**CASI DI TEST FANORONA**

Nome funzione**: controllare\_coordinate\_spostamento(pedina \*p )**

p->Cordx = {A,a,B,b,C,c,D,d,E,e}

p->Cordy = {1,2,3,4,5,6,7,8,9}

Caso 1:

p->Cordx = A

p->Cordy = 0

Output: Messaggio di errore “**Coordinate errate!!**”

Caso 2:  
p->Cordx = F

p->Cordy = 2

Output: Messaggio di errore “**Coordinate errate**!!”

Caso 3:

p->Cordx = 5

p->Cordy= 7

Output: Messaggio di errore “**Coordinate errate**!!”

Caso 4:

p->Cordx = 5

p->Cordy= 0

Output: Messaggio di errore “**Coordinate errate**!!”

Caso 5

p->Cordx = e

p->Cordy= 7

Output:

Caso 6:

p->Cordx = A

p->Cordy = 1

Output:

Nome Funzione: **controllare\_colonna(pedina\*dap,pedina\*versop,check\*ck)**

**CLASSI DI EQUIVALENZA**

Ctrl ={0,1,2}

Caso 1

Dap->cordx=B

Dap->cordy=5

Versop->cordx=B

Versop->cordy=6

Output: ctrl=1, lo spostamento avviene correttamente

Caso 2

Dap->cordx=B

Dap->cordy=5

Versop->cordx=B

Versop->cordy=7

Output: ctrl=0, Errore: Ci stiamo spostando di due o più colonne

Caso 3

Dap->cordx=B

Dap->cordy=7

Versop->cordx=B

Versop->cordy=5

Output: ctrl=0, Errore: Ci stiamo spostando di due o più colonne

Caso 4

Dap->cordx=B

Dap->cordy=5

Versop->cordx=C

Versop->cordy=5

Output: ctrl=2, Ci stiamo spostando sulla stessa colonna

Nome Funzione: **controllare\_riga(pedina\*dap,pedina\*versop,check\*ck)**

**CLASSI DI EQUIVALENZA**

Ctrl={0,1,2}

Caso 1

Dap->cordx=D

Dap->cordy=5

Versop->cordx=C

Versop->cordy=5

Output: ctrl=1 , Ci stiamo spostando correttamente di una riga

Caso 2

Dap->cordx=D

Dap->cordy=5

Versop->cordx=B

Versop->cordy=5

Output: ctrl= 0 , Errore: Ci stiamo spostando di due o più righe

Caso 3

Dap->cordx=C

Dap->cordy=4

Versop->cordx=C

Versop->cordy=5

Output: ctrl=2, Ci stiamo spostando sulla stessa riga

Nome Funzione: **effettuare\_spostamento(partita\*g,pedina\*dap,pedina\*versop)**

**CLASSI DI EQUIVALENZA**

Exit={FALSO,VERO}

Caso 1

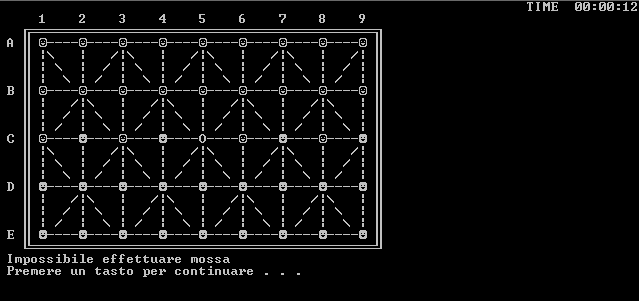
Dap->cordx=C

Dap->cordy=4

Versop->cordx=B

Versop->cordy=4

Output: Exit=VERO , Errore: Ci stiamo spostando in una casella non vuota



Caso 2

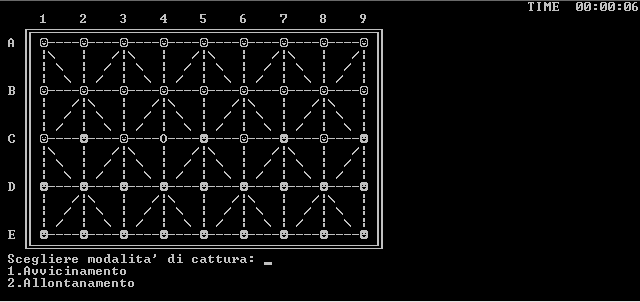
Dap->cordx=C

Dap->cordy=4

Versop->cordx=C

Versop->cordy=5

Output: Exit= FALSO , Ci stiamo spostando correttamente



Nome Funzione: **mangiare\_pedina\_avvicinamento(partita\*p, check \*cordpt, pedina\*dap, pedina \*versop, int pedg)**

