

Lezione 9

Esercizi d'esame

Rossano Venturini

rossano@di.unipi.it

Pagina web del corso

<http://didawiki.cli.di.unipi.it/doku.php/informatica/all-b/start>

Esercizio 1

Qsort su interi

Scrivere un programma che utilizzi la funzione `qsort` e ordini un vettore di interi (in input), in modo da ottenere il seguente effetto. L'array ordinato deve contenere prima tutti i numeri pari e, a seguire, i numeri dispari. Si assuma che il numero 0 sia pari. I numeri pari devono essere ordinati in modo *crescente* fra di loro. I numeri dispari devono essere ordinati in modo *decrescente* fra di loro.

Esercizio 1

```
int cmpfunc(const void *a, const void *b) {  
    int val1, val2, mod1, mod2;  
  
    val1 = *( (int*)a );  
    val2 = *( (int*)b );  
    mod1 = abs(val1 % 2);  
    mod2 = abs(val2 % 2);  
    if (mod1 != mod2) return mod1 - mod2;  
    else if (mod1 == 0) return val1 - val2;  
    else return val2 - val1;  
}
```

...

```
qsort(array, len, sizeof(int), cmpfunc);
```

...

Esercizio 2

Qsort su stringhe

Scrivere un programma che legga in input un array A di stringhe e che utilizzi la funzione `qsort` per ordinare in ordine alfabetico *decrescente* le stringhe in input. Assumere che la lunghezza massima di una stringa sia 100 caratteri.

Esercizio 2

```
int cmpfunc (const void * a, const void * b) {  
    char *x = *(char**)a, *y = *(char**)b;  
    return -strcmp(x,y);  
}
```

...

```
qsort(array, len, sizeof(char*), cmpfunc);
```

...

Esercizio 3

Qsort e struct

Scrivere un programma che utilizzi la funzione `qsort` per ordinare un vettore di punti del piano cartesiano. Ciascun punto è formato da una coppia di coordinate (x, y) .

I punti devono essere ordinati per ascissa crescente. A parità di ascissa, si ordina per ordinata decrescente.

Esercizio 3

```
typedef struct {  
    int x;  
    int y;  
} point;  
  
int cmpfunc(const void * a, const void * b) {  
    point aa = *(point*)a, bb = *(point*)b;  
    if (aa.x != bb.x) return aa.x - bb.x;  
    return bb.y - aa.y;  
}  
  
...  
qsort(points, len, sizeof(point), cmpfunc);  
...
```

Esercizio 4

Qsort su stringhe e struct

Scrivere un programma che utilizzi la funzione `qsort` per ordinare un array di stringhe. Le stringhe devono essere ordinate per lunghezza crescente. A parità di lunghezza, si utilizzi l'ordinamento lessicografico. Utilizzare una struct che memorizzi una stringa e la sua lunghezza per evitare di calcolare quest'ultima ad ogni confronto.

Esercizio 4

```
typedef struct {  
    char* str;  
    int len;  
} string;
```

```
int cmpfunc (const void * a, const void * b) {  
    string x = *(string*)a, y = *(string*)b;  
    if (x.len == y.len) return strcmp(x.str, y.str);  
    return x.len - y.len;  
}
```

...

```
qsort(strings, len, sizeof(string), cmpfunc);
```

...

Esercizio 1

Prova del 18/05/2009

Scrivere un programma che legga da tastiera due interi N e K e una sequenza di N stringhe e che stampi le K stringhe più frequenti in essa contenute, in ordine lessicografico.

Si può assumere che:

- non esistano due stringhe con la stessa frequenza;
- il valore di K sia minore o uguale al numero di stringhe distinte;
- le stringhe contengono soltanto caratteri alfanumerici ($a - z$ minuscoli e maiuscoli o numeri, nessuno spazio o punteggiatura) e sono lunghe al più 100 caratteri ciascuna.

Esercizio 2

Prova del 14/07/2009

Si consideri il quadrante positivo del piano cartesiano. Un *punto colorato* sul piano è caratterizzato da una tripla (x, y, c) , dove x , y e c sono valori interi non-negativi. Il primo intero della tripla caratterizza l'ascissa del punto, il secondo intero l'ordinata, il terzo intero è il colore assegnato al punto.

Sia A un insieme di N punti colorati. Lo scopo del programma è quello di rispondere a una sequenza di interrogazioni sui punti di A . Un' *interrogazione* è definita da due coppie $\langle (x_1, y_1); (x_2, y_2) \rangle$, dove $x_1 < x_2$ e $y_1 < y_2$, che identificano un rettangolo R :

$$R = \{(u, v) \in \mathbb{N}^2 \mid x_1 \leq u \leq x_2 \wedge y_1 \leq v \leq y_2\}$$

Data un'interrogazione R , si vuole calcolare il numero di colori **distinti** dei punti di A che ricadono in R (i punti sul perimetro del rettangolo devono essere considerati nel conteggio).

Scrivere un programma che legga da tastiera una sequenza A di N punti colorati e un insieme Q di M interrogazioni, e stampi la risposta a ciascuna interrogazione su una riga distinta. Nel caso non vi siano punti all'interno del rettangolo stampare 0.

Esercizio 3

Prova del 11/09/2009

L'*anagramma principale* di una stringa S è l'anagramma di S ottenuto ordinando i suoi simboli individualmente. Ad esempio, l'anagramma principale di `abracadabra` è `aaaaabbcdrr`. Notare che l'anagramma principale di una stringa è unico e che stringhe diverse possono avere lo stesso anagramma principale.

Scrivere un programma che legga da tastiera una sequenza A di N stringhe di lunghezza variabile. Il programma deve poi raggruppare le stringhe di A aventi lo stesso anagramma principale e restituire le stringhe di ciascun gruppo in ordine lessicografico non-decrescente. I gruppi devono essere restituiti in ordine lessicografico non-decrescente del corrispondente anagramma principale.

Puzzle

Spia

Una spia si trova in territorio nemico e deve spedire un messaggio di 4 bit al suo capo senza far notare la sua presenza. Fortunatamente la radio ufficiale dei nemici trasmette in broadcast un messaggio di 15 bit. La spia può modificare un bit di questo messaggio e vuole sfruttare questa sua abilità per trasmettere il suo messaggio. Il capo dovrà essere in grado di ottenere il messaggio quando riceverà i 15 bit in broadcast. Come?

Nota: Il capo e la spia si sono precedentemente accordati su una strategia. Il messaggio di 15 bit non era noto al momento dell'accordo.