



Corso di Informatica Medica

Esercitazione IV

Alessandro A. Nacci
nacci@elet.polimi.it - alessandronacci.com

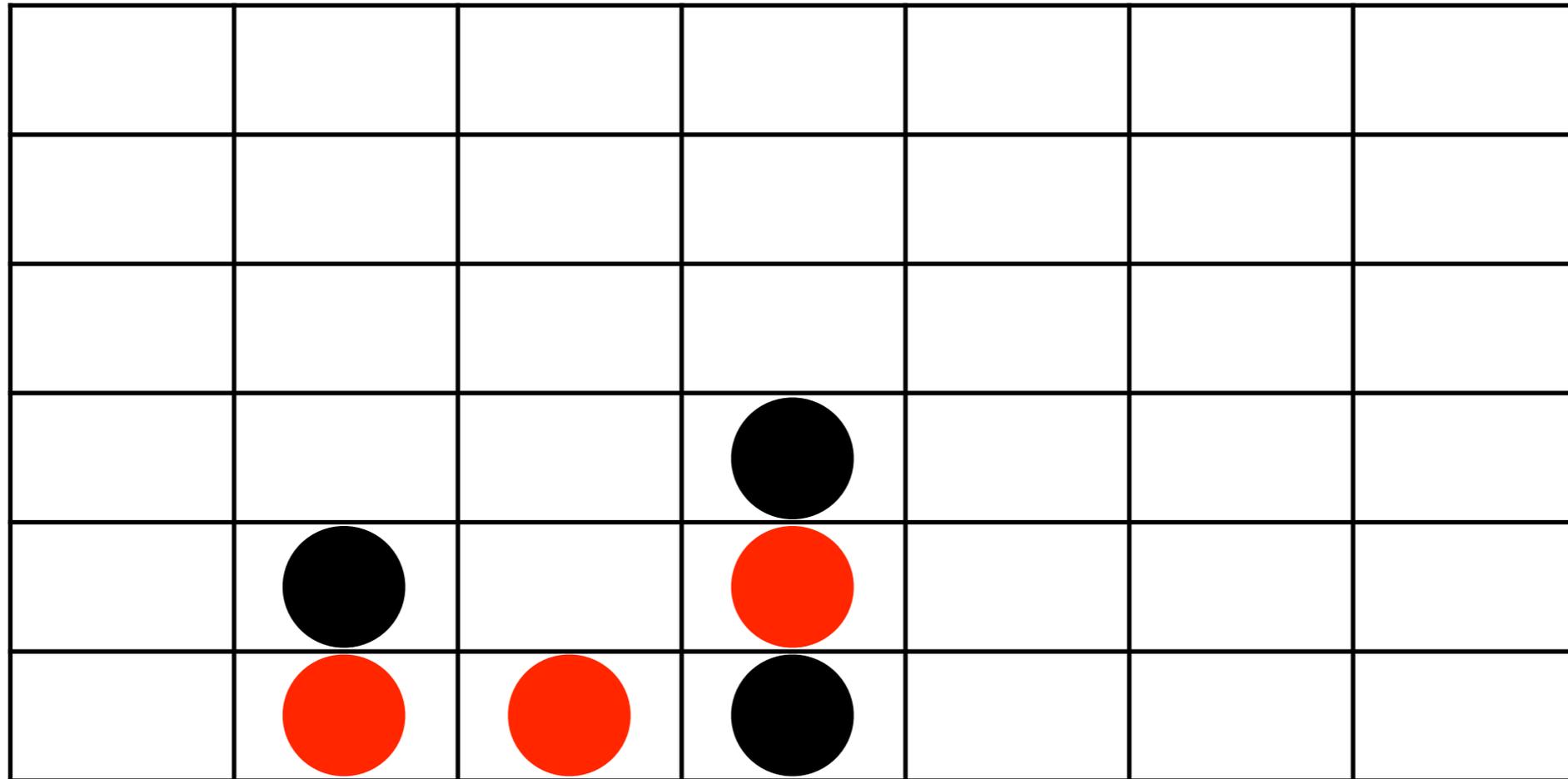


- Scriviamo un programma C che implementi il gioco del **Forza 4**
- Due giocatori, entrambi “reali”
- Il programma deve permettere di giocare
- Il programma deve annunciare il vincitore





Forza 4: rappresentazione



Come facciamo a rappresentare nel terminale, con quello che già conosciamo, delle pedine rosse e delle pedine nere?