**CLASSI DI EQUIVALENZA**

Caso 1

Dap->cordx=D

Dap->cordy=6

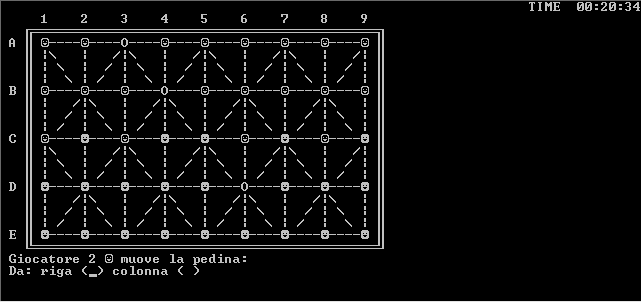
Versop->cordx=C

Versop->cordy=5

Cordpt->up=VERO

Cordpt->sx=VERO

Output: Nella posizione B,4 e A,3 della “matrice\_pedine” cambiano il valore in 0



Caso 2

Dap->cordx=B

Dap->cordy=5

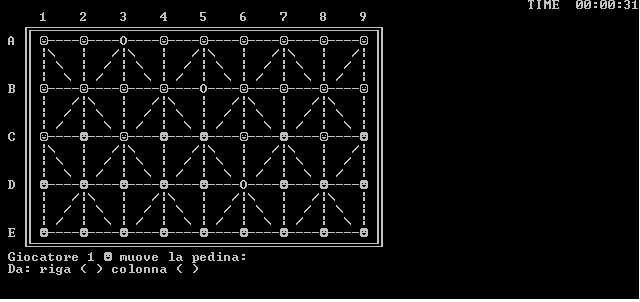
Versop->cordx=B

Versop->cordy=4

Cordpt->up=FALSO

Cordpt->down=FALSO

Cordpt->sx=VERO

Output: 

Caso 3

Dap->cordx=D

Dap->cordy=4

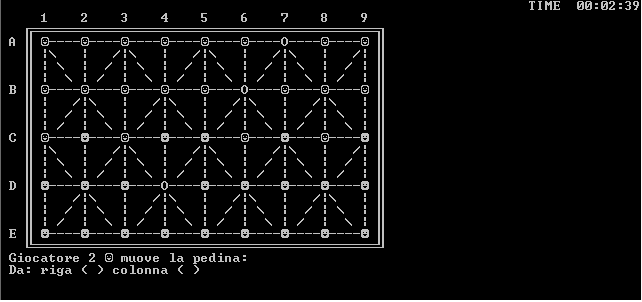
Versop->cordx=C

Versop->cordy=5

Cordpt->up=VERO

Cordpt->dx=VERO

Output: Nella posizione B,6 e A,7 della “matrice\_pedine” cambiano il valore in 0



Nome Funzione: **mangiare\_pedina\_allontanamento(partita\*p, check \*cordpt, pedina\*dap, pedina \*versop, int pedg)**

**CLASSI DI EQUIVALENZA**

Caso 1

Dap->cordx=c

Dap->cordy=5

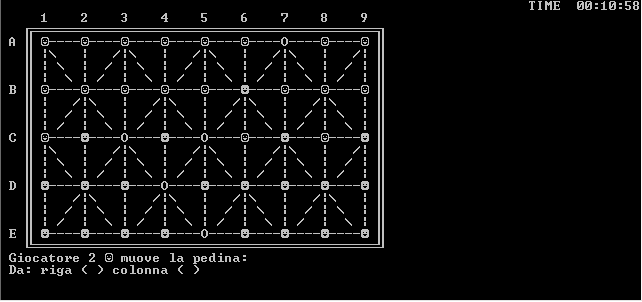
Versop->cordx=B

Versop->cordy=6

Cordpt->up=VERO

Cordpt->dx=VERO

Output: la posizione D,4 della “matrice\_pedine” cambia il valore in 0



Caso 2

Dap->cordx=C

Dap->cordy=4

Versop->cordx=D

Versop->cordy=4

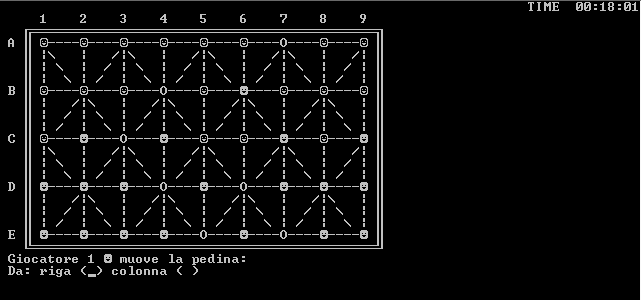
Cordpt->up=VERO

Cordpt->dx=FALSO

Cordpt->sx=FALSO

Output: la posizione D,4 della “matrice\_pedine” cambia il valore in 0

Prima :



Dopo: