

## Main()

### Dati di input

Nome	Descrizione	Vincoli
scelta_menu	scelta dell'utente	intero [0-5]

### Dati di output

Nome	Descrizione	Vincoli
scelta_menu	scelta dell'utente	intero [0-5]
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]
Record_file	file sul quale ci sono i record dei giocatori	Binario [pieno,vuoto]

Permette di modificare impostazioni

Permette di visualizzare i record

Permette di caricare una partita

Permette di salvare una partita

Inizio

    Inizializza\_gioco()

    Esegui

        Menu()

        leggere scelta\_menu

        Se (scelta\_menu diverso da 0)

            ALLORA

                SE(scelta\_menu=1)

                    Allora

                        Nuova\_partita()

                ALTRIMENTI

                    SE(scelta\_menu=2)

                        ALLORA

                            Carica\_Partita()

                        ALTRIMENTI

                            SE (scelta\_menu=3)

                                ALLORA

                                    Salva\_partita()

                                ALTRIMENTI

                                    SE(scelta\_menu=4)

  ALLORA

  Impostazioni()

  ALTRIMENTI

  SE (scelta\_menu=5)

  ALLORA

  Record()

  FINE SE

  FINE SE

  FINE SE

                                    FINE SE

                                FINE SE

        FINCHE (scelta\_menu diverso da 0)

    FINE

## Inizializzare\_gioco()

### Dati di input

Nome	Descrizione	Vincoli
Record_file	file sul quale ci sono i record dei giocatori	Binario [esiste,non esiste]
Partite_salvate	file sul quale ci sono le partite salvate	Binario [esiste,non esiste]
DEFAULT	costante che contiene il numero di giocatori base	intero 2

### Dati di output

Nome	Descrizione	Vincoli
Record_file	file sul quale ci sono i record dei giocatori	Binario [pieno,vuoto]
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]

### Dati di Lavoro

Nome	Descrizione	Vincoli
DEFINIRE STRUTTURA Record	Struttura che serve a contenere tutti i dati per stabilire i record	
intero n_giocatore		
intero p_vinte		
intero p_giocate		
fine definizione R		

```
DEFINIRE STRUTTURA Partite_salvate Struttura che serve a contenere
carattere nome_part tutti i dati da salvare su un file
intero posg1 per far riprendere una partita
intero posg2
intero posg3
intero posg4
intero n
intero turno
fine definizione Ps
i indice
```

INIZIO

i=0

R record

Ps Par\_sal

n\_partecipanti=DEFAULT

Aprire il file "Record\_file"

SE (il file non esiste)

ALLORA

creare il file "Record\_file"

MENTRE (i<GIOCATORI\_MAX)

record.n\_giocatore=i+1

record.p\_vinte=0

record.p\_giocate=0

i=i+1

Scrivere nel file nella posizione corrente il record

FINE MENTRE

FINE SE

Chiudere il file "Record\_file"

Aprire il file "Partite\_salvate"

SE (il file non esiste)

ALLORA

creare il file "Partite\_salvate"

Fine SE

Chiudere il file "Partite\_salvate"

Fine

## Menu()

### Dati di output

Nome	Descrizione	Vincoli
	stampa interfaccia menu a video	

Inizio

Scrivere "SCALE E SERPENTI"

Scrivere "[1].NUOVA PARTITA"

Scrivere "[2].CARICA PARTITA"

Scrivere "[3].SALVA PARTITA"

Scrivere "[4].IMPOSTAZIONI"

Scrivere "[5].RECORD"

Scrivere "[0].Esci"

Scrivere "Fai la tua scelta premendo un tasto tra i numeri indicati:"

Fine

## Nuova\_partita()

### Dati di input

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
vincitore	indica il vincitore	intero da [1-4]
num	indica il turno del giocatore	intero da [0-3]
n_partecipanti	indica il numero di partecipanti	intero da [2-4]
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]
Record_file	file sul quale ci sono i record dei giocatori	Binario [pieno,vuoto]
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi	4

### Dati di output

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
vincitore	indica il vincitore	intero da [1-4]
Record_file	file sul quale ci sono i record dei giocatori	Binario [pieno,vuoto]

INIZIO

Inizializzare\_partita()

Spostare\_pedina()

SE (vincitore diverso da 0)

Salvare\_risultati()

FINE SE

FINE

## Grafica\_Nuova\_partita()

### Dati di input

Nome	Descrizione	Vincoli
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]
DIM_CAMPO	Dimensione del campo da gioco	

### Dati di Lavoro

Nome	Descrizione	Vincoli
x	coordinata stampa a video x	
y	coordinata stampa a video y	
cont	contatore	
num	Variabile locale che è una stringa	stringa di caratteri con max 4

### Dati di output

Nome	Descrizione	Vincoli
	stampa a video il campo da gioco	

INIZIO

x=1

y=1

cont=100

Scrivere bordi di tutto il campo da gioco

(Qui stamperà i numeri da 100 a 91, da 80 a 71, da 60 a 51..., da 20 a 11. )

MENTRE(y<=29)

x=1

MENTRE(x<=46)

Scrivere il numero nella casella di coordinate x,y  
decrementa cont di 1

x=x+5

FINE MENTRE

cont=cont-10

y=y+6

FINE MENTRE

x=1;

y=4;

cont=81

(Qui stamperà i numeri da 90 a 81, da 70 a 61, da 50 a 41..., da 10 a 1)

MENTRE(y<=29)

x=1;

MENTRE(x<=46)

Scrivere il numero nella casella di coordinate x,y

x=x+5

cont=cont+1

FINE MENTRE

cont=cont-30

y=y+6

FINE MENTRE

SCRIVERE "LEGENDA,TASTI,TORNA AL MENU,INVIO LANCIO DADO,SIMBOLI"(TUTTO A DESTRA DOPO IL CAMPO"