Forza 4: rappresentazione

0

x

POSSIAMO **RIUTILIZZARE** PRATICAMENTE TUTTO
QUELLO CHE ABBIAMO
FATTO NELL'ESERCIZIO PRECEDENTE...

ECCO A COSA SERVONO LE FUNZIONI! :D

y

*Quindi abbiamo una matrice,
descritta da due indici e
che contiene caratteri...*



Ma è il PIANO
CARTESIANO!



- Anche se possiamo utilizzare buona parte del codice già scritto per il “PIANO CARTESIANO”, qualche concetto ancora ci manca:
 - Come rappresentiamo un GIOCATORE?
 - Come gestiamo l’inserimento di una pedina?
 - Come controlliamo la vincita?



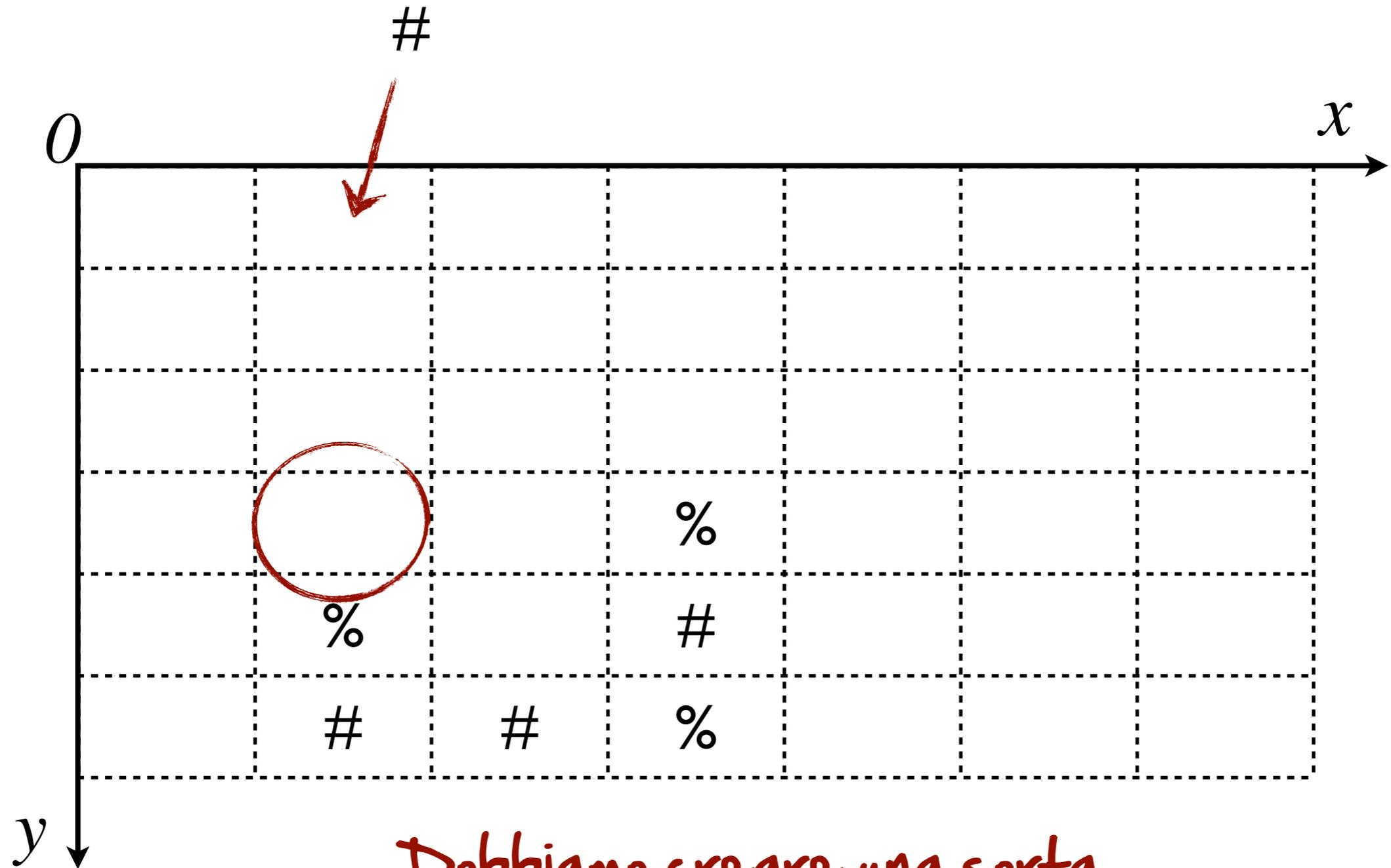
Il giocatore

- Un giocatore è identificato da:
 - IL SUO NOME
 - QUALE PEDINA HA SCELTO

```
typedef struct {  
