Corso di Informatica Medica

Esercitazione V

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

SIETE CARICHI OGGI?


READY...
SET...
GO

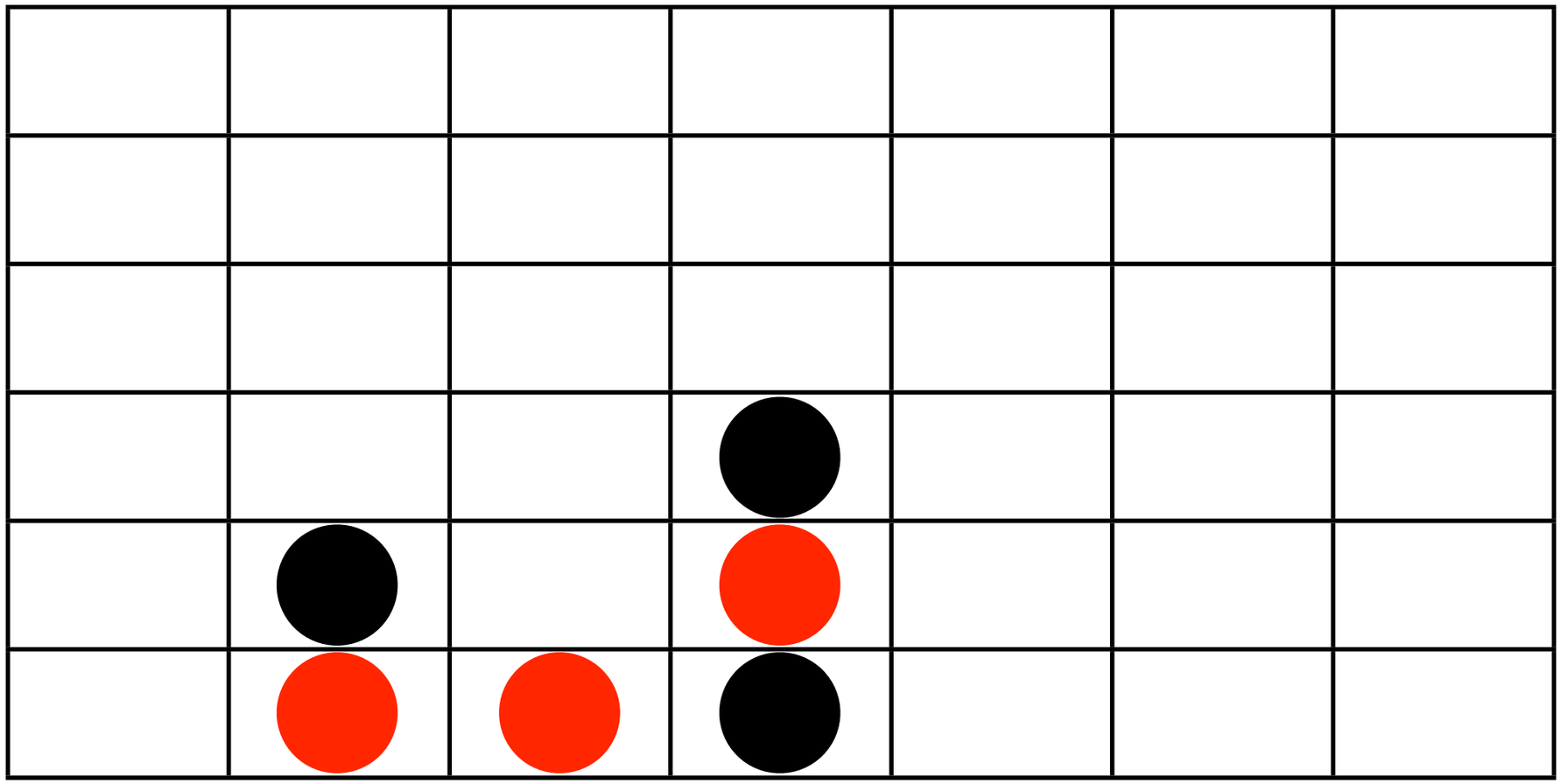
Un semplice videogioco