SCRIVERE "GIOCATORE 1, GIOCATORE 2, GIOCATORE 3,GIOCATORE 4"(TUTTO A DESTRA DOPO IL CAMPO"

SCRIVERE le scale e i serpenti predefiniti nelle apposite celle di coordinate predefinite

```

Scrivere "PARTECIPANTI" (TUTTO A DESTRA DOPO IL CAMPO)
SE (n_partecipanti=2)
    ALLORA
        Scrivere " SIMBOLO PEDINA 1,SIMBOLO PEDINA 2"
ALTRIMENTI
    SE (n_partecipanti=3)
        ALLORA
            Scrivere " SIMBOLO PEDINA 1,SIMBOLO PEDINA 2,SIMBOLO PEDINA 3"
        ALTRIMENTI
            SE (n_partecipanti=4)
                ALLORA
                    Scrivere "SIMBOLO PEDINA 1,SIMBOLO PEDINA 2,SIMBOLO PEDINA 3, SIMBOLO PEDINA 4"
                fine SE
            fine SE
        fine SE
    fine SE
FINE

```

## Inizializzare\_partita()

### Dati di input

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]
num	indica il turno del giocatore	intero da [0-3]
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi	4

### Dati di output

Nome	Descrizione	Vincoli
num	indica il turno del giocatore	intero da [0-3]
vincitore	indica il vincitore	intero da [1-4]
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
	stampa a video il campo da gioco	

### INIZIO

```

    num=0
    MENTRE (num<n_partecipanti)
        cella_g[num]=0
        num=num+1
    FINE MENTRE
    num=0
    vincitore=0
    Grafica_Nuova_partita()
FINE

```

## Spostare\_pedina()

### Dati di input

Nome	Descrizione	Vincoli
scelta_tasto	tasto scelto dall'utente	intero [INVIO,ESC]
dado	variabile locale che contiene il valore casuale generato	intero[1,MAXDADO]
MAXDADO	variabile che contiene il max della generazione	intero positivo
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
num	indica il turno del giocatore	intero da [0-3]
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]
vincitore	indica il vincitore	intero da [1-4]
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi	4

### Dati di output

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
num	indica il turno del giocatore	intero da [0-3]
vincitore	indica il vincitore	intero da [1-4]
	sposta la pedina graficamente	

### Dati di Lavoro

Nome	Descrizione	Vincoli
dado	variabile locale che contiene il valore casuale generato	intero
ESC	costante globale che contiene il numero ascii per riconoscere il valore associato al tasto ESC	intero 27

### INIZIO

intero dado=0

scelta\_tasto=0

ESEGUI

scelta\_tasto=Turno\_giocatore()

SE (scelta\_tasto diversa da ESC )

ALLORA

dado = Lanciare\_dado()

Scrivere "valore del dado: dado"

cella\_prec\_g[num]=cella\_g[num]

cella\_g[num]=cella\_g[num]+dado

SE (cella\_g[num] > DIM\_CAMPO)

ALLORA

cella\_g[num]=DIM\_CAMPO

FINE SE

Grafica\_cancellare\_pedina()

Grafica\_muovere\_pedina()

Vedere\_vittoria()

SE(vincitore=0)

ALLORA

Vedere\_Scala()

Vedere\_Serpente()

```

        FINE SE
        Scrivere "Turno del giocatore,num+1"
        Scrivere "fai la scelta.."
        Scrivere "lancio del dado.."
        Scrivere "valore del dado: dado"
        num=num+1
        SE(num=n_partecipanti)
        ALLORA
            num=0
        FINE SE
    FINE SE
    FINCHE( scelta_tasto diverso da ESC AND vincitore=0)
FINE

```

## Salvare\_risultati()

### Dati di input

Nome	Descrizione	Vincoli
vincitore	indica il vincitore	intero da [1-4]
Record_file	file sul quale ci sono i record dei giocatori	Binario [pieno,vuoto]
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]

### Dati di Lavoro

Nome	Descrizione	Vincoli
DEFINIRE STRUTTURA Record	Struttura che serve a contenere tutti i dati per stabilire i record	
intero n_giocatore		
intero p_vinte		
intero p_giocate		
fine definizione R		
i	indice	

### Dati di output

Nome	Descrizione	Vincoli
Record_file	file sul quale ci sono i record dei giocatori	Binario [pieno,vuoto]

### INIZIO

```

    i=0
    R record
    Aprire il file "Record_file"
    leggere il record nella prima posizione e poni i suoi valori rispettivamente nella struttura record
    MENTRE(i<n_partecipanti AND il file non è terminato)
        SE(vincitore=record.n_giocatore)
            ALLORA
                record.p_vinte=record.p_vinte+1
            FINE SE
        record.p_giocate=record.p_giocate+1
        leggere il record nella posizione seguente e poni i suoi valori rispettivamente nella struttura
    record
        i=i+1
    FINE MENTRE
    Chiudere il file "Record_file"
FINE

```

## Turno\_giocatore()

### Dati di input

Nome	Descrizione	Vincoli
s	variabile locale che indica il carattere letto	intero [8,13]
num	indica il turno del giocatore	intero da [0-3]

### Dati di output

Nome	Descrizione	Vincoli
s	variabile locale che indica il carattere letto	intero

### Dati di Lavoro

Nome	Descrizione	Vincoli
ESC	costante globale che contiene il numero ascii per riconoscere il valore associato al tasto ESC	intero 27
INVIO	costante globale che contiene il numero ascii per riconoscere il valore associato al tasto INVIO	intero 13
INIZIO		
s=0		
ESEGUI		
SCRIVERE "Turno del Giocatore num+1"		
SCRIVERE "Fai la scelta : "		
LEGGERE s		
SE(s=INVIO)		
ALLORA		
Scrivere "lancio dado..."		
FINE SE		
FINCHE' (s diverso da ESC AND s diverso da INVIO)		
FINE		

## Lanciare\_dado()

### Dati di input

Nome	Descrizione	Vincoli
MAXDADO	valore massimo da generare	intero

### Dati di output

Nome	Descrizione	Vincoli
d	variabile locale contiene il valore generato casualmente	intero[1,MAXDADO]

### Dati di Lavoro

Nome	Descrizione	Vincoli
MAXDADO	valore massimo da generare	intero

### INIZIO

d=0

d= valore tra 1 e MAXDADO generato casualmente (inclusi gli estremi [1,MAXDADO]



FINE

## Vedere\_scala()

### Dati di input

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi	4
num	Variabile locale che è una stringa	stringa di caratteri con max 4 caratteri
ELEM_V	Quantità elementi del vettore scala	intero 5
v_scala_partenza(20, 17, 10, 34, 59)	Sequenza di interi di 5 elementi	
v_scala_arrivo(64, 38, 76, 95, 83)	Sequenza di interi di 5 elementi	

### Dati di output

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
cella_g_prec[GIOCATORI_MAX]	vettore che contiene tutte le posiz.prec di ogni giocatore sposta le pedine graficamente	intero da [0-100]

### Dati di Lavoro

Nome	Descrizione	Vincoli
rilevata ] i salvaind INIZIO i=0 rilevata=FALSO MENTRE (i<ELEM_V) SE ( cella_g[num] =v_scala_partenza[i] ) ALLORA rilevata=VERO salvaind=i FINE SE i=i+1 FINE MENTRE SE (rilevata=VERO) ALLORA SCRIVERE "Giocatore num+1 sulla scala!" cella_g[num]=v_scala_arrivo[salvaind] cella_g_prec[num]=v_scala_partenza[salvaind] Grafica_cancellare_pedina() Grafica_muovere_pedina() FINE SE FINE	variabile temporanea booleana  indice scala variabile che contiene l'indice corretto	BOOLEANA[FALSO,VERO]

## Vedere\_serpente()

### Dati di input

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
cella_g_prec[GIOCATORI_MAX]	vettore che contiene tutte le posiz.prec di ogni giocatore	intero da [0-100]
num	indica il turno del giocatore	intero da [0-3]
ELEM_V	quantità elementi nei vettori dei serpenti	
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi	4
V_serpente_partenza=(96, 88, 77, 46, 67)	Sequenza di interi di 5 elementi	
V_serpente_arrivo=(52, 23, 7, 36, 29)	Sequenza di interi di 5 elementi	

### Dati di output

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
cella_g_prec[GIOCATORI_MAX]	vettore che contiene tutte le posiz.prec di ogni giocatore sposta le pedine graficamente	intero da [0-100]

### Dati di Lavoro

Nome	Descrizione	Vincoli
rilevata ]	variabile temporanea	booleana
i	indice scala	BOOLEANA[FALSO,VERO]
salvaind	variabile che contiene l'indice corretto	

INIZIO

```
i=0
rilevata=FALSO
MENTRE (i<ELEM_V)
    SE ( cella_g[num]=V_serpente_partenza[i] )
        ALLORA
            rilevata=VERO
            salvaind=i
        FINE SE
    i=i+1
FINE MENTRE
SE (rilevato=VERO)
    ALLORA
        SCRIVERE "Giocatore num+1 sul serpente!"
        cella_g[num]=V_serpente_arrivo[salvaind]
        cella_prec_g[num]=v_serpente_partenza[salvaind]
        Grafica_cancellare_pedina()
        Grafica_muovere_pedina()
    FINE SE
```

FINE

## Vedere\_vittoria()

### Dati di input

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
num	indica il turno del giocatore	intero da [0-3]
DIM_CAMPO	dimensione del campo da gioco	intero 100
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi	4

### Dati di output

Nome	Descrizione	Vincoli
vincitore	indica il vincitore	intero da [1-4]

INIZIO

```
SE(cella_g[num]>=DIM_CAMPO)
  ALLORA
    SCRIVERE "Giocatore num+1 ha vinto!"
    vincitore=num+1
```

FINE SE

FINE

## Grafica\_cancellare\_pedina()

### Dati di input

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
cella_g_prec[GIOCATORI_MAX]	vettore che contiene tutte le posiz.prec di ogni giocatore	intero da [0-100]
num	indica il turno del giocatore	intero da [0-3]
DIM_CAMPO	dimensione del campo da gioco	intero 100

### Dati di output

Nome	Descrizione	Vincoli
	cancella dal campo la posizione precedente della pedina associata al giocatore	

### Dati di Lavoro

Nome	Descrizione	Vincoli
casella	casella che indica dove si trova il gicocatore	
unita	resto della divisione parte intera	
decina	resto della divisione parte decimale	
resto	variabile temporanea che contiene i resti all'occorrenza	
flag	variabile booleana che può valere	FALSO O VERO
giocatore	indica il turno del giocatore	
i	indice	
x	coordinate per la stampa	
y	coordinate per la stampa a video	

INIZIO

x=1

```

y=1
casella,decina,unita,resto=0, ,giocatore
BOOLEAN FLAG
casella=cella_prec_g[num]
giocatore=num
decina=casella/10
unita=casella%10
i=2
/*SE è UN MULTIPLIO DI DIECI*
SE (unita=0)
  ALLORA
    /*SE è pari*/
    SE(decina%2=0)
      ALLORA
        x=1
        i=0
        y=38
        MENTRE (i<=10 AND flag=FALSO)
          y=y-6
          SE(DECINA=i)
            ALLORA
              flag=VERO
            FINE SE
          i=i+2
        FINE MENTRE
      ALTRIMENTI
        /*SE DISPARI*/
        i=1
        x=46
        y=35
        MENTRE(i<10 AND flag=FALSO)
          y=y-6
          SE (decina=i)
            ALLORA
              flag=VERO
            FINE SE
          i=i+2
        FINE MENTRE
      FINE SE
    /*SE NON è MULTIPLIO DI DIECI*/
    ALTRIMENTI
      resto=decina%2
      SE (resto=0)
        ALLORA
          /*SE PARI*/
          i=0
          y=29
          /*trova la y*/
          flag=FALSO
          MENTRE (i<=8 AND flag=FALSO)
            SE (decina=i)
              ALLORA
                flag=VERO

