

Corso di Informatica Medica

Esercitazione III

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



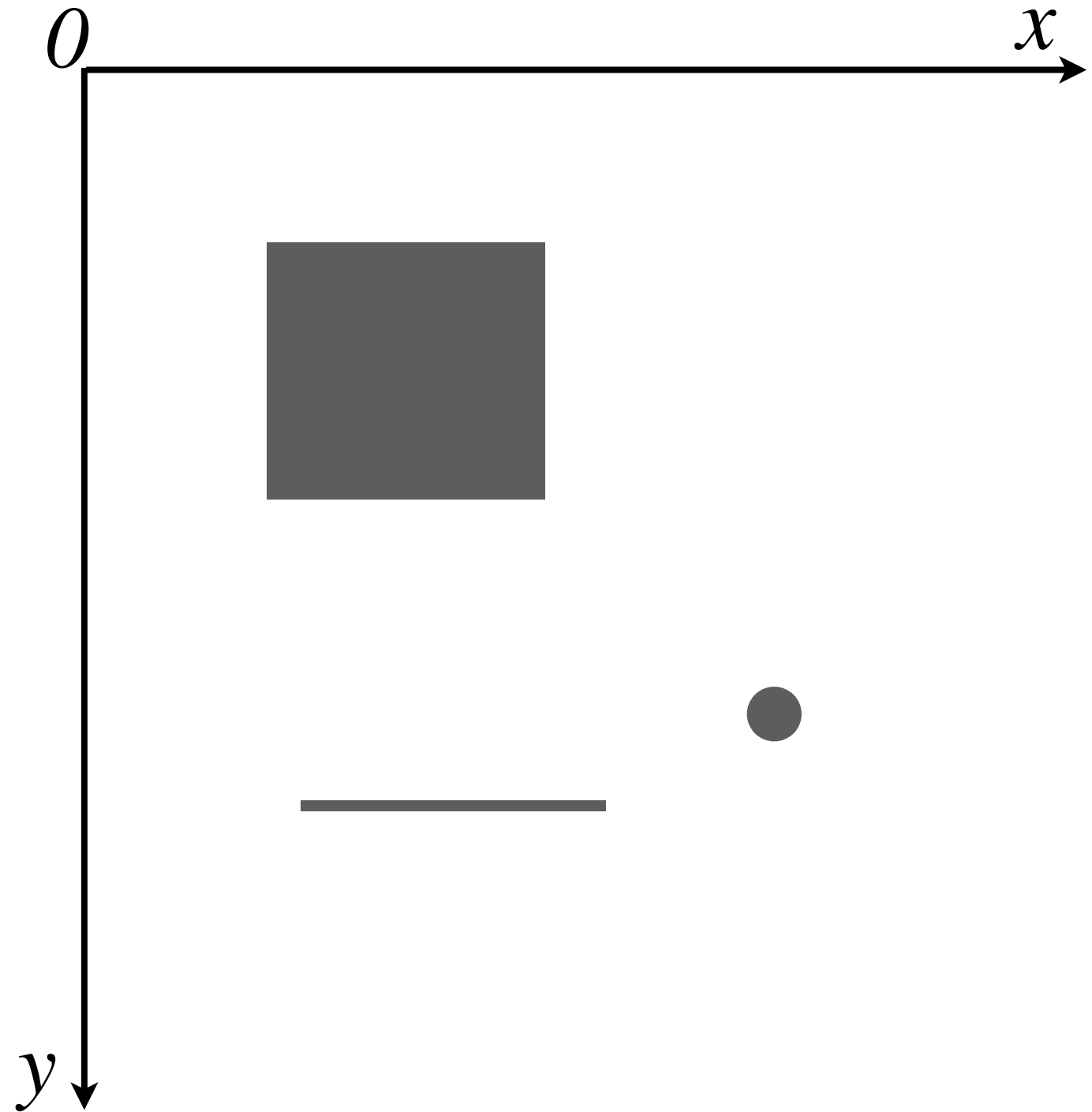


WARNING

**QUANTO FATTO OGGI
VERRA' UTILIZZATO
NELLE PROSSIME LEZIONI!**

- Scrivere in C un programma che rappresenti un piano cartesiano
- Il programma deve poter rappresentare e visualizzare a schermo PUNTI, LINEE e QUADRATI
- Deve essere inoltre possibile manipolare le forme create (spostarle, cancellarle, ingrandirle, etc..)
- Il programma deve poter rappresentare la curva di una funzione di secondo grado:

$$y = a_2 \cdot x^2 + a_1 \cdot x + a_0$$



La rappresentazione

- Per visualizzare a schermo abbiamo a disposizione il solo terminale:
- Formato da RIGHE e COLONNE di caratteri ASCII
- Ci *arrangiamo* con quello che abbiamo

ATTENZIONE !

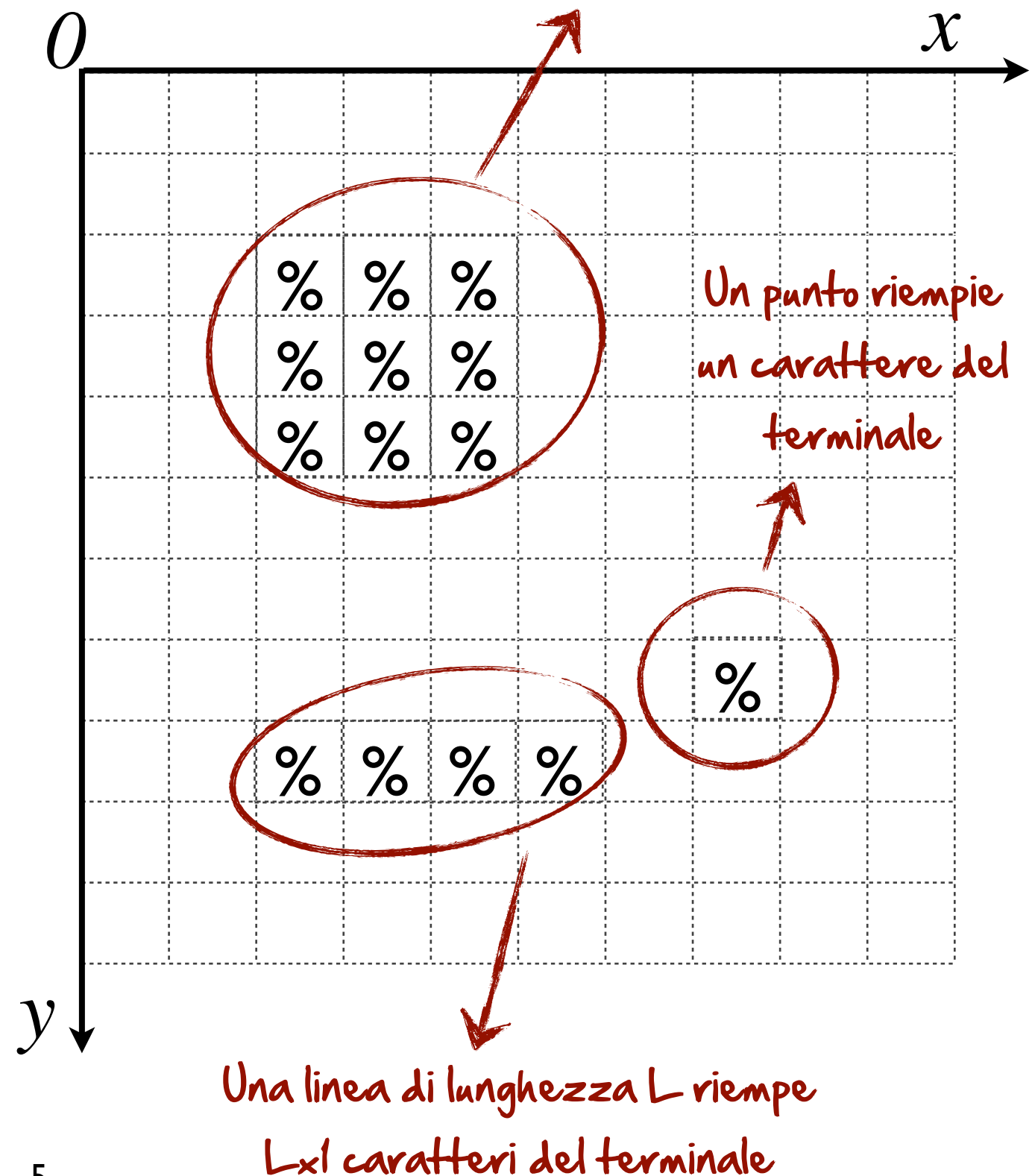
NON POTREMO RAPPRESENTARE I BORDI DELLE FIGURE!

