

Esercizi per il nono laboratorio

Antonio Miele Ettore Speciale Michele Tartara

9 dicembre 2010

10 dicembre 2010

14 dicembre 2010

Gli esercizi proposti in questo laboratorio richiedono di utilizzare i sottoprogrammi e l'allocazione dinamica della memoria. Ognuno di essi deve essere implementato all'interno di un sottoprogramma che viene invocato dal `main`.

Duplicatore di stringhe - `strcpy`

Scrivere un sottoprogramma che ricevuta in ingresso una stringa ne crea una nuova contenente una copia della stringa ricevuta in ingresso e la restituisce al chiamante. La dimensione della nuova stringa è quella strettamente necessaria in base al contenuto della stringa in ingresso.

Convertitore decimale-binario

Scrivere un programma che chiesto all'utente un numero intero positivo, lo rappresenta in base 2 e lo visualizza. Allo scopo sviluppare un sottoprogramma che ricevuto in ingresso il numero acquisito dall'utente, **restituisce** un array di interi contenente la rappresentazione binaria (posizione 0 bit più significativo). L'array deve essere dimensionato opportunamente. La dimensione di questo nuovo array viene **trasmessa** al chiamante.

Si noti che è necessario calcolare la dimensione dell'array per poterlo allocare dinamicamente. Perciò si suggerisce di utilizzare i seguenti sottoprogrammi della libreria matematica:

- `double log2 (double x);` che calcola il logaritmo in base 2 di un numero
- `double ceil (double x);` che calcola l'arrotondamento per eccesso di un numero

Per utilizzare la libreria matematica bisogna includere all'inizio del codice sorgente il file header `math.h`; inoltre, quando si compila il programma bisogna eseguire il link con la libreria `libm`:

```
gcc -o convertitore -lm convertitore.c
```

Assemblatore di stringhe

Scrivere un sottoprogramma che ricevette in ingresso due stringhe le concatena, e restituisce la stringa risultante al chiamante. La stringa risultante deve essere creata dal sottoprogramma e deve essere della dimensione strettamente necessaria in base al contenuto.

Esempio

Supposto che il sottoprogramma sia invocato con le stringhe:

$$s_1 = \text{"casa"}, s_2 = \text{"matta"}$$

la stringa risultante e restituita sarà:

$$s_3 = \text{"casamatta"} \quad (1)$$

Formattatore di stringhe

Scrivere un sottoprogramma che una volta richiesto un carattere jolly j e due stringhe provvede a creare una nuova stringa in cui ricopia la prima stringa e ad ogni occorrenza del carattere j nella nuova stringa sostituisce l'intera seconda stringa. La nuova stringa va poi restituita al programma chiamante.

Per calcolare la dimensione della stringa risultante (formattata) si consiglia di implementare un sottoprogramma, che preso in ingresso una stringa ed un carattere restituisce il numero di occorrenze del carattere nella stringa. Usando tale informazione, assieme alla dimensione della stringa da sostituire, è possibile calcolare esattamente la dimensione della stringa formattata.

Esempio

Supposto che l'utente abbia inserito i seguenti dati:

```
jolly = '*'
template = "Hello *, I'm *"
subs = "Michele"
```

Esso costruisce la stringa:

```
formatted = "Hello Michele, I'm Michele"
```

Costruttore di matrici

Scrivere un sottoprogramma che riempie una matrice bidimensionale con dei dati inseriti dall'utente. La matrice non ha una dimensione fissa, e quindi deve essere allocata dinamicamente, previa richiesta delle dimensioni all'utente.

Successivamente il sottoprogramma mostra un menù che permette di:

- cancellare una colonna della matrice;
- cancellare una riga della matrice;

- interrompere il sottoprogramma.

Dopo aver effettuato un'operazione di modifica, il sottoprogramma ripresenta in menù, permettendo di selezionare una nuova operazione.

Consigliamo di implementare ogni operazione di modifica della matrice attraverso un sottoprogramma che preso l'indirizzo della matrice, applica la trasformazione e ritorna un puntatore alla matrice modificata. Per operare con la matrice all'interno dei vari sottoprogrammi si sfrutti la linearizzazione delle matrici.