENTI
                                    Conferma=FALSO
                                    FINE
                                ALTRIMENTI
                                    creare_file("slot10.txt")
                                    Scrivere_file_partite(turno_gioco,campo_gioco,"slot10txt")
                                    conferma=VERO
                                FINE
                            FINE
                        FINE
                    FINCHE'(conferma=FALSO)
                FINE
            FINE

```

## MODULO "regolegioco"

### FUNZIONE "stampare\_regole\_gioco"

OUTPUT

NOME	DESCRIZIONE	TIPO
REGOLE DE GIOCO	Interfaccia che mostra le regole del Gioco	TESTO

PSEUDOCODICE

Pulire\_interfaccia()

Stampa "Pentago è un gioco per 2 giocatori. Si gioca su un campo di 6x6 caselle suddiviso in 4 sottocampi 3x3. I due giocatori devono posizionare a turno una pedina in unacasella libera e, successivamente, deve ruotare uno dei sottocampi in senso orario o antiorario di 90 gradi. Vince il giocatore che riesce a posizionare 5 pedine del suo colore in una fila continua, in orizzontale, verticale o obliquo, prima o dopo la rotazione. Se uno dei due giocatori fa una mossa che consente ad entrambi i giocatori di comporre una fila da 5, la partita finisce in pareggio. Anche nel caso in cui tutte le caselle del campo vengono riempite e nessuno dei giocatori riesce a fare una fila da 5, la partita finisce in pareggio."

Stampa "ESC MENU' ISTRUZIONI GIOCO »"

### FUNZIONE "stampare\_istruzioni\_gioco"

OUTPUT

NOME	DESCRIZIONE	TIPO
ISTRUZIONI DI GIOCO	Interfaccia che mostra le istruzioni per giocare	TESTO

PSEUDOCODICE

Pulire\_interfaccia()

Stampa "Per giocare al computer, l'inserimento della pedina avviene mediante l'utilizzo delle frecce direzionali dell'INVIO, che serve per inserire la pedina nella posizione selezionata. Durante la partita, il giocatore può premere ESC per scegliere di salvare la partita. La scelta del sottocampo da ruotare e del verso in cui ruotarlo avviene sempre tramite l'utilizzo delle frecce direzionali e dell'invio per selezionare il sottocampo e il verso di rotazione. Il sottocampo in alto a sinistra è il num. 1, quello in alto a destra è il num. 2, quello in basso a sinistra il num. 3 e quello in basso a destra il num. 4. Anche negli altri menù del gioco, per effettuare la propria scelta si utilizza il tasto INVIO, il tasto ESC per tornare al menù principale e le frecce direzionali per muoversi nel menù.

In alcuni casi, può venire richiesto al giocatore di confermare l'operazione appena eseguita. Il giocatore premerà quindi 's' per confermare o 'n' per annullare."

Stampa "« REGOLE GIOCO ESC MENU'"