NELLA LEZIONE PRECEDENTE

- 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 vincintore

Forza 4: rappresentazione

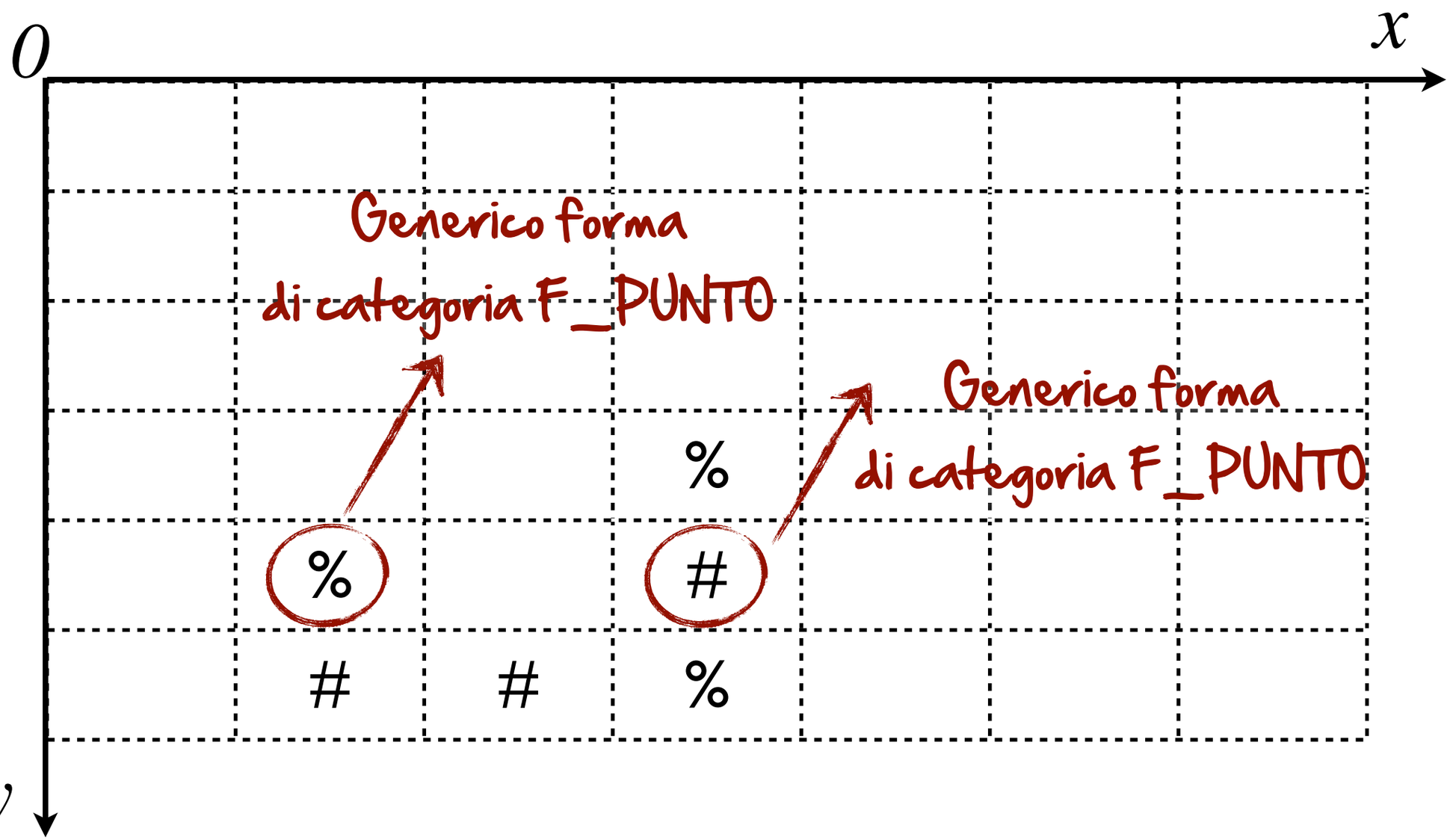
NELLA LEZIONE PRECEDENTE



