Fondamenti di Informatica: esercitazione di laboratorio n. 4
Stringhe

Pier Luca Montessoro

1. Concatenazione di stringhe

Si scriva la funzione

```
void strcatenate (char t[], char s[])
```

che inserisce la stringa s alla fine della stringa t.

Si scriva il relativo main per provarla. Il formato dei dati di ingresso e di uscita può essere dedotto dall'esempio riportato più oltre.

Nota: la stringa in ingresso va letta con la funzione `fgets` e la stringa in uscita va scritta con la funzione `puts`. La funzione `fgets` supera il problema di vulnerabilità della `gets` includendo la dimensione massima del vettore che conterrà la stringa letta. Per esempio, se il vettore ha dimensione 32 byte, la `fgets` andrà scritta così:

```
fgets (s, 32, stdin);
```

Essa leggerà dalla tastiera (`stdin`) al massimo 31 caratteri (il trentaduesimo servirà per il `'\0'`).

```
prima stringa: testo
seconda stringa: prova
testoprova
```

2. Inserimento di una stringa all'inizio di un'altra

Si scriva la funzione `void strinsert (char s[], char t[])` che aggiunge il contenuto della stringa t all'inizio della stringa s.

Esempio: se s = "pianta" e t = "si " si otterrà s = "si pianta".

NOTA1: non esiste una funzione di libreria con questa funzionalità.
NOTA2: attenzione all'algoritmo, è facile sbagliare ed è possibile che l'errore si verifichi soltanto con alcune lunghezze delle stringhe.

Si scriva il relativo main per provarla. Il formato dei dati di ingresso e di uscita può essere dedotto dall'esempio riportato più oltre. Per le funzioni da usare per leggere e scrivere le stringhe di test si faccia riferimento all'esercizio 1.

Esempio:

```
prima stringa: testo
seconda stringa: prova
provatesto
```

3. Confronto di stringhe

Si scriva la funzione

```
int strcompare (char a[], char b[])
```

che confronta la stringa `a` e la stringa `b` e restituisce:
`-1` se la stringa `a` precede, in ordine alfabetico, la stringa `b`
`0` se la stringa `a` è uguale alla stringa `b`
`1` se la stringa `a` segue, in ordine alfabetico, la stringa `b`

Si scriva il relativo main per provarla. Il formato dei dati di ingresso e di uscita può essere dedotto dagli esempi riportati più oltre. Per le funzioni da usare per leggere e scrivere le stringhe di test si faccia riferimento all'esercizio 1.

Esempi:

```
prima stringa: a
seconda stringa: a
strcompare ha restituito 0

prima stringa: a
seconda stringa: b
strcompare ha restituito -1
```

```
prima stringa: b
seconda stringa: a
strcompare ha restituito 1

prima stringa: aa
seconda stringa: a
strcompare ha restituito 1

prima stringa: a
seconda stringa: aa
strcompare ha restituito -1

prima stringa: aa
seconda stringa: ab
strcompare ha restituito -1

prima stringa: aab
seconda stringa: aaa
strcompare ha restituito 1

prima stringa: aaabb
seconda stringa: aaaba
strcompare ha restituito 1
```

4. Inversione delle parole in una stringa

Scrivere una funzione in linguaggio C che accetti in ingresso una stringa contenente una frase (parole separate soltanto da spazi) e inverta l'ordine delle lettere di ogni singola parola.

Si scriva il relativo main per provarla. Il formato dei dati di ingresso e di uscita può essere dedotto dall'esempio riportato più oltre. Per le funzioni da usare per leggere e scrivere le stringhe di test si faccia riferimento all'esercizio 1.

Esempio:

```
testo di prova
otset id avorp
```

5. Complemento a due con stringhe

Si scriva una funzione che riceva come argomento una stringa di caratteri che rappresenta un numero binario e ne restituisca in una seconda stringa il complemento a 2. Attenzione: il numero può anche essere zero!

Si scriva il relativo main per provarla. Il formato dei dati di ingresso e di uscita può essere dedotto dagli esempi riportati più oltre. Per le funzioni da usare per leggere e scrivere le stringhe di test si faccia riferimento all'esercizio 1.