```

```

        ALTRIMENTI
            y=y-6
        FINE SE
        i=i+2
    FINE MENTRE
/*TROVA LA X*/
x=1
i=1
flag=FALSO
MENTRE(i<=9 AND flag=VERO)
    SE(i=unita)
        ALLORA
            flag=VERO
        ALTRIMENTI
            x=x+5
        FINE SE
        i=i+1
    FINE MENTRE
ALTRIMENTI
/*SE DISPARI*/
    i=1
    y=26
    /*trova la y*/
    flag=FALSO
    MENTRE (i<=9 AND flag=FALSO)
        SE (decina=i)
            ALLORA
                flag=1
            ALTRIMENTI
                y=y-6
            FINE SE
            i=i+2
        FINE MENTRE
    /*TROVA LA X*/
    x=46
    i=1
    flag=0
    MENTRE (i<=9 AND flag=FALSO)
        SE(i=unita)
            ALLORA
                flag=VERO
            ALTRIMENTI
                x=x-5
            FINE SE
            i=i+1
        FINE MENTRE
    FINE SE
FINE SE

/*SPOSTO LE PEDINE EFFETTIVAMENTE IN BASE AL NUMERO DEL GIOCATORE*/
x=x+giocatore
SE(giocatore=0)
    ALLORA

```

```

        cancello la pedina 1 in coordinate x,y
    ALTRIMENTI
        SE(giocatore=1)
            ALLORA
                cancello la pedina 2 in coordinate x,y
            ALTRIMENTI
                SE(giocatore=2)
                    ALLORA
                        cancello la pedina 3 in coordinate x,y
                    ALTRIMENTI
                        SE (giocatore=3)
                            ALLORA
                                cancello la pedina 3 in coordinate x,y
                            FINE SE
                        FINE SE
                    FINE SE
                FINE SE
            FINE SE
        FINE SE
    FINE

```

## Grafica\_muovere\_pedina()

### Dati di input

Nome	Descrizione	Vincoli
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
num	indica il turno del giocatore	intero da [0-3]
DIM_CAMPO	dimensione del campo da gioco	intero 100

### Dati di output

Nome	Descrizione	Vincoli
	Stampa a video la nuova posizione nel campo della pedina associata al giocatore	

### Dati di Lavoro

Nome	Descrizione	Vincoli
casella	casella che indica dove si trova il giocatore	
unita	resto della divisione parte intera	
decina	resto della divisione parte decimale	
resto	variabile temporanea che contiene i resti all'occorrenza	
flag	variabile booleana che può valere FALSO O VERO	
giocatore	indica il turno del giocatore	
i	indice	
x	coordinate per la stampa	
y	coordinate per la stampa a video	
INIZIO		

```

    x=1
    y=1
    casella,decina,unita,resto=0 ,giocatore
    boolean flag
    casella=cella_g[num]
    giocatore=num
    decina=casella/10
    unita=casella%10
    i=2
    /*SE è UN MULTIPLO DI DIECI*
    SE (unita=0)

```

```

ALLORA
    /*SE è pari*/
    SE(decina%2=0)
    ALLORA
        x=1
        i=0
        y=38
        MENTRE (i<=10 AND flag=FALSO)
            y=y-6
            SE(DECINA=i)
            ALLORA
                flag=VERO
            FINE SE
            i=i+2
        FINE MENTRE
    ALTRIMENTI
        /*SE DISPARI*/
        i=1
        x=46
        y=35
        MENTRE(i<10 AND flag=FALSO)
            y=y-6
            SE (decina=i)
            ALLORA
                flag=VERO
            FINE SE
            i=i+2
        FINE MENTRE
    FINE SE
    /*SE NON è MULTIPLO DI DIECI*/
ALTRIMENTI
    resto=decina%2
    SE (resto=0)
    ALLORA
        /*SE PARI*/
        i=0
        y=29
        /*trova la y*/
        flag=FALSO
        MENTRE (i<=8 AND flag=FALSO)
            SE (decina=i)
            ALLORA
                flag=VERO
            ALTRIMENTI
                y=y-6
            FINE SE
            i=i+2
        FINE MENTRE
        /*TROVA LA X*/
        x=1
        i=1
        flag=FALSO
        MENTRE(i<=9 AND flag=FALSO)

```

```

        SE(i=unita)
            ALLORA
                flag=VERO
            ALTRIMENTI
                x=x+5
            FINE SE
            i=i+1
        FINE MENTRE
    ALTRIMENTI
        /*SE DISPARI*/
        i=1
        y=26
        /*trova la y*/
        flag=FALSO
        MENTRE (i<=9 AND flag=FALSO)
            SE (decina=i)
                ALLORA
                    flag=VERO
                ALTRIMENTI
                    y=y-6
            FINE SE
            i=i+2
        FINE MENTRE
        /*TROVA LA X*/
        x=46
        i=1
        flag=FALSO
        MENTRE (i<=9 AND flag=FALSO)
            SE(i=unita)
                ALLORA
                    flag=VERO
            ALTRIMENTI
                x=x-5
            FINE SE
            i=i+1
        FINE MENTRE
    FINE SE
FINE SE

/*SPOSTO LE PEDINE EFFETTIVAMENTE IN BASE AL NUMERO DEL GIOCATORE*/
x=x+giocatore
SE(giocatore=0)
    ALLORA
        sposto la pedina 1 in coordinate x,y
    ALTRIMENTI
        SE(giocatore=1)
            ALLORA
                sposto la pedina 2 in coordinate x,y
            ALTRIMENTI
                SE(giocatore=2)
                    ALLORA
                        sposto la pedina 3 in coordinate x,y
                    ALTRIMENTI

