ASCII Art

Progetto d'esame del corso di "Programmazione e Laboratorio"

Docenti: Paolo Boldi e Violetta Lonati

Traccia valida a partire dall'appello di settembre 2009

Introduzione

Obiettivo del progetto è realizzare un *interprete* per un semplice linguaggio di manipolazione di immagini monocromatiche rappresentate tramite matrici di caratteri.

Iniziamo con qualche semplice definizione. Un *pixel* è un elemento di immagine che può trovarsi di volta in volta in uno di due possibili stati: *acceso* o *spento*. Una *bitmap* di *dimensione* $n > 0$ è una matrice di $n \times n$ pixel; chiameremo *coordinate* del pixel nella riga r e colonna c di una bitmap la coppia di indici (r,c) (dove sia le righe che le colonne sono numerate a partire da 0). Una *figura* è data da un insieme finito di coordinate, alcuni esempi sono:

- il *segmento verticale* di dimensione h con coordinate (r,c), dato dall'insieme $\{(r,c),(r+1,c),\ldots,(r+h-1,c)\}$,
- il *segmento orizzontale* di dimensione l con coordinate (r,c), dato dall'insieme $\{(r,c),(r,c+1),\ldots,(r,c+l-1)\}$,
- il *rettangolo* di dimensioni $l \times h$ con coordinate (r,c), dato dall'unione dei quattro segmenti: due verticali di dimensione h e coordinate (r,c) e $(r,c+l-1)$ e due orizzontali di dimensione l con coordinate (r,c) e $(r+h-1,c)$,
- il *timbro* della matrice[1] $A = (a_{i,j})$ con coordinate (r,c), dato dall'insieme $\{(r+i,c+j) \mid a_{i,j} \neq 0\}$.

Data una bitmap e una figura, *disegnare* la figura nella bitmap significa accendere i pixel della bitmap corrispondenti alle coordinate della figura (eventualmente ignorando le coordinate cui non corrispondono pixel della bitmap). In particolare, il disegno del timbro della matrice A con coordinate (r,c) può essere definito informalmente come segue: per prima cosa si "sovrappone" la matrice A alla bitmap in modo che l'elemento di riga 0 e colonna 0 della matrice si trovi "sopra" il pixel di coordinate (r,c) della bitmap, quindi si accendono i pixel della bitmap che si trovano "sotto" gli elementi non nulli della matrice.

Concludiamo l'introduzione osservando come una bitmap di dimensione n può essere semplicemente raffigurata tramite n righe di n caratteri ciascuna in cui il carattere nella colonna c della riga r rappresenti il pixel di coordinate (r,c), adottando la convenzione di indicare con uno spazio i pixel spenti e con un asterisco quelli accesi.

Ad esempio, la raffigurazione seguente corrisponde ad una bitmap di dimensione 4 (con tutti i pixel inizialmente spenti)

	0	1	2	3
0	*	␣	␣	␣
1	*	␣	*	*
2	*	␣	␣	␣
3	*	␣	␣	␣

in cui sono stati disegnati i segmenti verticali di dimensione 4 e coordinate $(0,0)$ e il segmento orizzontale di dimensione 2 e coordinate $(1,2)$ (i numeri di riga e colonna nella prima riga e prima colonna della tabella sono riportati solo per facilitare la lettura e non fanno parte della raffigurazione della bitmap che è data solo dagli asterischi `*` e dagli spazi, che qui sono rappresentati come ␣ per aumentarne la leggibilità).

[1] Per uniformità con quanto avviene nelle bitmap, gli indici di riga e colonna delle matrici qui partono da 0.

I comandi dell'interprete

L'interprete gestisce una bitmap di dimensioni assegnate ed esegue alcune manipolazioni su di essa tra le quali, ad esempio, il disegno di alcune figure. In più, l'interprete gestisce un insieme di matrici (di varie dimensioni) che possono essere usate per i timbri; inizialmente tutti gli elementi di tutte le matrici sono nulli. L'interprete inizia l'esecuzione con una bitmap di dimensione 1.

Ogni comando è dato da un *nome* (costituito da un carattere minuscolo) e, se necessario, da un elenco di uno o più numeri interi detti *parametri*. La Tabella 1 riporta nome e parametri dei comandi dell'interprete. I comandi `s` e `d` si possono considerare facoltativi.

Nome	Parametri
`n`	n
`c`	
`i`	
`r`	
`l`	
`x`	r c
`o`	r c
`h`	l r c
`v`	h r c
`b`	h l r c
`s`	m r c $a_{0,0}\dots a_{0,c-1}a_{1,0}\dots a_{1,c-1}\dots a_{r-1,0}\dots a_{r-1,c-1}$
`d`	m r c
`p`	

Tabella 1: Nomi e parametri dei comandi dell'interprete.

Un primo insieme di comandi riguarda direttamente la bitmap. Il comando `n` con parametro n crea una nuova bitmap di dimensione n in cui tutti i pixel sono inizialmente spenti (se era presente una bitmap precedente, essa viene eliminata; se n eccede il limite massimo specificato nella sezione seguente, il comando viene ignorato). Il comando `c` spegne tutti i pixel della bitmap, mentre il comando `i` inverte lo stato dei pixel rendendo accesi quelli spenti e viceversa. I comandi `r` e `l` producono una rotazione di 90 gradi della bitmap, rispettivamente verso destra e sinistra.

La manipolazione diretta dei pixel avviene tramite i comandi `x` e `o` che, rispettivamente, accendono e spengono il pixel le cui coordinate sono specificate dai parametri del comando. Se le coordinate eccedono la dimensione della bitmap, il comando sarà ignorato.

Il disegno di figure è poi ottenuto grazie ai comandi `v`, `h` e `b` che disegnano rispettivamente un segmento verticale, uno orizzontale ed un rettangolo con le dimensioni e coordinate specificate dai parametri.

La gestione dei timbri si basa su due comandi. Il comando `s` legge nell'm-esima matrice gestita dall'interprete una matrice di dimensione $r \times c$ (le matrici sono numerate da 0); i valori degli elementi della matrice sono specificati, come parametri, per riga (ossia: i primi c parametri sono i valori della prima riga, quindi i successivi c parametri quelli della seconda riga e così di seguito). Se era già stata letta la matrice m-esima, la nuova matrice letta rimpiazzerà quella precedente. Il comando `d` disegna il timbro dell'm-esima matrice alle coordinate specificate dai parametri. Anche nel caso dei timbri, se i parametri di un comando eccedono i limiti stabiliti nella sezione seguente, il comando sarà ignorato.

L'ultimo comando `p` produce la raffigurazione a caratteri della bitmap così come illustrato nell'introduzione.

Formato dell'input/output e vincoli

L'interprete legge da *standard input* una sequenza di comandi (coi relativi parametri[2]) come specificati nella Tabella 1 e, fatta esclusione per il comando `p`, non emette alcun output. Nel caso del comando `p` l'interprete deve emettere su *standard output*, per ciascuna delle n righe della bitmap, n caratteri scelti tra spazio e asterisco (a seconda dello stato dei pixel nella bitmap) e quindi il carattere di "a capo" (ossia, a fronte di una bitmap di dimensione n, un totale di esattamente $n(n+1)$ caratteri, considerando anche gli "a capo"). Il programma termina l'esecuzione qualora non ci siano più comandi in input.

Potete assumere che tutti i parametri siano numeri naturali tali da poter essere rappresentati con variabili di tipo `int`. Inoltre, potete assumere che la dimensione massima della bitmap sia 500 e che vadano gestite al più 50 matrici relative ai timbri, ciascuna di dimensione massima pari a 70×70.

[2]Comandi e parametri sono separati tra loro da uno o più caratteri *white-space*, ossia caratteri su cui la funzione `isspace` restituisce un valore non nullo.

A titolo di esempio, in Figura 1, si riportano quattro esecuzioni dell'interprete dove le parti in grassetto corrispondono a quanto il programma legge da *standard input*, mentre le altre (non in grassetto) a quanto scrive su *standard output* (e dove, come in precedenza, il carattere ␣ è usato al posto dello spazio, per aumentare la leggibilità).

- Iniziamo con l'esempio a sinistra che corrisponde al caso illustrato nell'introduzione dove, in una nuova bitmap di dimensione 4, vengono disegnati i segmenti verticali di dimensione 4 e coordinate $(0,0)$ e il segmento orizzontale di dimensione 2 e coordinate $(1,2)$.
- Proseguiamo quindi con il secondo esempio, in cui viene illustrato l'uso dei timbri; in particolare, il comando `s 0 2 3 1 0 0 0 2 0` legge nella matrice 0-esima la matrice di dimensione 2×3 data da

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix}$$

  che viene poi utilizzata per disegnare (grazie ai due comandi `d`) il timbro di coordinate $(0,0)$ e $(2,2)$.
- Il terzo esempio riguarda alcuni spegnimenti e una inversione. Dapprima viene disegnato un rettangolo e quindi ne viene spento lo spigolo in alto a sinistra, quindi vengono spenti gli altri tre spigoli (ma viene stampato solo il risultato finale delle tre cancellazioni). Infine, viene invertito lo stato dei pixel.
- L'ultimo esempio riguarda le rotazioni. Per prima cosa viene disegnata una specie di Γ che poi viene ruotata a destra. Tramite una rotazione a sinistra si torna alla figura originale, quindi viene eseguita un'altra rotazione a sinistra.

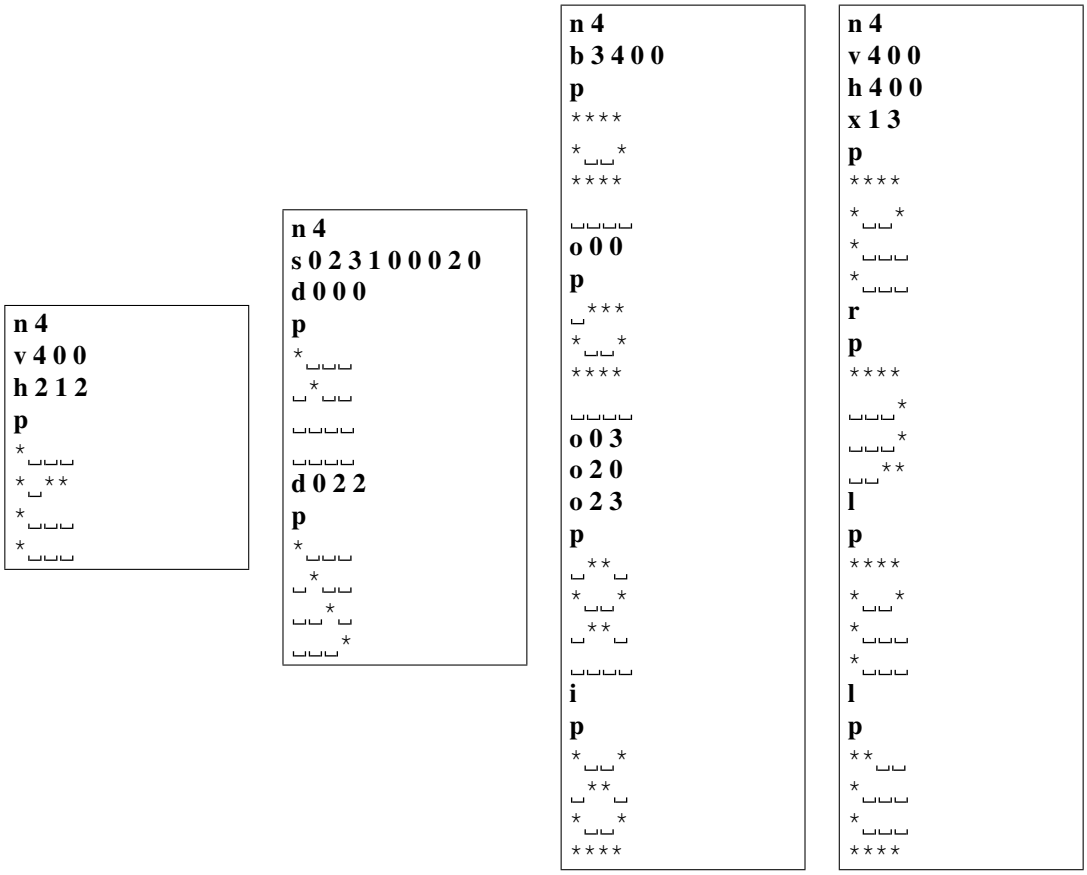

Figura 1: Quattro esempi di esecuzione.

Si ribadisce nuovamente che il formato dell'input/output deve attenersi strettamente a quanto specificato in questa sezione. **In mancanza di completa e stretta aderenza alle indicazioni precedenti il progetto non sarà ritenuta valido.**

Modalità di consegna

La consegna del progetto deve avvenire tramite l'invio di una sola mail, da spedire via mail all'indirizzo

`lonati@dsi.unimi.it`

nei mesi in cui si svolgono di norma gli appelli (gennaio, febbraio, giugno, luglio, settembre). La mail deve tassativamente provenire dall'indirizzo di posta ufficiale dello studente, ossia l'indirizzo che termina con `@studenti.unimi.it`. La mail deve contenere un unico file allegato, di tipo archivio `zip`, il cui nome deve essere composto dal cognome seguito dal numero di matricola del studente (es: per lo studente Mario Rossi matr. 422481, il file deve chiamarsi `rossi422481.zip`); l'archivio deve contenere:

- il codice sorgente (compresi eventuali header file),
- la relazione in formato `pdf` o `txt`.

I sorgenti devono essere ANSI C e compilare tutti senza errori col compilatore `gcc` installato in laboratorio; non è consentito l'uso di librerie oltre alla *standard library*. Le specifiche di formato per l'input e l'output definite sopra devono essere rispettate rigorosamente. **In presenza di errori di compilazione, la consegna non sarà ritenuta valida. Programmi consegnati diversamente da come specificato in questa sezione non saranno presi in considerazione.**

La discussione del progetto e l'eventuale prova orale si svolgeranno indicativamente, previa convocazione via email, entro 15 giorni dalla data di consegna dei progetti (e comunque nei periodi solitamente previsti per gli appelli d'esame). **E' necessario presentarsi alla prova orale con una copia stampata della relazione e del codice.**

NOTA IMPORTANTE: la realizzazione di questo progetto è una prova d'esame da svolgersi individualmente. I progetti giudicati frutto di collaborazione saranno estromessi d'ufficio dalla valutazione.