    char nome[MAX_NOME_GIOCATORE];  
    char simbolo_pedina;  
} giocatore;
```



L'inserimento di una pedina



*Dobbiamo creare una sorta
di effetto gravita!*



L'inserimento di una pedina: codice C

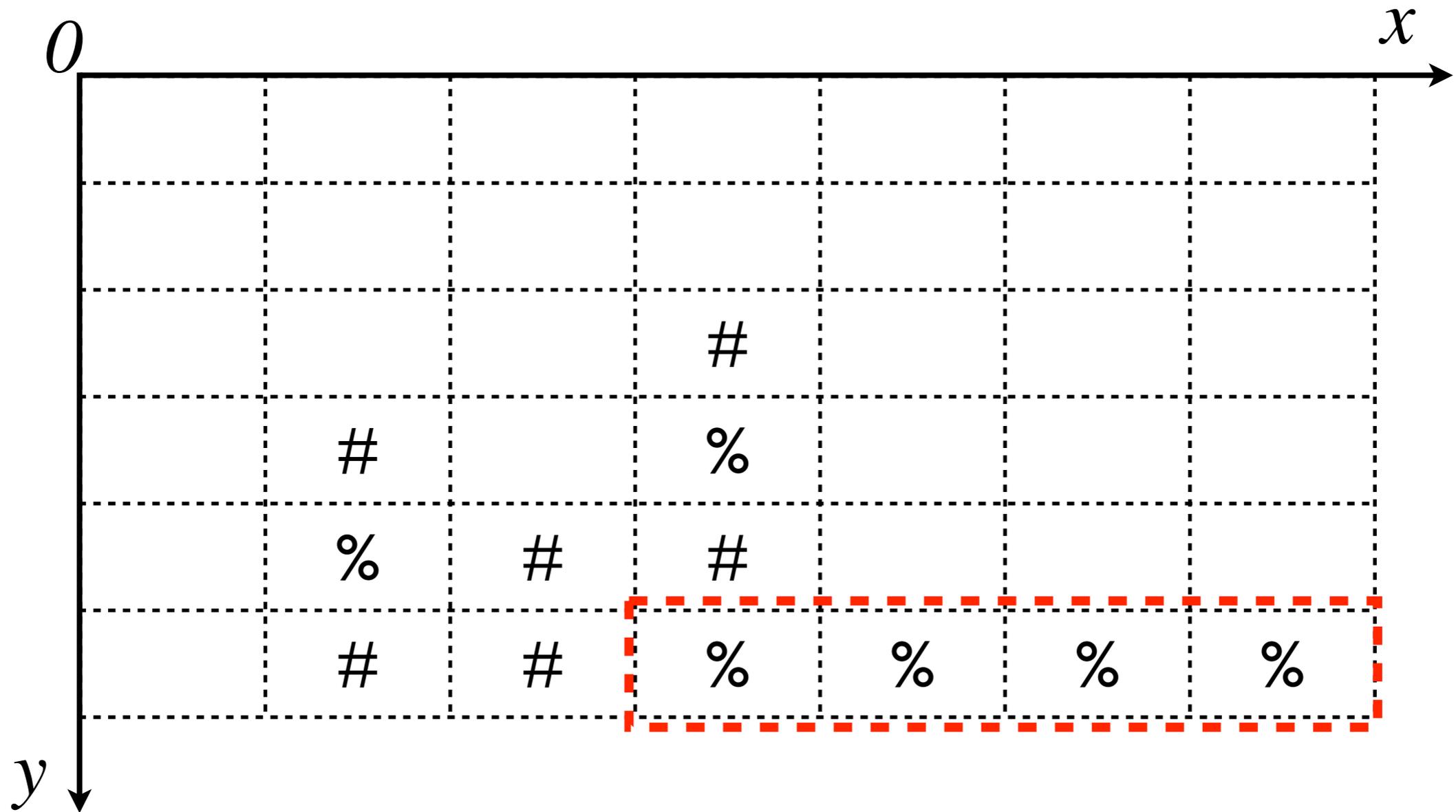
```
int inserisci_pedina(char schermo[SCREEN_W][SCREEN_H],
                    int colonna, char simbolo_pedina)
{
    int y;
    punto_schermo p;
    forma pedina;

    for (y = SCREEN_H-2; y >= 0; y--)
    {
        if (schermo[colonna][y] == ' ')
        {
            p = crea_punto_schermo(colonna, y, 0);
            pedina = genera_punto(simbolo_pedina);
            disegna_forma(pedina, p, schermo);
            return 1;
        }
    }

    return 0;
}
```