```



```

        SE (gicatore=3)
            ALLORA
                spostato la pedina 3 in coordinate x,y
            FINE SE
        FINE SE
    FINE SE
FINE SE
FINE

```

## Carica\_partita()

### Dati di input

Nome	Descrizione	Vincoli
nome_scelto	nome inserito var locale	carattere [5]
size	dimensione del file "Partite_salvate"	long
scelta_carica	scelta dell'utente se tornare indietro o no	intero[ s,ESC]

### Dati di output

Nome	Descrizione	Vincoli
Partite_salvate	(aggiornato senza file sul quale ci sono le partite salvate la partita appena caricata)	Binario [pieno,vuoto]
cella_g[GIOCATORI_MAX]	indica la cella del giocatore Ripresa esecuzione partita	intero da [0-100]

### Dati di Lavoro

Nome	Descrizione	Vincoli
ESC	costante globale che contiene il numero ascii per riconoscere il valore associato al tasto ESC	intero 27
trovato	variabile locale che indica se un la partita è stata trovata	booleano [FALSO,VERO]

```

INIZIO
scelta_carica=0
caricato=FALSO
trovato=FALSO
size=Controllo_esistenza_partita()
SE(size diverso da 0)
    ALLORA
        scelta_carica=Input_carica()
        SE (scelta_carica diverso da ESC)
            ALLORA
                ESEGUI
                    Leggere_partite_salvate()
                    Input_nome_partita()
                    trovato=Cerca_partita()
                    SE (trovato=VERO)
                        ALLORA
                            caricato=VERO
                            Scrivere "PARTITA CARICATA CON SUCCESSO!"
                        ALTRIMENTI
                            caricato=FALSO
                            Scrivere "NOME PARTITA NON ESISTENTE"
                    FINE SE
            FINE SE
        FINE SE
    FINE SE

```

```

    FINCHE(caricato=FALSO)
    SE(caricato=VERO)
        ALLORA
            Cancellare_partita_caricata()
            Grafica_Nuova_partita()
            Settare_pedine()
            Spostare_pedina()
            SE(vincitore diverso da 0)
                allora
                    Salvare_risultati()
            FINE SE
        FINE SE
    FINE SE
ALTRIMENTI
    Scrivere "ATTENZIONE!NON CI SONO PARTITE DA CARICARE!!!"
FINE SE
FINE

```

## Cancellare\_partita\_caricata()

### Dati di input

Nome	Descrizione	Vincoli
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]

### Dati di output

Nome	Descrizione	Vincoli
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]

### Dati di Lavoro

Nome	Descrizione	Vincoli
DEFINIRE STRUTTURA Partite_salvate	Struttura che serve a contenere	
carattere nome_part	tutti i dati da salvare su un file	
intero posg1	per far riprendere una partita	
intero posg2		
intero posg3		
intero posg4		
intero n		
intero turno		
fine definizione Ps		
vecchio_nome	sequenza di caratteri con il nome del 1 file	
nuovo_nome	sequenza di caratteri con il nome del 2 file	

### INIZIO

```

Ps Par_sal
vecchio_nome="Partite_salvate"
nuovo_nome="Partite_salvate2"
Aprire il file "Partite_salvate"
Aprire il file "Partite_salvate2"
leggere primo record e porlo nella struttura Par_sal
Mentre (il file non è terminato)
    SE(Par_sal.nome_part=nome)
        ALLORA
            Scrivere il record nel file(Partite_salvate2)
    FINE SE

```

leggere il seguente record nel file "partite salvate"  
FINE MENTRE  
chiudere il file "Partite\_salvate"  
chiudere il file "Partite\_salvate2"  
rimuovere il file vecchio\_nome  
rinominare il file vecchio\_nome con nuovo\_nome  
FINE

## Controllo\_Esistenza\_partite()

### Dati di input

Nome	Descrizione	Vincoli
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]

### Dati di output

Nome	Descrizione	Vincoli
size	dimensione del file "Partite_salvate"	long

### Dati di Lavoro

Nome	Descrizione	Vincoli
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]

INIZIO  
size=0  
aprire il file "Partite\_salvate"  
size=dimensione del file  
chiudere il file "Partite salvate"  
FINE

## Grafica\_Carica\_partita()

### Dati di output

Nome	Descrizione	Vincoli
	stampa a video la prima parte dell'interfaccia del carica partita	

INIZIO  
Scrivere "CARICA PARTITA"  
Scrivere "-----"  
FINE

## Grafica\_Scelta\_carica()

### Dati di output

Nome	Descrizione	Vincoli
	stampa a video un'altra parte dell'interfaccia del carica partita	

INIZIO  
Scrivere "Vuoi caricare una partita salvata?");  
Scrivere "s=SI ESC = indietro "  
Scrivere "Fai la tua scelta: "  
FINE

## Input\_Carica()

### Dati di input

#### Nome

s

#### Descrizione

variabile locale che indica il carattere inserito

#### Vincoli

intero

### Dati di output

#### Nome

s

#### Descrizione

variabile locale che indica il carattere inserito

#### Vincoli

intero diverso da a and z  
diverso da A and Z e diverso da INVIO

mostra le interfacce grafiche del carica

### Dati di Lavoro

#### Nome

s

#### Descrizione

variabile locale che indica il carattere inserito

#### Vincoli

intero diverso da a and z  
diverso da A and Z e diverso da INVIO

ESC

costante globale che contiene il numero  
ascii  
per riconoscere il valore  
associato al tasto ESC

intero 27

INIZIO

s=0

ESEGUI

Grafica\_Carica\_partita()

Grafica\_Scelta\_Carica()

leggere s

FINCHE (s diverso da 's' AND s diverso da ESC)

FINE

## Settare\_Pedina()

### Dati di input

#### Nome

num

#### Descrizione

indica il turno del giocatore

#### Vincoli

intero da [0-3]

n\_partecipanti

quantità di partecipanti alla partita

intero [2-4]

### Dati di output

#### Nome

#### Descrizione

scrivere a video le pedine nella posizione caricata

#### Vincoli

### Dati di Lavoro

#### Nome

temp

#### Descrizione

salva num

#### Vincoli

dati di lavoro

INIZIO

temp=num

num=0

Mentre(num<n\_partecipanti)

    Grafica\_muovere\_pedina()

    incremento di 1 num

FINE MENTRE

num=temp

FINE

## Salva\_Partita()

### Dati di input

Nome	Descrizione	Vincoli
num	indica il turno del giocatore	intero da [0-3]
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]
nome_scelto	nome inserito var locale	carattere [5]
scelta_carica	scelta dell'utente se tornare indietro o no	intero[ INVIO,ESC]
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi	4
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]

### Dati di output

Nome	Descrizione	Vincoli
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]

### Dati di Lavoro

Nome	Descrizione	Vincoli
trovato	se la partita è già esistente	booleano[FALSO,VERO]
salvato	esito del salvataggio	booleano[FALSO,VERO]
]		
ESC	costante globale che contiene il numero ascii per riconoscere il valore associato al tasto ESC	INTERO 27

INIZIO

scelta\_salva=0;

trovato=FALSO;

salvato=FALSO;

scelta\_salva=Input\_salva()

SE (scelta\_salva diverso da ESC)

    ALLORA

        ESEGUI

            Leggere\_Partite\_salvate()

            Input\_nome\_partita()

            SE (trovato=FALSO)

                ALLORA

                    Scrivere\_nel\_file()

                        salvato=VERO

                        Scrivere "PARTITA SALVATA CON SUCCESSO!"

            ALTRIMENTI

                salvato=FALSO

                Scrivere "PARTITA GIA ESISTENTE"

            FINE SE

        FINCHE (salvato=FALSO)

FINE SE  
FINE

## Cerca\_partita()

### Dati di input

Nome	Descrizione	Vincoli
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]

### Dati di output

Nome	Descrizione	Vincoli
trov	variabile locale booleana	booleano [FALSO,VERO]
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
SOLO SE è UN CARICA:		
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]
num	indica il turno del giocatore	intero da [0-3]

### Dati di Lavoro

Nome	Descrizione	Vincoli
trov	variabile locale booleana	booleano [FALSO,VERO]
DEFINIRE STRUTTURA Partite_salvate	Struttura che serve a contenere	
carattere nome_part	tutti i dati da salvare su un file	
intero posg1	per far riprendere una partita	
intero posg2		
intero posg3		
intero posg4		
intero n		
intero turno		
fine definizione Ps		

Inizio

Ps Par\_sal

Aprire il file "Partite\_salvate"

leggere primo record del file e porlo nella struttura Par\_sal

Mentre(il file non è terminato e trov=FALSO)

SE (nome=Par\_sal.nome\_part)

ALLORA

trov=VERO

SE (scelta\_menu=2)

ALLORA

num=Par\_sal.turno

n\_partecipnati=Par\_sal.n

```

SE(n_partecipanti=2)
  ALLORA
    cella_g[0]=Par_sal.posg1
    cella_g[1]=Par_sal.posg2
  ALTRIMENTI
    SE(n_partecipanti=3)
      ALLORA
        cella_g[0]=Par_sal.posg1
        cella_g[1]=Par_sal.posg2
        cella_g[2]=Par_sal.posg3
      ALTRIMENTI
        SE(n_partecipanti=4)
          ALLORA
            cella_g[0]=Par_sal.posg1
            cella_g[1]=Par_sal.posg2
            cella_g[2]=Par_sal.posg3
            cella_g[3]=Par_sal.posg4
          FINE SE
        FINE SE
      FINE SE
    FINE SE
  FINE SE
  leggere il seguente record del file e porlo nella struttura Par_sal
FINE MENTRE
chiudere il file "Partite_salvate"
FINE

```

## Grafica\_elenco\_partite()

### Dati di input

Nome	Descrizione	Vincoli
nome	nome della partita	sequenza di caratteri massimo 5
n	quantità di partecipanti	intero positivo

### Dati di output

Nome	Descrizione	Vincoli
	stampa a video elenco partite	

INIZIO

Scrivere " nome della partita salvata    numero dei partecipanti"

FINE

## Grafica\_Fine\_elenco()

### Dati di input

Nome	Descrizione	Vincoli
scelta_menu	scelta dell'utente	intero [0-5]

### Dati di output

Nome	Descrizione	Vincoli
	stampa a video il resto dell'interfaccia di salva partita	

Inizio

Scrivere "inserire il nome della"

SE(scelta\_menu=2)

Scrivere "partita da caricare"

ALTRIMENTI

Scrivere "partita da salvare"

FINE SE

Scrivere "e premere invio"

Scrivere "(MAX 5 LETTERE!)"

FINE

## Grafica\_Inizio\_Elenco()

Dati di output

Nome	Descrizione	Vincoli
	stampa a video le intestazioni della tabella che conterrà l'elenco delle partite	

INIZIO

Scrivere "NOME PARTITA #PARTECIPANTI"

FINE

## Grafica\_Salva\_parita()

Dati di output

Nome	Descrizione	Vincoli
	scrive a video la prima parte dell'interfaccia del salva partita	

INIZIO

Scrivere "SALVA PARTITA"

Scrivere "-----"

FINE

## Grafica\_Scelta\_salva()

Dati di output

Nome	Descrizione	Vincoli
	stampa un'altra parte dell'interfaccia di salva partita	

INIZIO

Scrivere "vuoi salvare la partita in corso?"

Scrivere "s=SI ESC = indietro"

Scrivere "Fai la tua scelta: "

FINE

## Input\_nome\_partita()

Dati di input

Nome	Descrizione	Vincoli
nome	nome inserito dall'utente della partita da salvare	sequenza di caratteri 5 massimo

Dati di output

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



nome	nome inserito dall'utente della partita da salvare	sequenza di caratteri 5 massimo
------	--	---------------------------------

#### Dati di Lavoro

Nome	Descrizione	Vincoli
i	indica la lunghezza della parola	intero
car	ultimo carattere inserito	carattere
ESC	costante globale	intero 27
INVIO	costante globale	intero 13
A	costate globale	intero 65
Z	costante globale	intero 90
a	costante globale	intero 97
z	costante globale	intero 122

INIZIO

i=0

car=0

Esegui

Scrivere "e premere invio: "

leggere nome

leggere l'ultimo carattere inserito mettilo in car

SE(car diverso da INVIO AND car diverso da ESC AND ((car varia da A a Z inclusi) OPPURE (car è compreso tra a e z)))

ALLORA

nome[i]=car

incrementa i di 1

ALTRIMENTI

SE(car=ESC AND i>0)

decrementa i di 1

nome[i]=simbolo di fine stringa

FINE SE

FINE SE

FINCHE ((i<5 AND i=0) OPPURE (i<5 AND car diverso da INVIO))

FINE

## Input\_salva()

#### Dati di input

Nome	Descrizione	Vincoli
s	variabile locale che indica il carattere inserito	intero diverso da a and z diverso da A and Z e diverso da INVIO

#### Dati di output

Nome	Descrizione	Vincoli
s	variabile locale che indica il carattere inserito	intero diverso da a and z diverso da A and Z e diverso da INVIO
	stampa la grafica della scelta	

#### Dati di Lavoro

Nome	Descrizione	Vincoli
ESC	costante globale che contiene il numero ascii per riconoscere il valore associato al tasto ESC	intero 27

```

INIZIO
s=0
ESEGUI
    Grafica_salva_partita()
    Grafica_Scelta_salva()
    leggere s
FINCHE(s è diverso da 's' AND ESC)

FINE

```

## Leggere\_partite\_salvate()

### Dati di input

Nome	Descrizione	Vincoli
scelta_menu	scelta dell'utente	intero [0-5]
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]

### Dati di output

Nome	Descrizione	Vincoli
	stampa a video i nomi di tutte le partite gia salvate sul file	

### Dati di Lavoro

Nome	Descrizione	Vincoli
DEFINIRE STRUTTURA Partite_salvate	Struttura che serve a contenere	
carattere nome_part	tutti i dati da salvare su un file	
intero posg1	per far riprendere una partita	
intero posg2		
intero posg3		
intero posg4		
intero n		
intero turno		
fine definizione Ps		

```

INIZIO
PS Par_sal
SE(scelta_menu = 2)
    ALLORA
        Grafica_Carica_Partita()
    ALTRIMENTI
        Grafica_Salva_partita()
FINE SE
aprire il file "Partite_salvate"
Grafica_Inizio_elenco()

```

```

leggere il primo record del file e porlo nella struttura Par_sal
MENTRE (il file non è terminato)
    Grafica_elenco_partite()
    leggere il record seguente del file e porlo nella struttura Par_sal
FINE MENTRE
Grafica_Fine_elenco()
Chiudere il file "Partite_salvate"
FINE

```

## Scrivere\_nel\_file()

### Dati di input

Nome	Descrizione	Vincoli
nome	nome inserito dall'utente della partita da salvare	sequenza di caratteri 5 massimo

### Dati di output

Nome	Descrizione	Vincoli
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]

### Dati di Lavoro

Nome	Descrizione	Vincoli
trov	variabile locale booleana	booleano [FALSO,VERO]
DEFINIRE STRUTTURA Partite_salvate	Struttura che serve a contenere	
carattere nome_part	tutti i dati da salvare su un file	
intero posg1	per far riprendere una partita	
intero posg2		
intero posg3		
intero posg4		
intero n		
intero turno		
fine definizione Ps		
cella_g[GIOCATORI_MAX]	indica la cella del giocatore	intero da [0-100]
num	indica il turno del giocatore	intero da [0-3]
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]
Partite_salvate	file sul quale ci sono le partite salvate	Binario [pieno,vuoto]
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi	4

### INIZIO

```

    trov=FALSO
    Ps Par_sal

```

```

aprire il file "Partite_salvate"
Par_sal.n=n_partecipanti
Par_sal.turno=num
Par_sal.posg1=cella_g[0]
Par_sal.posg2=cella_g[1]
Par_sal.posg3=cella_g[2]
Par_sal.posg4=cella_g[3]
Par_sal.nome_part=nome
Scrivere nel file il record in fondo al file
chiudere il file"Partite_salvate"

```

FINE

## Impostazioni()

### Dati di input

Nome	Descrizione	Vincoli
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]
DUE	costante numerica	intero 50
TRE	costante numerica	intero 51
QUATTRO	costante numerica	intero 52

### Dati di output

Nome	Descrizione	Vincoli
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]

INIZIO

```

n_partecipanti=Input_giocatori()
Se (n_partecipanti==TRE)
    Allora
        n_partecipanti=3
Altrimenti
    Se (n_partecipanti==DUE)
        allora
            n_partecipanti=2
    Altrimenti
        n_partecipanti=4
    Fine se
Fine se

```

FINE

## Grafica\_Impostazioni()

### Dati di output

Nome	Descrizione	Vincoli
	Stampa a video la l'interfaccia per la schermata impostazioni	

INIZIO

```

Scrivere "IMPOSTAZIONI"

```

```

Scrivere "-----"
Scrivere "numero giocatori [2-4]"
Scrivere "inserire il numero dei giocatori"
Scrivere "oppure premi ESC per tornare al menu"
FINE

```

## Input\_giocatori()

### Dati di input

Nome	Descrizione	Vincoli
n_partecipanti	quantità di partecipanti alla partita	intero [2-4]

### Dati di output

Nome	Descrizione	Vincoli
num	variabile locale che contiene il valore letto	intero [2-4]

### Dati di Lavoro

Nome	Descrizione	Vincoli
DUE	costante numerica	intero 50
TRE	costante numerica	intero 51
QUATTRO	costante numerica	intero 52
ESC	costante globale che contiene il numero ascii per riconoscere il valore associato al tasto ESC	intero 27

INIZIO

num=0

Grafica\_impostazioni()

LEGGERE num

MENTRE ((num<DUE OR num>QUATTRO) AND num!=ESC)

Scrivere "VALORE NON VALIDO!INSERIRE UN VALORE TRA 2-4"

Grafica\_impostazioni()

LEGGERE num

FINE MENTRE

FINE

## Record()

### Dati di input

Nome	Descrizione	Vincoli
Record_file	file sul quale ci sono i record	Binario [pieno,vuoto]

s	carattere inserito dall'utente	Booleano [FALSO,VERO]
---	--------------------------------	-----------------------

### Dati di output

Stampa a video una tabella che conterrà i record con le intestazioni dei campi

Ordina i record in modo decrescente delle partite vinte

Stampa a video i record contenuti nel file "Record\_file"

## Dati di Lavoro

Nome	Descrizione	Vincoli
ESC	costante globale che contiene il numero ascii per riconoscere il valore associato al tasto ESC	intero 27

INIZIO

 $s=0$ 

## Grafica\_Record()

Leggere\_risultati()

Scrivere "Premi ESC per tornare al menù : "

ESEGUI

leggere s

FINCHE (s diverso da ESC)

FINE

## Grafica\_record()

### Dati di input

Nome	Descrizione	Vincoli
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi 4	

### Dati di output

stampa a video la tabella con l'intestazione dei campi

### Dati di Lavoro

Nome	Descrizione	Vincoli
i	indice	intero

INIZIO

Scrivere "RECORD"

Scrivere "-----"

Scrivere "GIOCATORE PARTITE VINTE PARTITE FATTE"

Scrivere il bordo della tabella

```
MENTRE(i<GIOCATORI MAX)
```

Scrivere bordo tabella

$$i=i+1$$

FINE MENTRE

FINE

## Leggere\_risultati()

### Dati di input

Nome	Descrizione	Vincoli
Record_file	file sul quale ci sono i record Binario	[pieno,vuoto]

### Dati di output

Nome	Descrizione	Vincoli
DEFINIRE STRUTTURA Record	Struttura che serve a contenere	
intero n_giocatore	tutti i dati per stabilire i record	
intero p_vinte		
intero p_giocate		
fine definizione R		

giocatori[GICATORI_MAX]	vettore di interi ordinato in base a partite
partite_vinte[GIOCATORI_MAX]	vettore di interi ordinato decrescente
partite_fatte[GIOCATORI_MAX]	vettore di interi ordinato in base a partite vinte

### Dati di Lavoro

Nome	Descrizione	Vincoli
DEFINIRE STRUTTURA Record	Struttura che serve a contenere	
intero n_giocatore	tutti i dati per stabilire i record	
intero p_vinte		
intero p_giocate		
fine definizione R		
i	intero	

### INIZIO

```

i=0
R record
Aprire il file "Record_file"
leggere il primo record del file e poni rispettivamente i valori nella struttura record
ESEGUI
FINCHE' ( il file non è terminato)
    giocatori[i]=R.n_giocatore
    partite_vinte[i]=R.p_vinte
    partite_fatte[i]=R.p_giocate
    leggere il record seguente del file e poni rispettivamente i valori nella struttura record
    i=i+1
FINE ESEGUI
Chiudere il file "Record_file"
Ordinare_record()
MENTRE(i<GIOCATORI_MAX)
    SCRIVERE giocatori[i]
    SCRIVERE partite_vinte[i]
    SCRIVERE partite_fatte[i]
    i=i+1
FINE MENTRE
FINE

```

## Ordinamento\_record()

### Dati di input

Nome	Descrizione	Vincoli
giocatori[GICATORI_MAX]	vettore di interi ordinato in base a partite	
partite_vinte[GIOCATORI_MAX]	vettore di interi ordinato decrescente	
partite_fatte[GIOCATORI_MAX]	vettore di interi ordinato in base a partite vinte	
GIOCATORI_MAX	costante globale che indica la quantità di giocatori massimi	4

### Dati di output

Nome	Descrizione	Vincoli
g [GICATORI_MAX]	vettore di interi ordinato in base a partite indica num giocatore	
f[GIOCATORI_MAX]	vettore di interi ordinato decrescente indica le partite fatte	
v [GIOCATORI_MAX]	vettore di interi ordinato in base a partite vinte indica le partite vinte	

## Dati di Lavoro

Nome	Descrizione	Vincoli
temp	serve per l'ordinamento	
i	indice primo ciclo	
j	indice secondo ciclo	

INIZIO

// Ordinare record per partite\_vinte in modo decrescente

MENTRE (i<GIOCATORI\_MAX)

temp=v[i]

j=0

MENTRE

SE( temp>v[j] )

ALLORA

temp=g[j]

gi[j]=g [i]

gi [i]=temp

temp=f[j]

f[j]=f[i]

f[i]=temp

temp=v[j]

v[j]=v[i]

v[i]=temp

FINE SE

j=j+1

FINE MENTRE

i=i+1

FINE MENTRE

FINE