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

Forza 4: rappresentazione

NELLA LEZIONE PRECEDENTE



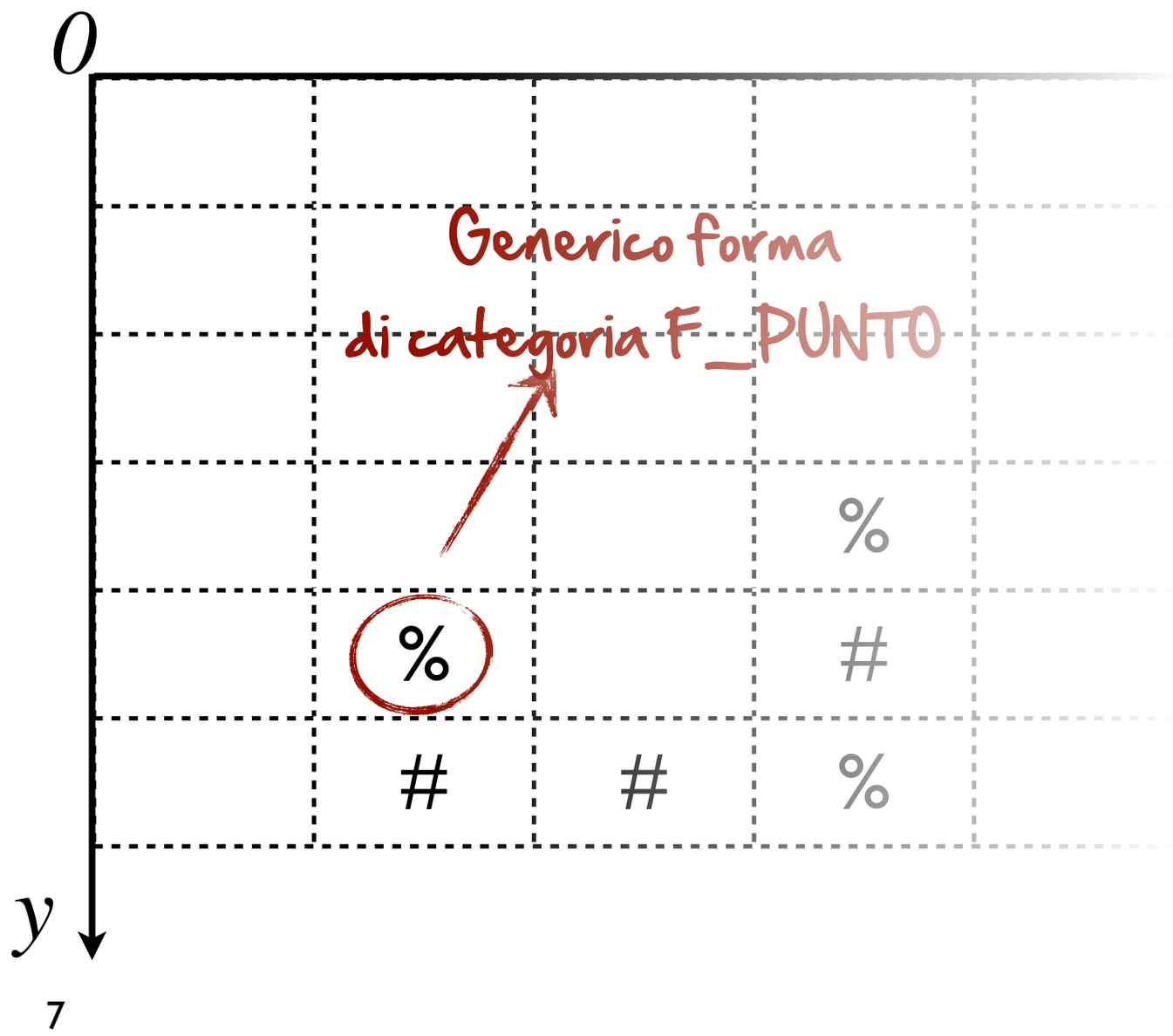
Quindi abbiamo una matrice,
descritta da due indici e
che contiene caratteri...

Fino ad ora...

NELLA LEZIONE PRECEDENTE

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

int inserisci_pedina(char schermo[SCREEN_W][SCREEN_H], int colonna, char pedina);
int controlla_vettore(char* vettore, int dim_vettore, char occorrenza);
```

- Ora, implementiamo il controllo vincita:
  - orizzontale
  - verticale
  - obliquo



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;

}
```

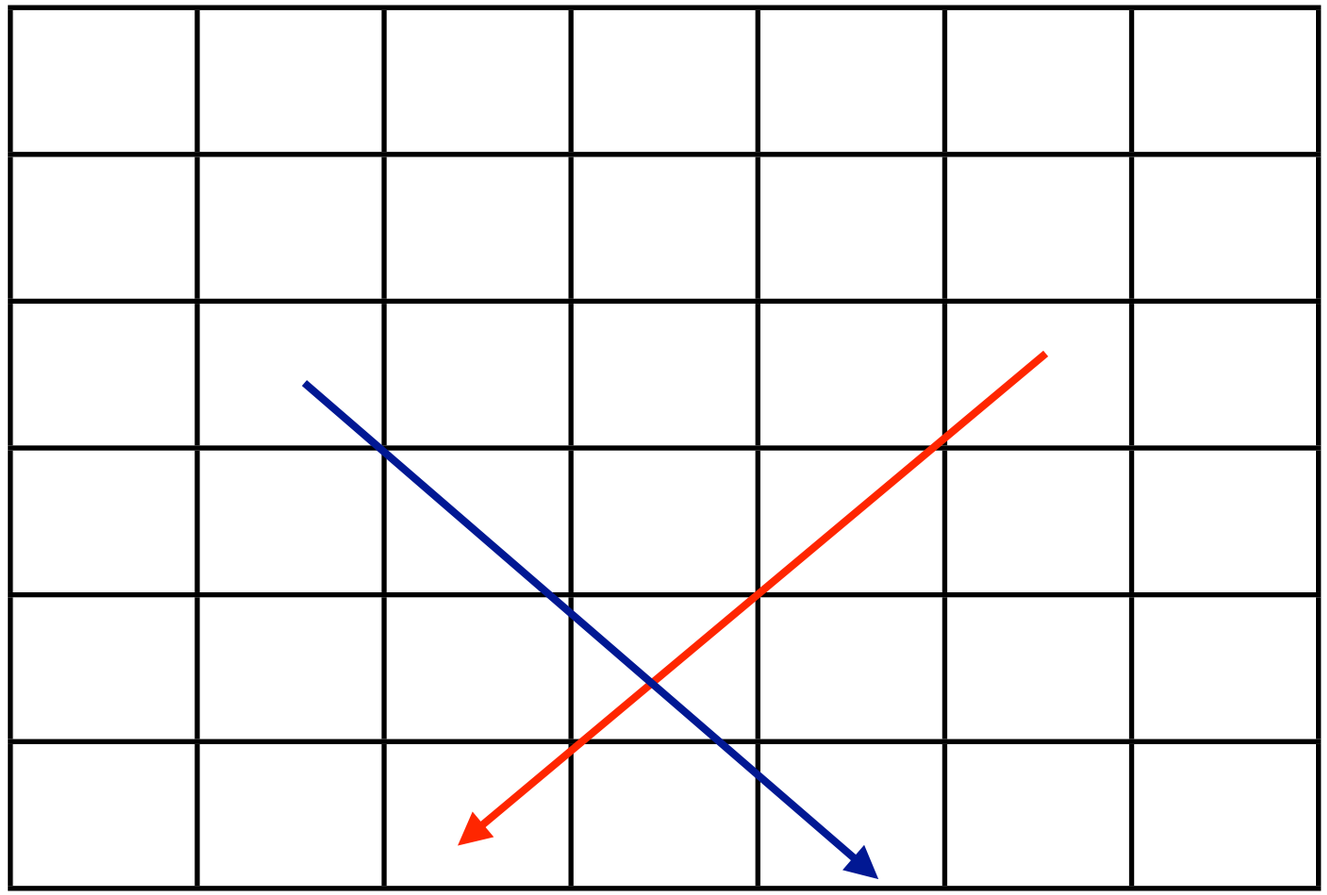
Controlla vincita verticale

```
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;

}
```

Controlla vincita obliquo

- In obliquo abbiamo due direzioni possibili

- Quindi creaiamo una funzione parametrica che dato un parametro direzione estra un vettore orizzontale...



```
int estra_controlla_vettore_obliquo(int x, int y, char schermo[SCREEN_W][SCREEN_H],
                                    char occorrenza, int orientamento)
```

Estazione array obliquo (codice C)

```c
int estra_controlla_vettore_obliquo(int x, int y, char schermo[SCREEN_W][SCREEN_H],
                                    char occorrenza, int orientamento)
{


    char vettore[DIAG];
    int curr_x, curr_y;
    int i = 0;

    do {
        curr_x = x + (orientamento) * i;
        curr_y = y +  i;
        vettore[i] = schermo[curr_x][curr_y];
        i++;

    } while (curr_x >= 0 &&  curr_x < SCREEN_W && curr_y >= 0 && curr_y < SCREEN_H);

    return controlla_vettore(vettore, i-1, occorrenza);

}
```

Controllo vincita in obliquo

- Quindi, usando la funzione appena creata ora possiamo controllare le vincite in obliquo

```
int controlla_obliquo(char schermo[SCREEN_W][SCREEN_H], char occorrenza)
{

    int x,y,i;
    int risultato_controllo = 0;

    for (y = 0; y < SCREEN_H; y++)
        for (x = 0; x < SCREEN_W; x++)
        {
            risultato_controllo = estra_controlla_vettore_obliquo(x,y,schermo,occorrenza, 1);
            if (risultato_controllo) return 1;

            risultato_controllo = estra_controlla_vettore_obliquo(SCREEN_W-x,y,schermo,occorrenza, -1);
            if (risultato_controllo) return 1;
        }



    return 0;
}
```

Controllo di una qualsiasi vincita

- Usando le funzioni precedenti possiamo controllare una qualsiasi vincita

```
int controlla_vincita(char schermo[SCREEN_W][SCREEN_H], char simbolo_pedina)
{

}
```

Controllo di una qualsiasi vincita

- Usando le funzioni precedenti possiamo controllare una qualsiasi vincita

```
int controlla_vincita(char schermo[SCREEN_W][SCREEN_H], char simbolo_pedina)
{
    int result = 0;

    result += controlla_orizzontale(schermo, simbolo_pedina);
    result += controlla_verticale(schermo, simbolo_pedina);
    result += controlla_obliquo(schermo, simbolo_pedina);

    return result;
}
```

- Prima di scrivere il main(), ci mancano ancora un paio di funzioni comode...

```
// Restituisce una variabile 'giocatore' dato il nome del giocatore e il simbolo della pedina
giocatore crea_giocatore(char nome[MAX_NOME_GIOCATORE], char pedina);

// Richiede a schermo dove inserire pedina. Restisce '1' se tutto ok, '0' in caso di errore
int richiedi_inserimento_pedina();

// Inserisce la pedina 'pedina' nella colonna 'colonna' dello campo di gioco 'schermo'
int inserisci_pedina(char schermo[SCREEN_W][SCREEN_H], int colonna, char pedina);
```