Verifica delle vincite



Dobbiamo controllare che, dato un simbolo, un vettore contiene 4 occorrenze di quest'ultimo **CONSECUTIVE!**



- Per verificare una vincita, dobbiamo prima di tutto saper controllare se, dato un vettore generico, esistono FORZA=4 occorrenze consecutive di un carattere '*occorrenza*' dato in ingresso

```
int controlla_vettore(char* vettore, int dim_vettore, char occorrenza)
```

- Con questa funzione potremo poi controllare sia vettori verticali, che orizzontali che obliqui



Controllo di un generico vettore

- Se ho un vettore A , e voglio controllare se qualcuno ha vinto, potrei usare il seguente metodo
- Parto dal secondo elemento e, per ogni elemento
 - Se l'elemento $A[i]$ considerato è uguale al carattere *occorrenza* ed è uguale all'elemento $A[i-1]$, incremento un contatore *occorrenze* ($occorrenze++$)
 - Se l'elemento $A[i]$ non è uguale ad *occorrenza* oppure l'elemento $A[i] \neq A[i-1]$, *occorrenze* viene riportato a 0 ($occorrenze=0$)
- Ogni qual volta che *occorrenze* arriva a $FORZA-1$, nel nostro caso 4-1, vale a dire 3, ho verificato che c'è una vincita
- Se non succede mai che $occorrenze == FORZA-1$, allora non c'è nessuna vincita



Controllo di un generico vettore: codice C

```
int controlla_vettore(char* vettore, int dim_vettore, char occorrenza)
{

    int i;
    int occorrenze = 0;

    for (i = 1; i < dim_vettore; i++){

        if (vettore[i] == vettore[i-1] && vettore[i] == occorrenza)
        {
            occorrenze++;
            if (occorrenze == FORZA-1)
                return 1;
        }
        else
        {
            occorrenze = 0;
        }
    }
    return 0;
}
```



Controllo vincita orizzontale

```
int controlla_orizzontale(char schermo[SCREEN_W][SCREEN_H],
                        char simbolo_pedina)
{
    int x,y;
    char vettore[SCREEN_W];
    int risultato = 0;

    for (y = SCREEN_H-2; y >= 0; y--)
    {
        for (x = 0; x < SCREEN_W; x++){
            vettore[x] = schermo[x][y];
        }

        risultato = controlla_vettore(vettore, SCREEN_W, simbolo_pedina);
        if (risultato == 1)
            return 1;
    }

    return 0;
}
```



Controlla vincita verticale

```
int controlla_verticale(char schermo[SCREEN_W][SCREEN_H],
                       char simbolo_pedina)
{
    int x,y;
    char vettore[SCREEN_W];
    int risultato = 0;

    for (x = 0; x < SCREEN_W; x++)
    {
        for (y = SCREEN_H-2; y >= 0; y--)
        {
            vettore[y] = schermo[x][y];
        }

        risultato = controlla_vettore(vettore, SCREEN_W, simbolo_pedina);
        if (risultato == 1)
            return 1;
    }

    return 0;
}
```

FORZA QUATTRO



**Tutte il materiale sarà
disponibile sul mio sito internet!**

alessandronacci.com

**Potete lasciare il vostro giudizio
qui:**

<http://tinyurl.com/IEIMExe2014>

See You Next Time!