CI LIMITEREMO A RAPPRESENTARE IL CONTENUTO DELLE FIGURE CON DEI CARATTERI

NELLA LEZIONE PRECEDENTE

Un quadrato di lato L riempie

$L \times L$ caratteri del terminale





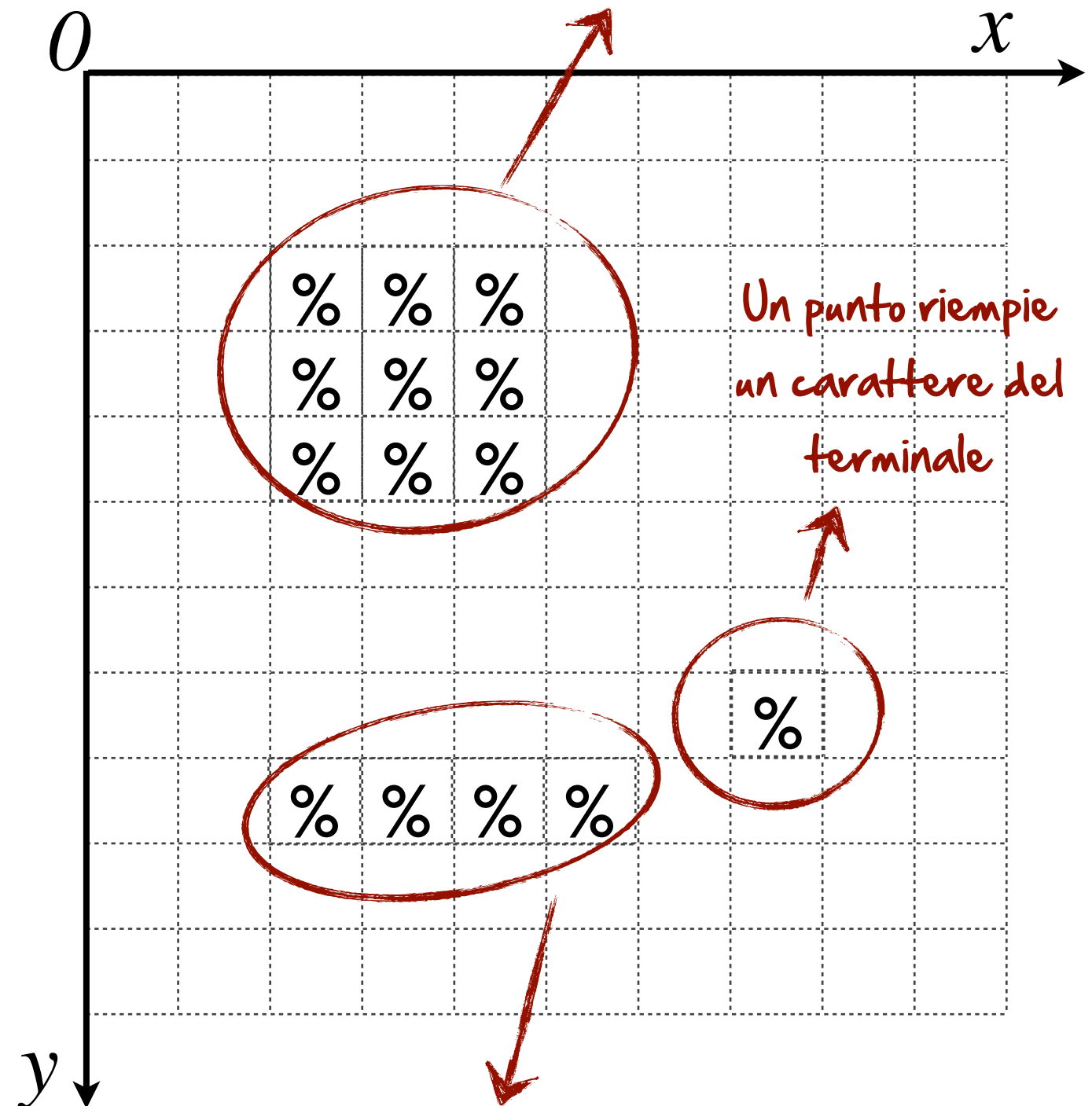
Il codice C

```
typedef enum {F_PUNTO, F_LINEA,  
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;  
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;  
  
typedef struct {  
    int x;  
    int y;  
    char valore;  
} punto_schermo;  
  
typedef struct {  
    punto_schermo pixels[MAX_PUNTI_FORMA];  
    int numero_pixel;  
    categoria_forma categoria;  
} forma;  
  
// Funzioni per la visualizzazione a video  
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);  
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);  
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);  
void aspetta_invio();  
void pulisci_terminale();  
  
// Modifica delle matriche che rappresenta lo schermo  
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);  
punto_schermo crea_punto_schermo(int x, int y, char valore);  
  
// Generazione delle forme  
forma genera_quadrato(int dim, char carattere);  
forma genera_linea(int dim, direzione direzione_linea, char carattere);  
forma genera_punto(char carattere);  
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);  
  
int main(){  
  
    char schermo[SCREEN_W][SCREEN_H];  
    forma quadrato;  
    punto_schermo p;  
    forma linea_or;  
    forma linea_vr;  
    forma punto;  
  
    // Disegniamo un quadrato  
    inizializza_schermo(schermo);  
    pulisci_terminale();  
  
    quadrato = genera_quadrato(4, '#');  
    p = crea_punto_schermo(1,1,0);  
  
    disegna_forma(quadrato, p, schermo);  
    disegna_schermo(schermo);  
  
    aspetta_invio();  
}
```

NELLA LEZIONE PRECEDENTE

Un quadrato di lato L riempie

$L \times L$ caratteri del terminale



Implementiamo le funzioni

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresentano lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);

int main(){

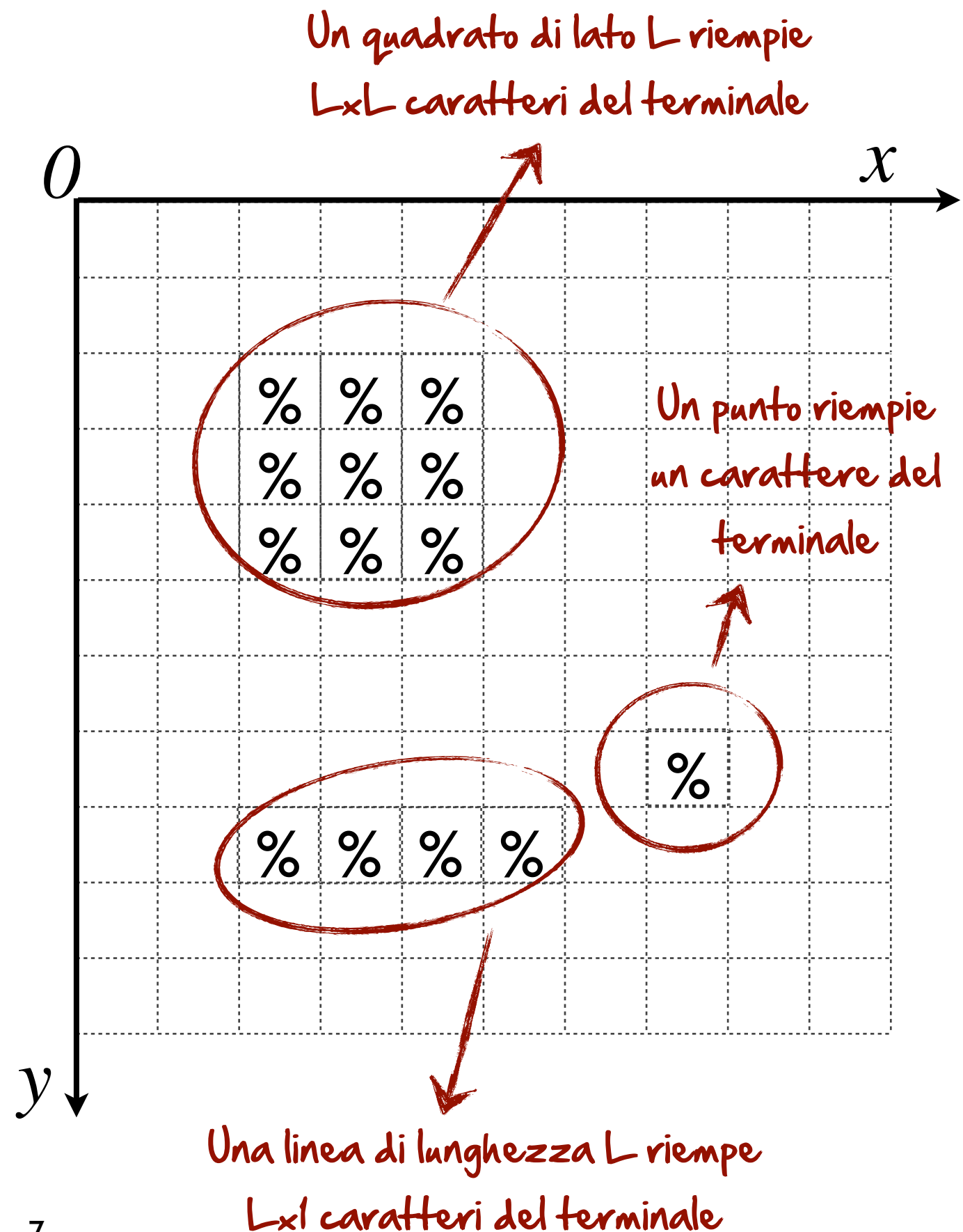
    char schermo[SCREEN_W][SCREEN_H];
    forma quadrato;
    punto_schermo p;
    forma linea_or;
    forma linea_vr;
    forma punto;

    // Disegniamo un quadrato
    inizializza_schermo(schermo);
    pulisci_terminale();

    quadrato = genera_quadrato(4, '#');
    p = crea_punto_schermo(1,1,0);

    disegna_forma(quadrato, p, schermo);
    disegna_schermo(schermo);

    aspetta_invio();
}
```



Implementiamo le funzioni

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresentano lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);

int main(){

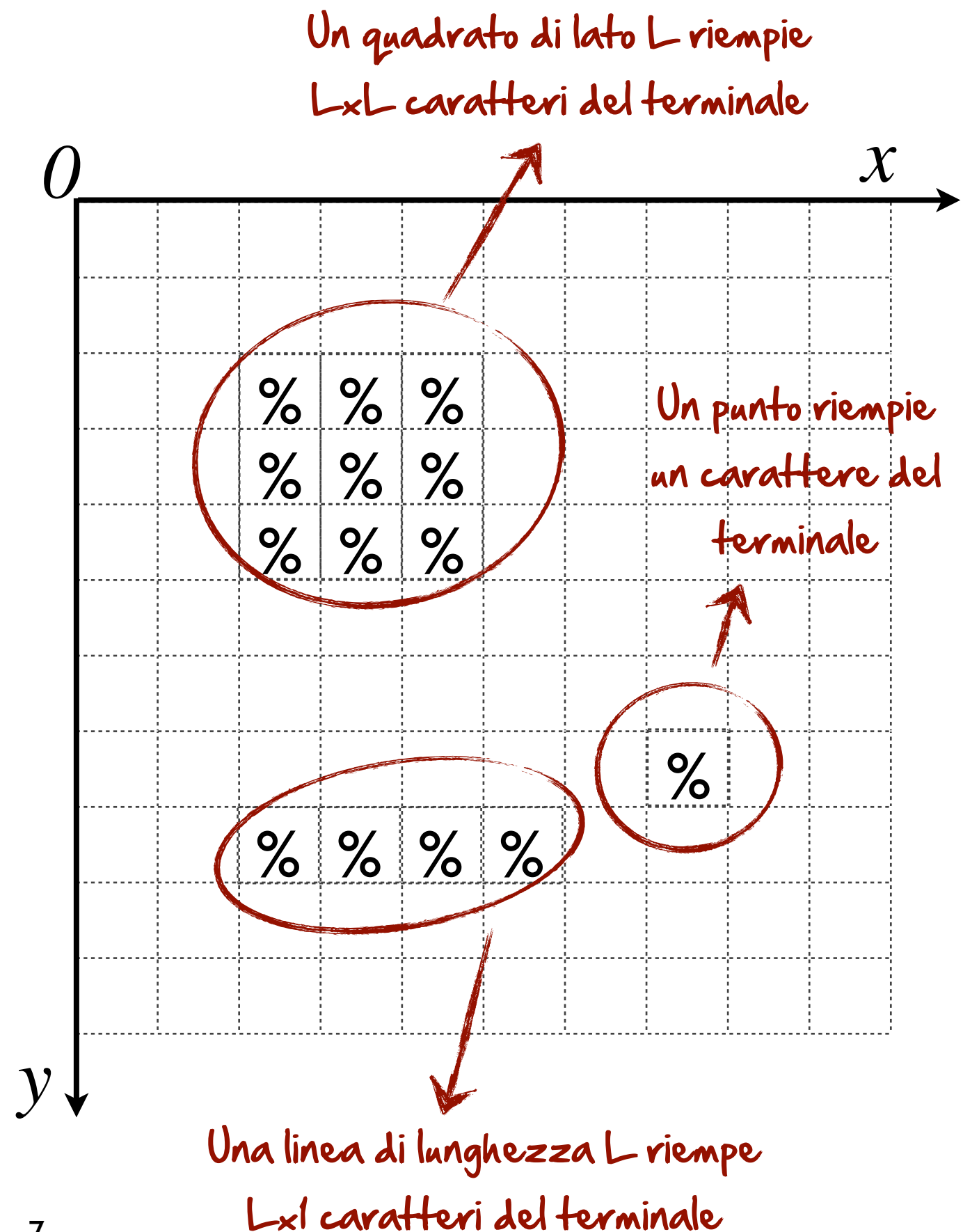
    char schermo[SCREEN_W][SCREEN_H];
    forma quadrato;
    punto_schermo p;
    forma linea_or;
    forma linea_vr;
    forma punto;

    // Disegniamo un quadrato
    inizializza_schermo(schermo);
    pulisci_terminale();

    quadrato = genera_quadrato(4, '#');
    p = crea_punto_schermo(1,1,0);

    disegna_forma(quadrato, p, schermo);
    disegna_schermo(schermo);

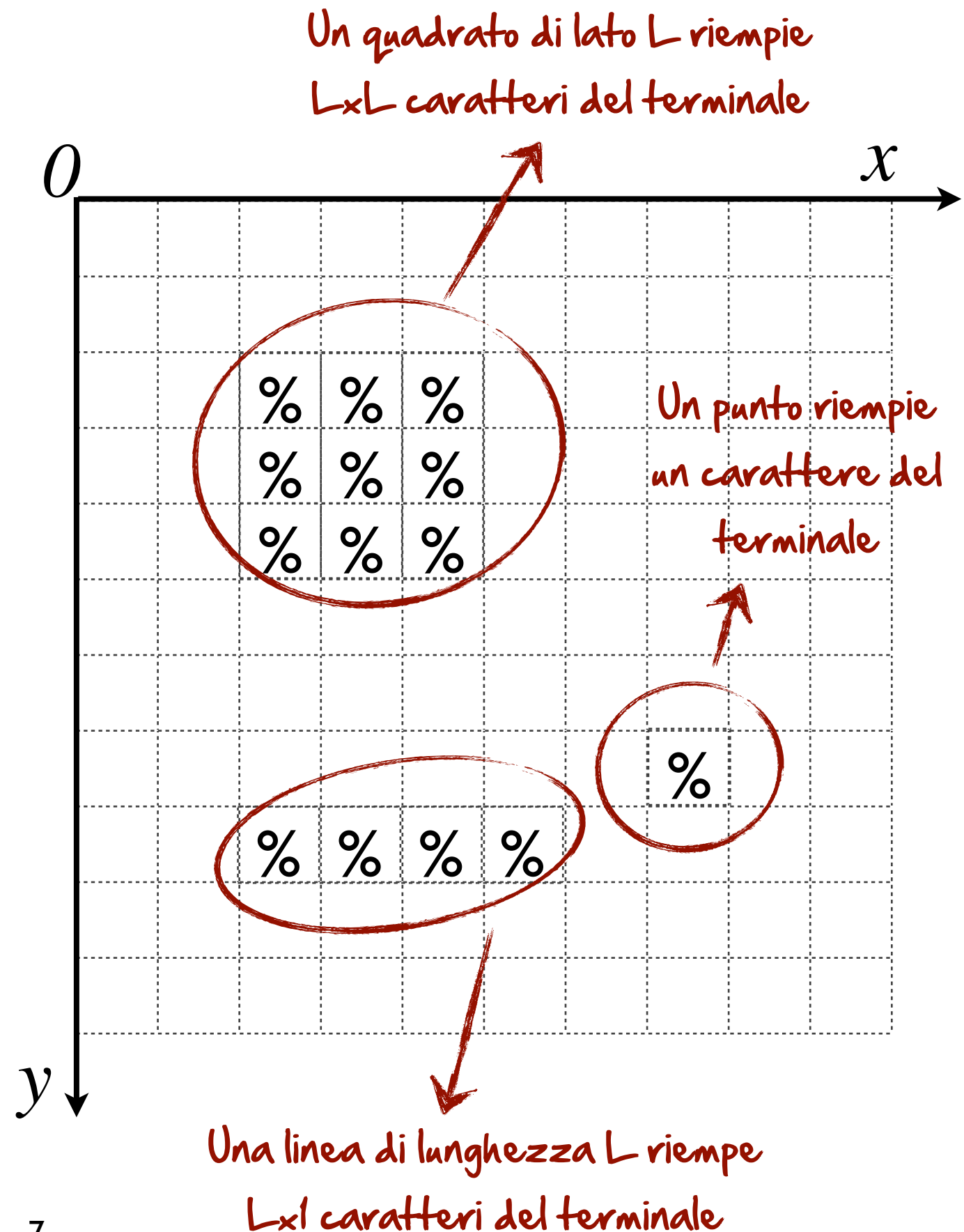
    aspetta_invio();
}
```





Implementiamo le funzioni

```
typedef enum {F_PUNTO, F_LINEA,  
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;  
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;  
  
typedef struct {  
    int x;  
    int y;  
    char valore;  
} punto_schermo;  
  
typedef struct {  
    punto_schermo pixels[MAX_PUNTI_FORMA];  
    int numero_pixel;  
    categoria_forma categoria;  
} forma;  
  
// Funzioni per la visualizzazione a video  
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);  
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);  
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);  
void aspetta_invio();  
void pulisci_terminale();  
  
// Modifica delle matriche che rappresentano lo schermo  
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);  
punto_schermo crea_punto_schermo(int x, int y, char valore);  
  
// Generazione delle forme  
forma genera_quadrato(int dim, char carattere);  
forma genera_linea(int dim, direzione direzione_linea, char carattere);  
forma genera_punto(char carattere);  
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);  
  
int main(){  
  
    char schermo[SCREEN_W][SCREEN_H];  
    forma quadrato;  
    punto_schermo p;  
    forma linea_or;  
    forma linea_vr;  
    forma punto;  
  
    // Disegniamo un quadrato  
    inizializza_schermo(schermo);  
    pulisci_terminale();  
  
    quadrato = genera_quadrato(4, '#');  
    p = crea_punto_schermo(1,1,0);  
  
    disegna_forma(quadrato, p, schermo);  
    disegna_schermo(schermo);  
  
    aspetta_invio();  
}
```



Implementiamo le funzioni

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresentano lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);

int main(){

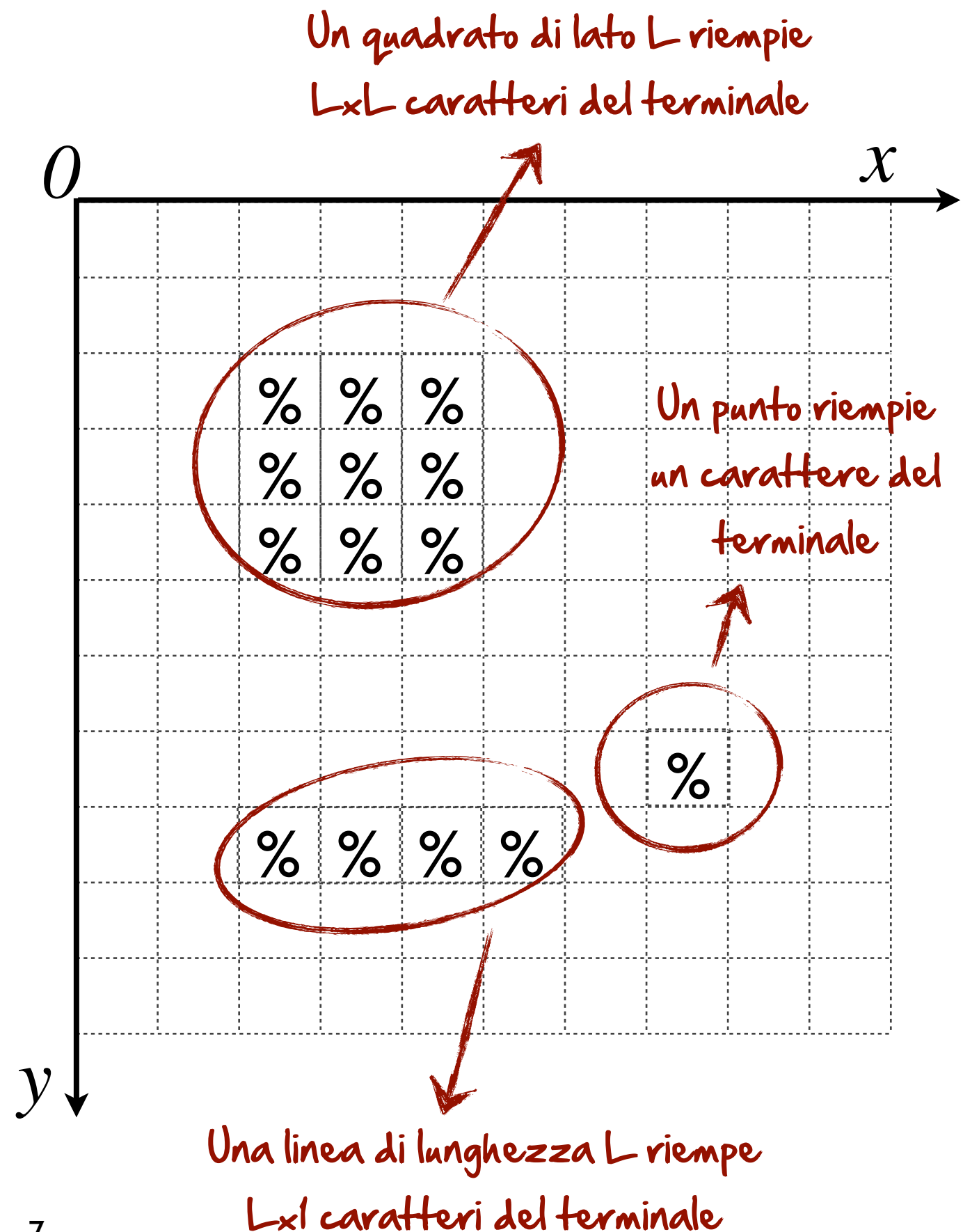
    char schermo[SCREEN_W][SCREEN_H];
    forma quadrato;
    punto_schermo p;
    forma linea_or;
    forma linea_vr;
    forma punto;

    // Disegniamo un quadrato
    inizializza_schermo(schermo);
    pulisci_terminale();

    quadrato = genera_quadrato(4, '#');
    p = crea_punto_schermo(1,1,0);

    disegna_forma(quadrato, p, schermo);
    disegna_schermo(schermo);

    aspetta_invio();
}
```



Implementiamo le funzioni

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresentano lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);

int main(){

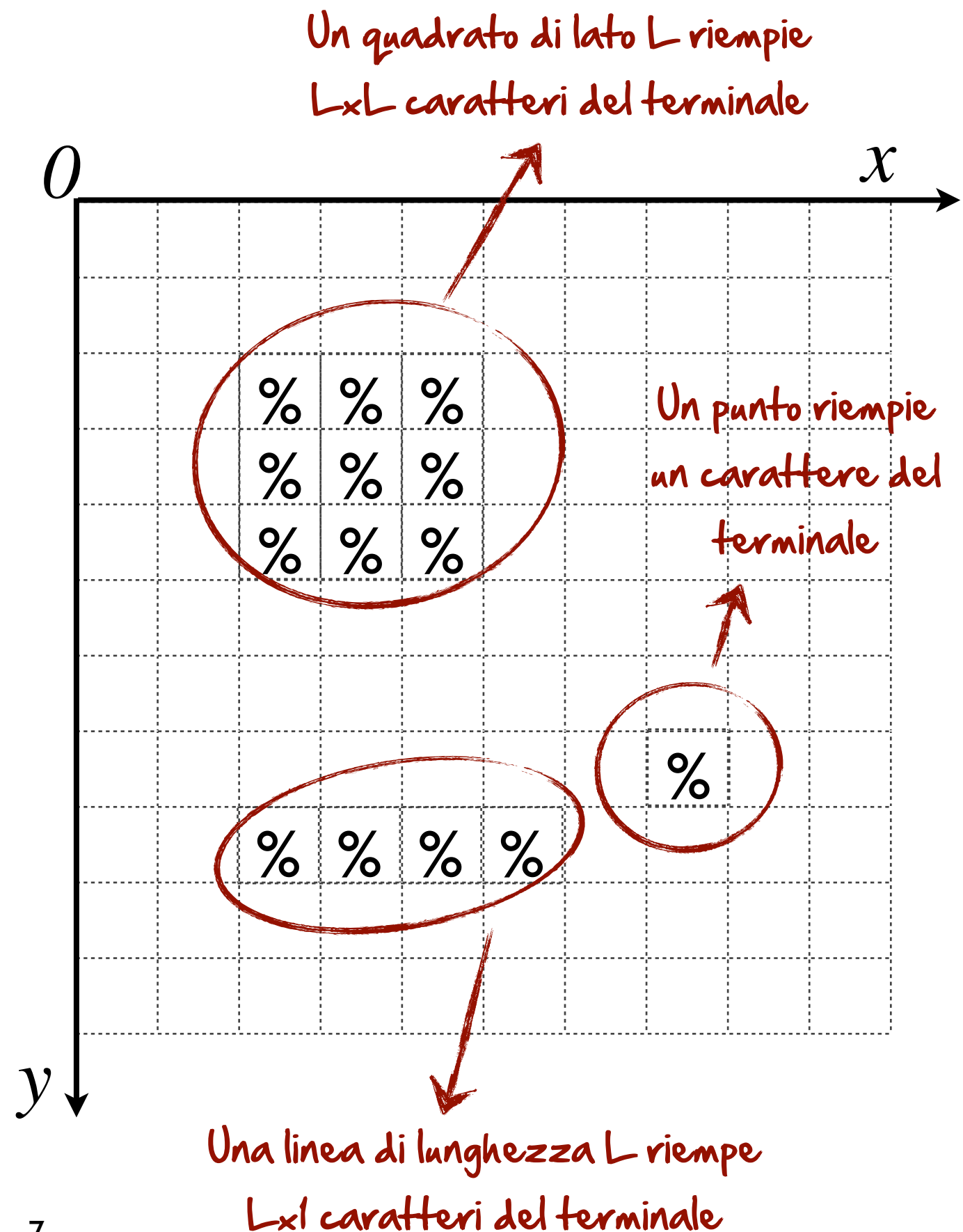
    char schermo[SCREEN_W][SCREEN_H];
    forma quadrato;
    punto_schermo p;
    forma linea_or;
    forma linea_vr;
    forma punto;

    // Disegniamo un quadrato
    inizializza_schermo(schermo);
    pulisci_terminale();

    quadrato = genera_quadrato(4, '#');
    p = crea_punto_schermo(1,1,0);

    disegna_forma(quadrato, p, schermo);
    disegna_schermo(schermo);

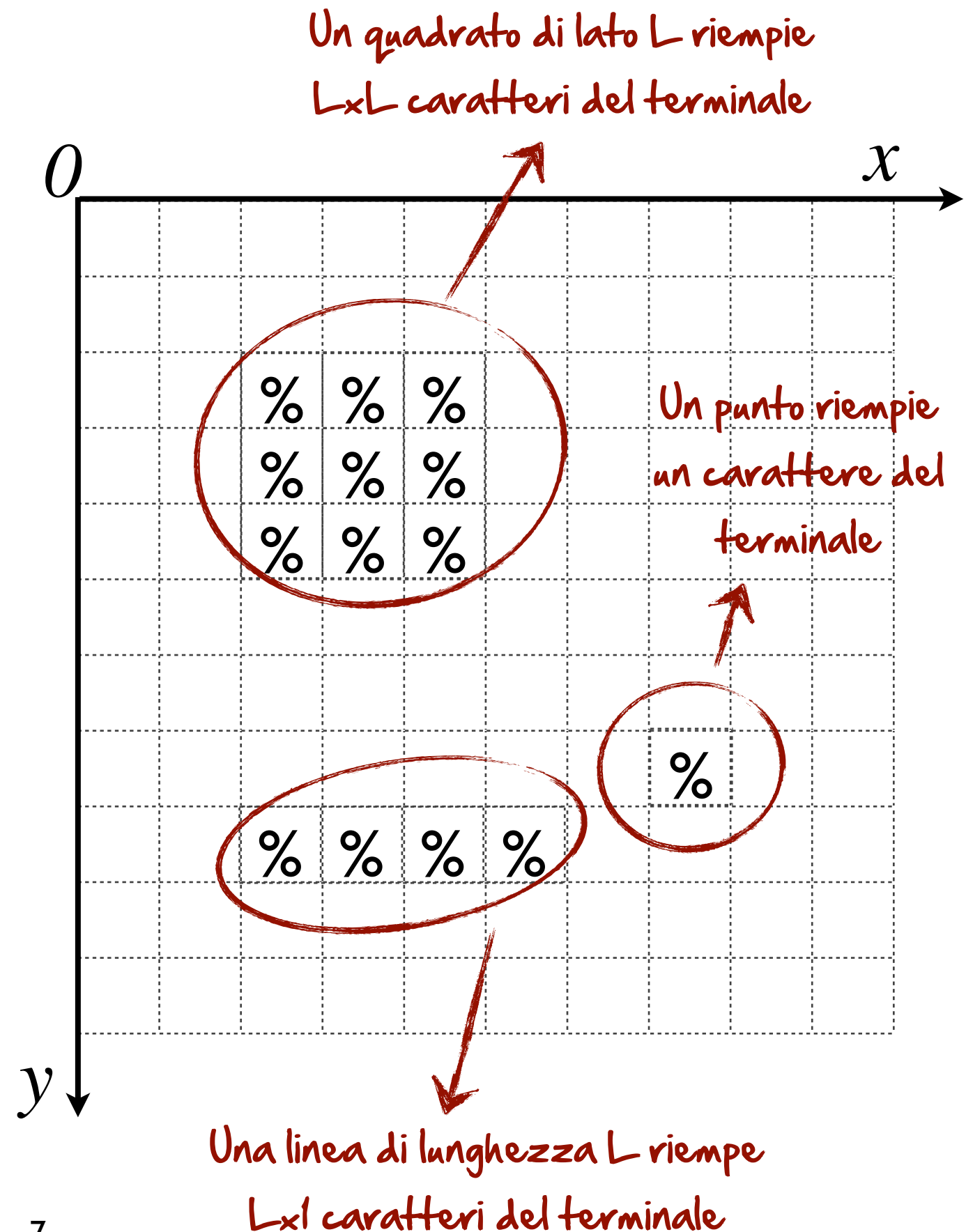
    aspetta_invio();
}
```





Implementiamo le funzioni

```
typedef enum {F_PUNTO, F_LINEA,  
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;  
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;  
  
typedef struct {  
    int x;  
    int y;  
    char valore;  
} punto_schermo;  
  
typedef struct {  
    punto_schermo pixels[MAX_PUNTI_FORMA];  
    int numero_pixel;  
    categoria_forma categoria;  
} forma;  
  
// Funzioni per la visualizzazione a video  
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);  
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);  
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);  
void aspetta_invio();  
void pulisci_terminale();  
  
// Modifica delle matriche che rappresenta lo schermo  
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);  
punto_schermo crea_punto_schermo(int x, int y, char valore);  
  
// Generazione delle forme  
forma genera_quadrato(int dim, char carattere);  
forma genera_linea(int dim, direzione direzione_linea, char carattere);  
forma genera_punto(char carattere);  
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);  
  
int main(){  
  
    char schermo[SCREEN_W][SCREEN_H];  
    forma quadrato;  
    punto_schermo p;  
    forma linea_or;  
    forma linea_vr;  
    forma punto;  
  
    // Disegniamo un quadrato  
    inizializza_schermo(schermo);  
    pulisci_terminale();  
  
    quadrato = genera_quadrato(4, '#');  
    p = crea_punto_schermo(1,1,0);  
  
    disegna_forma(quadrato, p, schermo);  
    disegna_schermo(schermo);  
  
    aspetta_invio();  
}
```



Inizializza schermo

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

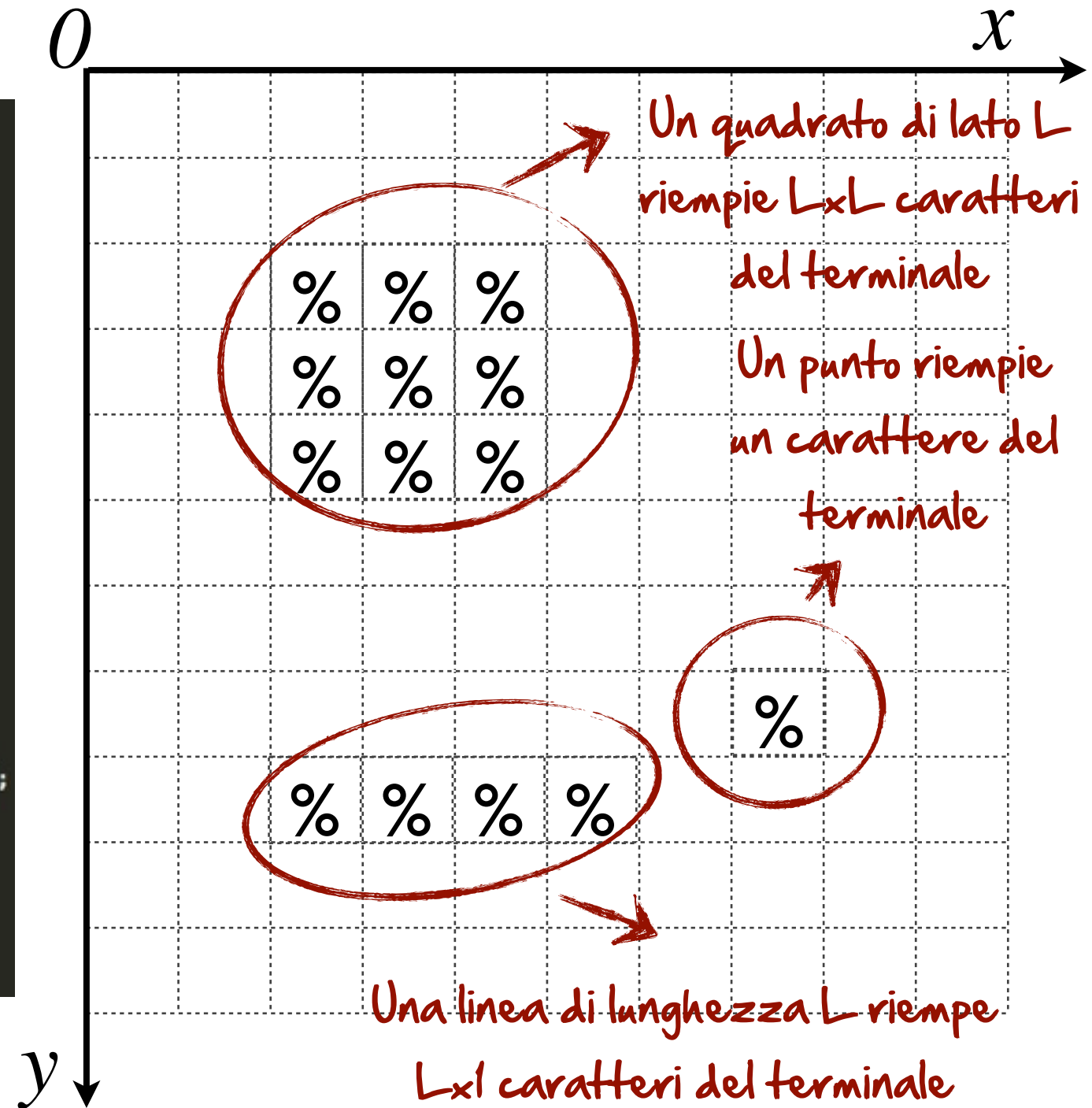
typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresenta lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);
```



```
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
```

Riempie di caratteri vuoti la matrice in ingresso 'schermo'

Inserisci bordi

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

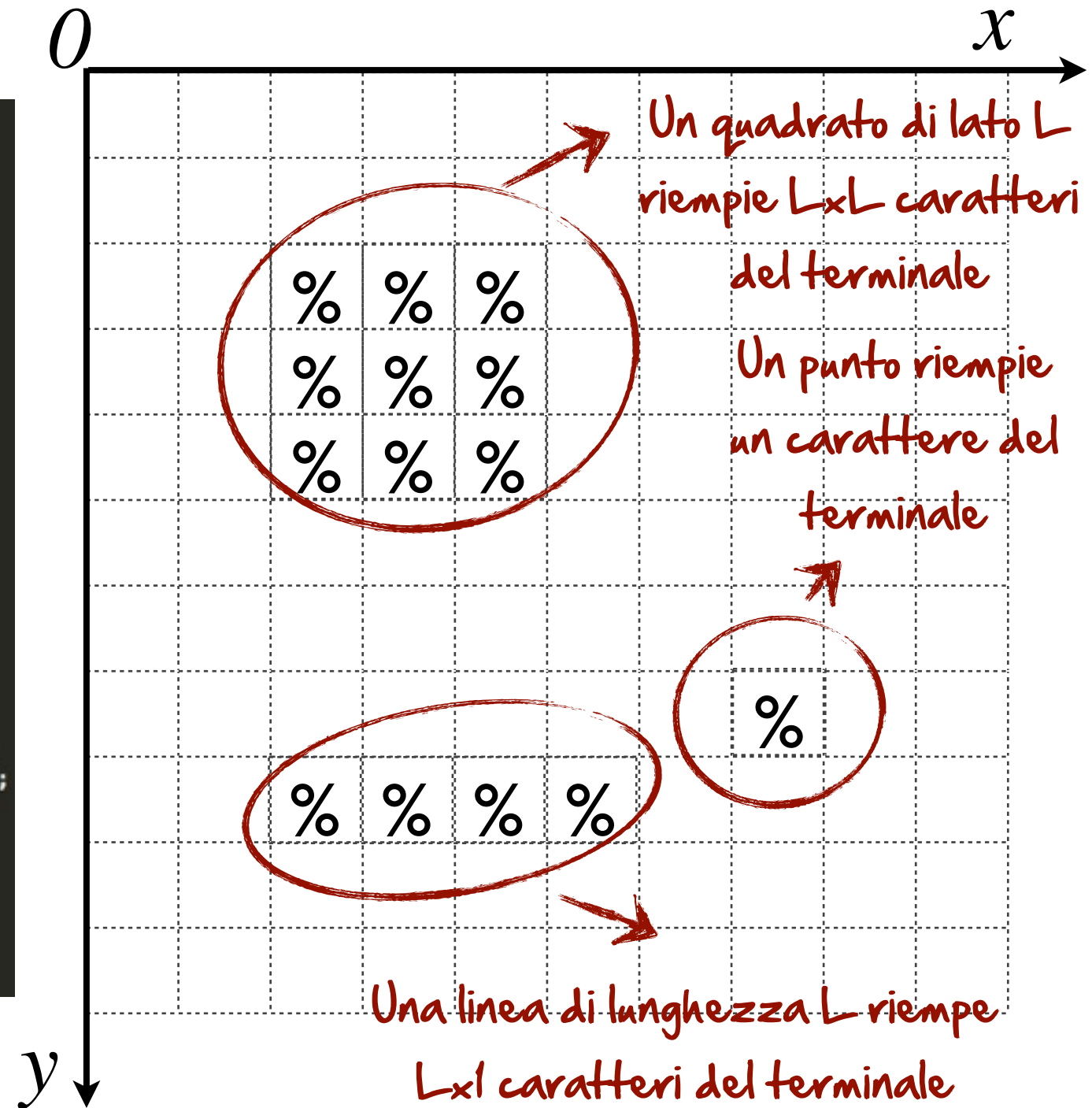
typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresenta lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);
```



```
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
```

Inserisce il carattere '%' sui bordi della matrice 'schermo'

Disegna schermo

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

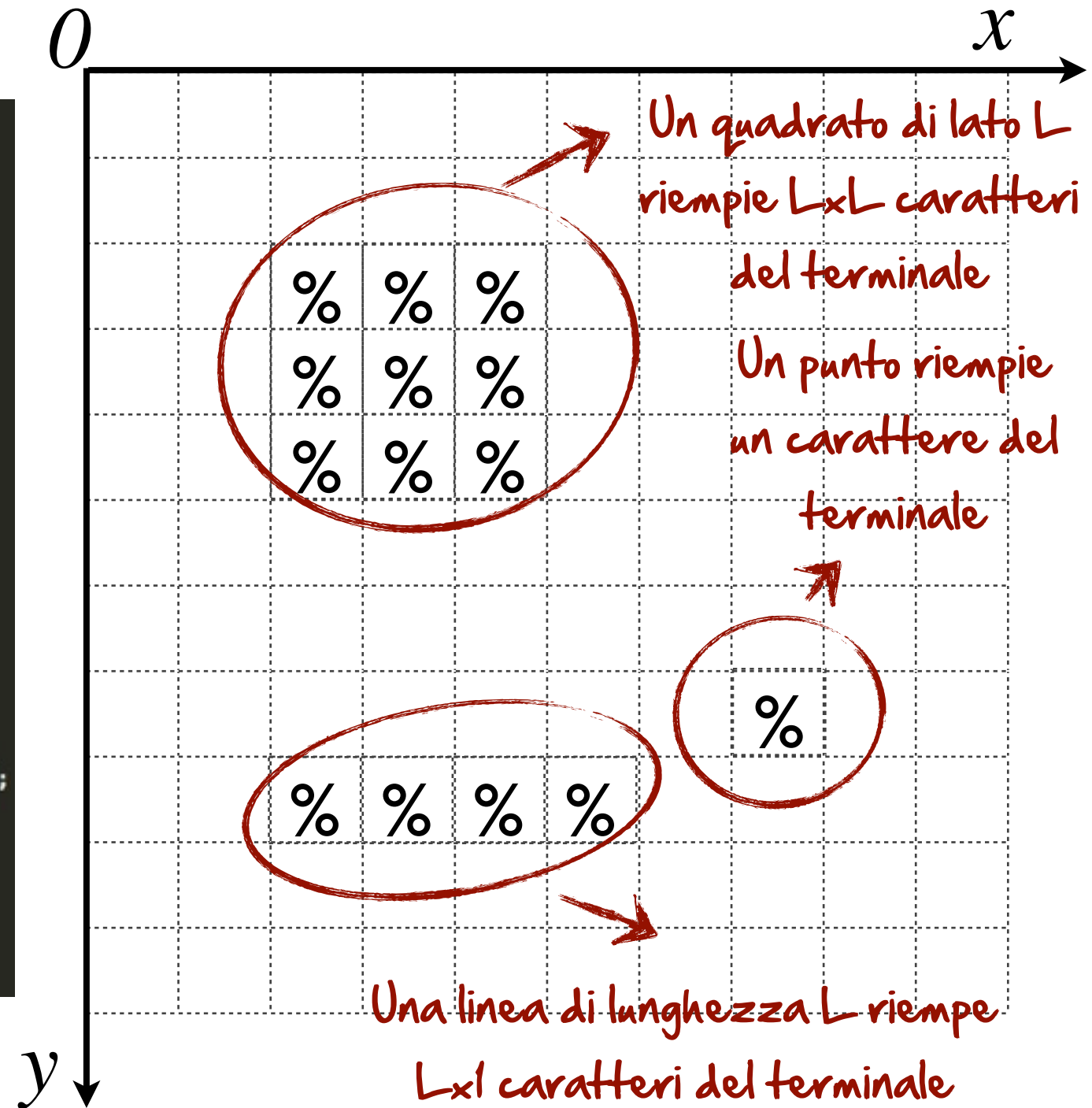
typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresenta lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);
```



```
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
```

Disegna a schermo la matrice in ingresso 'schermo'

Aspetta invio

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

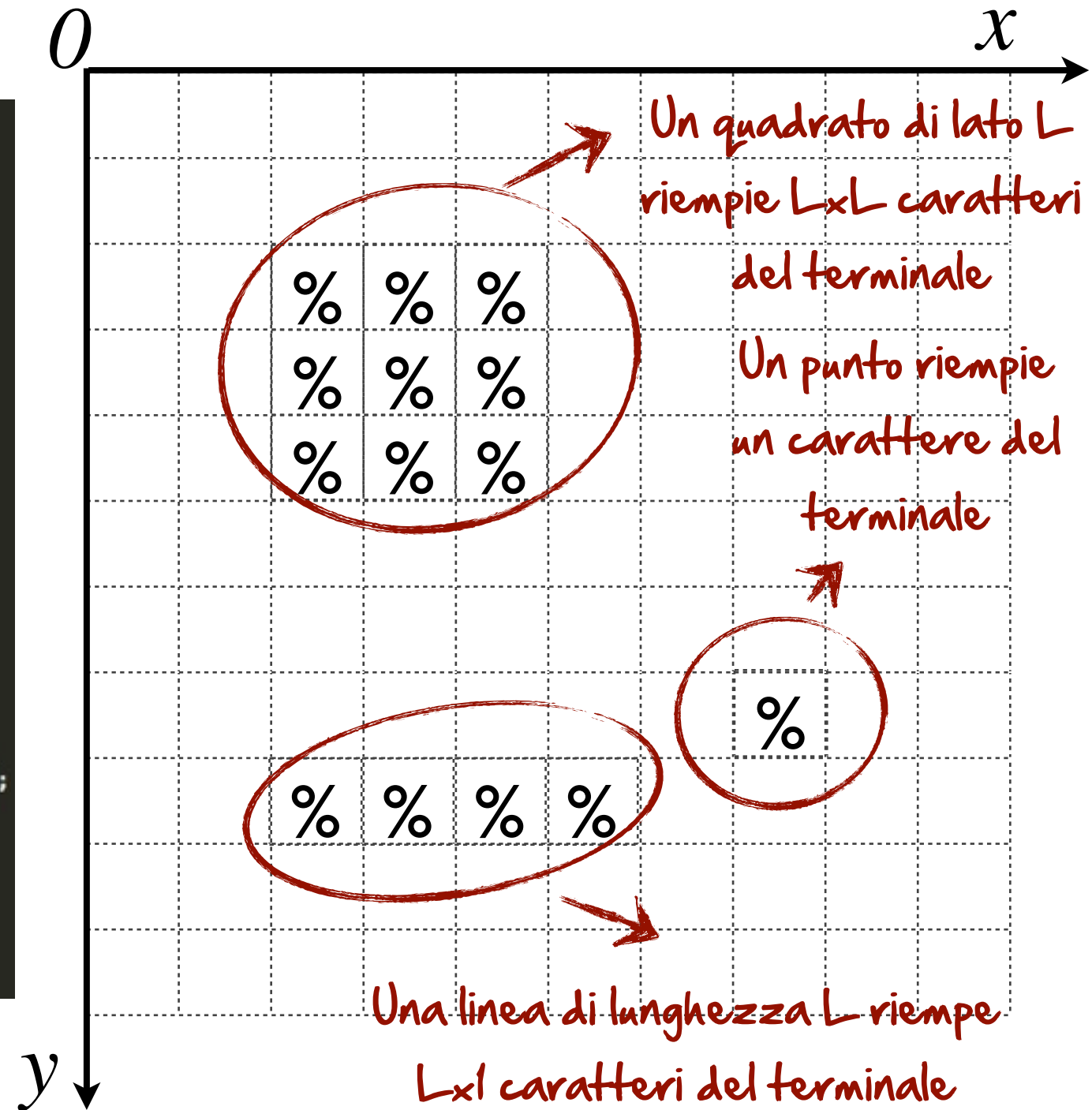
typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresenta lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);
```



```
void aspetta_invio();
```

Quando lanciata, mette il programma in attesa di un INVIO da parte dell'utente

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

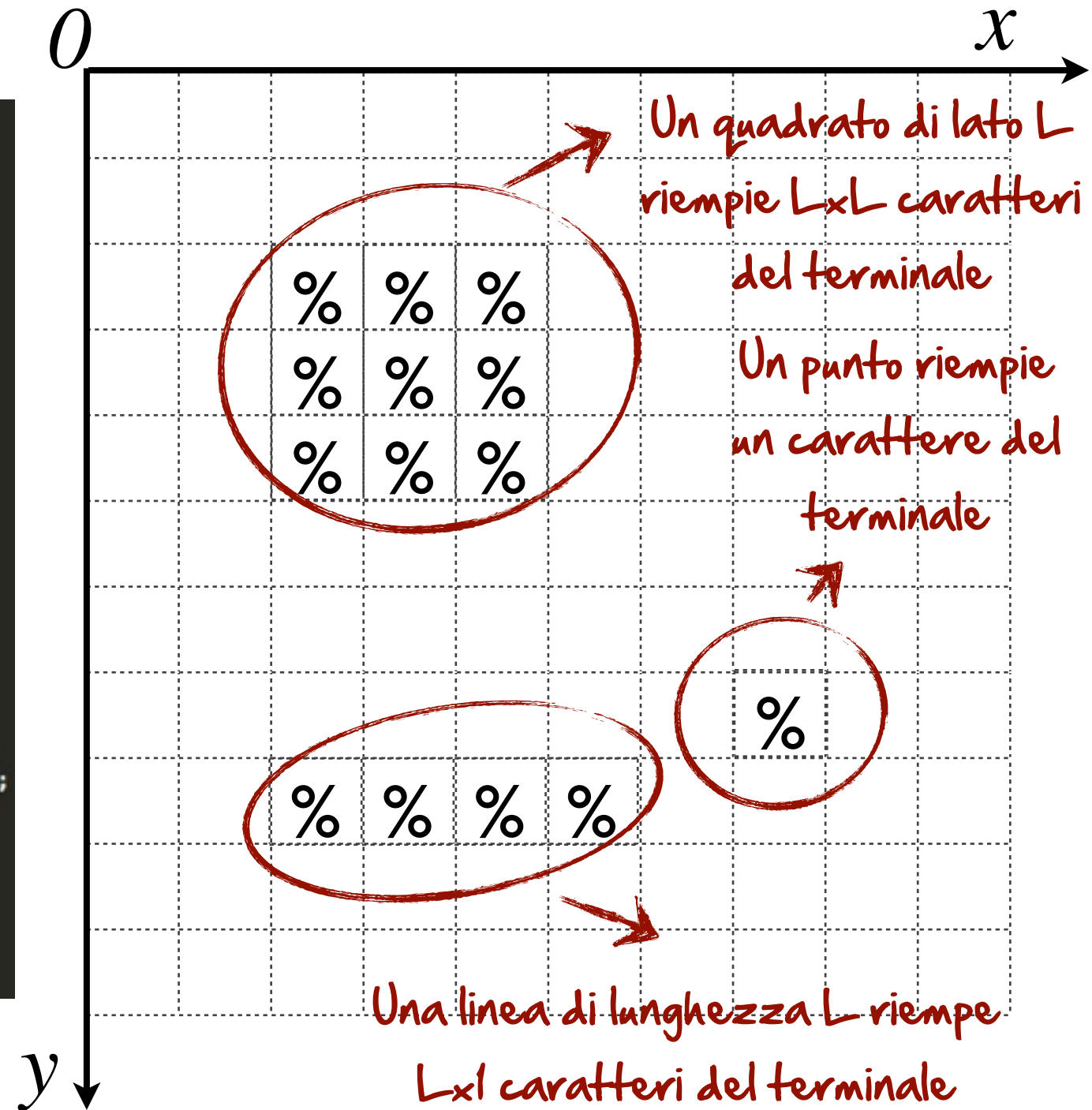
typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresenta lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);
```



```
void pulisci_terminale();
```

Cancella il contenuto del terminale su cui viene stampato l'output del programma

Disegna forma

```
typedef enum {F_PUNTO, F_LINEA,
             F_POLIGONO_QUADRILATERO, F_GENERICA} categoria_forma;
typedef enum {D_VERTICALE, D_ORIZZONTALE} direzione;

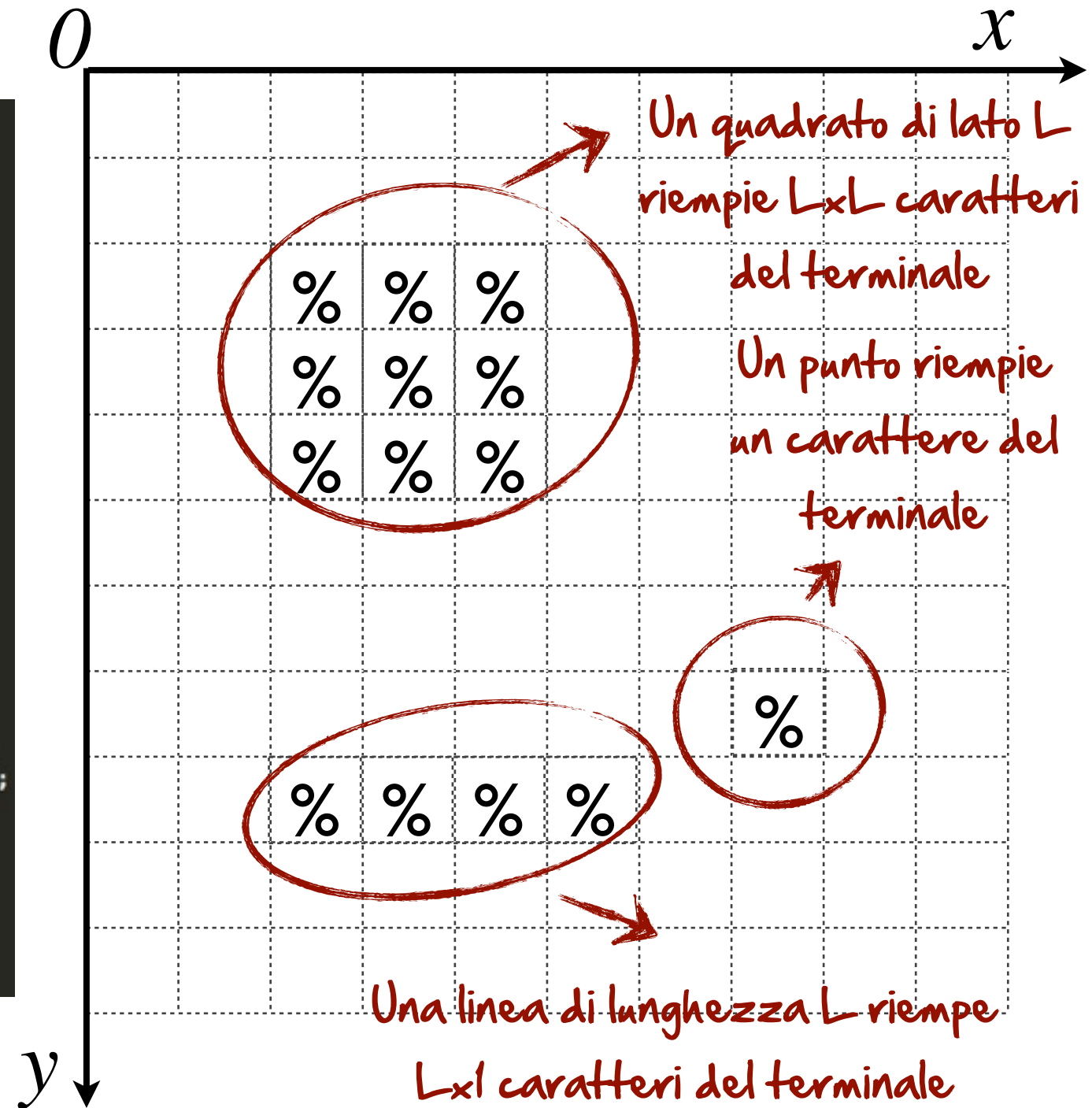
typedef struct {
    int x;
    int y;
    char valore;
} punto_schermo;

typedef struct {
    punto_schermo pixels[MAX_PUNTI_FORMA];
    int numero_pixel;
    categoria_forma categoria;
} forma;

// Funzioni per la visualizzazione a video
void inizializza_schermo(char schermo[SCREEN_W][SCREEN_H]);
void inserisci_bordi(char schermo[SCREEN_W][SCREEN_H]);
void disegna_schermo(char schermo[SCREEN_W][SCREEN_H]);
void aspetta_invio();
void pulisci_terminale();

// Modifica delle matriche che rappresenta lo schermo
void disegna_forma(forma f, punto_schermo p, char schermo[SCREEN_W][SCREEN_H]);
punto_schermo crea_punto_schermo(int x, int y, char valore);

// Generazione delle forme
forma genera_quadrato(int dim, char carattere);
forma genera_linea(int dim, direzione direzione_linea, char carattere);
forma genera_punto(char carattere);
forma genera_polinomio(int a2, int a1, int a0, int dim, char carattere);
```



```
void disegna_forma(forma f, punto_schermo p,
                  char schermo[SCREEN_W][SCREEN_H]);
```

Disegna una data un generica forma 'f' nella posizione 'p' dello schermo 'schermo'

Potete lasciare il vostro giudizio qui:

<http://tinyurl.com/IEIMExe2013>

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

alessandronacci.com

See You Next Time!