Crea giocatore: codice C

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

Crea giocatore: codice C

```
giocatore crea_giocatore(char nome[MAX_NOME_GIOCATORE], char simbolo_pedina)
{

    giocatore g;
    strcpy(g.nome, nome);
    g.simbolo_pedina = simbolo_pedina;
    return g;
}
```

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

Richiedi inserimento pedina: codice C

```c
int richiedi_inserimento_pedina()
{
    int colonna;
    printf("In quale colonna vuoi inserire la pedina? ==> ");
    scanf("%d", &colonna);
    return colonna;
}
```

Inserisci pedina: codice C

```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;

}
```

- Ed ora, scriviamo il main:
  - Dobbiamo creare due giocatori
  - Dobbiamo gestire l'alternanza dei due giocatori
  - Dobbiamo richiedere dove inserire la pedina
  - Dobbiamo controllare la vincita

```c
int main(){

    char schermo[SCREEN_W][SCREEN_H];
    giocatore giocatori[NUMERO_GIOCATORI];

    int giocatore_corrente = 0;
    int vincitore = -1;
    int colonna_richiesta;

    // Disegnamo un quadrato
    giocatori[0] = crea_giocatore("MARIO", 'o');
    giocatori[1] = crea_giocatore("PAOLO", 'x');

    inizializza_schermo(schermo);
    disegna_schermo(schermo);
    while(vincitore == -1)
    {
        printf ("Giocatore %d \n", giocatore_corrente + 1);
        colonna_richiesta = richiedi_inserimento_pedina();
        if (inserisci_pedina(schermo, colonna_richiesta, giocatori[giocatore_corrente].simbolo_pedina)){
            if (controlla_vincita(schermo, giocatori[giocatore_corrente].simbolo_pedina))
            {
                printf("Il giocatore %d, %s ha vinto!\n\n",
                    giocatore_corrente + 1, giocatori[giocatore_corrente].nome);
                return 0;
            }
            giocatore_corrente = (giocatore_corrente + 1) % NUMERO_GIOCATORI;
        } else {
            printf("La colonna inserita non è valida. Seleziona un'altra colonna.\n");
            aspetta_invio();
            aspetta_invio();    // RISOLVERE QUESTO PROBLEMA
        }
        disegna_schermo(schermo);
    }


}
```

FORZA QUATTRO

Automobili: seconda versione

- Rappresentare in C una automobile. Nel nostro caso, una automobile è descritta da un nome, un costo, un colore, da un insieme di componenti e da un libretto di circolazione.
- Un componente ha un nome, un costo ed una categoria. Le categorie possibili sono TRAZIONE, MULTIMEDIA, SICUREZZA
- Il libretto di circolazione riporta invece l'anno e la provincia di immatricolazione e in che classe Euro rientra.

- Il programma deve poter permettere la creazione di auto e la stampa a schermo di tutti i dati relativi ad un'auto
- Deve poter permettere inoltre di modificare il nome dell'auto
- Deve poter calcolare il costo totale per la produzione dell'auto

Automobili: Le strutture dati - Codice C

```c
typedef enum {TRAZIONE, MULTIMEDIA, SICUREZZA} tipi_categoria;

typedef struct {
    int anno_immatricolazione;
    char provincia[STR_LEN];
    int classe_euro;
} libretto_circolazione;

typedef struct {
    char nome[STR_LEN];
    double costo;
    tipi_categoria categoria;
} componente;

typedef struct {
    char nome[STR_LEN];
    double costo;
    char colore[STR_LEN];
    int numero_componenti;
    componente* componenti;
    libretto_circolazione libretto;
} automobile;
```

Potete lasciare il vostro giudizio qui:

http://tinyurl.com/IEIMExe2013

Tutte il materiale sarà disponibile sul mio sito internet:

alessandronacci.com