Esempi:

```
inserire il numero in binario: 1111
0001

inserire il numero in binario: 00000001
11111111

inserire il numero in binario: 00000101
11111011

inserire il numero in binario: 00000000
00000000
```

6. Frasi palindrome

Una frase è palindroma se appare identica se letta sia da destra che da sinistra, ignorando gli eventuali spazi, accenti, apostrofi e simboli di interpunzione. Un paio di esempi: "Angolo bar a Bologna", "avida di vita, desiai ogni amore vero, ma ingoiai sedativi, da diva". Si osservi che non sono significative le differenze tra lettere maiuscole e minuscole.

Si scriva la funzione `int frase_palindroma (char s[])` che restituisce il valore logico *vero* o *falso* (secondo lo standard del linguaggio C) in base a quanto sopra descritto.

Si scriva il relativo main per provarla. Il formato dei dati di ingresso e di uscita può essere dedotto dagli esempi riportati più oltre. Per le funzioni da usare per leggere e scrivere le stringhe di test si faccia riferimento all'esercizio 1.

Esempi:

```
oro
palindroma

radar
palindroma

prova
non palindroma

frase di prova
non palindroma

o mordo tua nuora o aro un autodromo
palindroma
```

7. Nomi abbreviati

Si consideri la rappresentazione di nomi e cognomi mediante una stringa nel seguente formato:

```
"cognome, nome"
```

Sia il nome che il cognome possono essere composti da più parole.

Si scriva la funzione `void nome_abbreviato (char *s, char *t)` in linguaggio C che riceve in ingresso una stringa `t` in tale formato e scriva in `s` il seguente formato abbreviato:

```
"n. cognome"
```

dove `n.` è l'abbreviazione del nome. Si assuma che la stringa `t` sia sicuramente in formato corretto.

ALGORITMO:

- cerca l'inizio del nome
- salta la virgola
- cerca, con un ciclo, le parole del nome e, per ciascuna, scrive l'iniziale nella stringa s (si userà una variabile `in_parola`, come nel programma wc.c)
- infine copia il cognome
- e poi chiude la stringa s con il `'\0'`

Si scriva il relativo main per provare la funzione. Il formato dei dati di ingresso e di uscita può essere dedotto dagli esempi riportati più oltre. Per le funzioni da usare per leggere e scrivere le stringhe di test si faccia riferimento all'esercizio 1.

Esempi:

```
Della Valle Di Sotto, Pier Luigi
P. L. Della Valle Di Sotto

Della Scala, Ferdinando Giuseppe Filippo Ugo
F. G. F. U. Della Scala

Rossi, Mario
M. Rossi
```

8. Conversione di formato per la data

Si considerino i seguenti formati per la data:
formato 1: `aaaa-mm-gg` (esempi: `2006-2-14`, `2000-1-1`)
formato 2: `gg mese` *`dell'anno`* `anno` (esempi: `14 febbraio dell'anno 2006`, `1 gennaio dell'anno 2000`)

Si scriva la funzione `void convertiData (char data1[], char data2[])` che riceva nella stringa `data1` una data nel formato 1 e la trasformi nel formato 2 scrivendolo nella stringa `data2`.

Evidentemente l'esercizio si compone di due sottoproblemi da associare a opportune funzioni:
- estrarre la data dalla stringa nel formato 1
- costruire una stringa nel formato 2

Attenzione alla rappresentazione della data nel passaggio tra il primo e il secondo sottoproblema. A prescindere dall'uso che ne dovete fare, una data si rappresenta con *tre numeri interi*.

Suggerimento: per scrivere il risultato nella stringa di uscita si suggerisce di utilizzare la funzione `sprintf` che è identica alla `printf` con in più una stringa come primo argomento. La stampa, infatti, non avviene sul monitor ma su tale stringa.
Esempio: `sprintf (s, "%d + %d = %d", 1, 2, 3);` scrive nella stringa s: `"1 + 2 = 3"`.

Per la prima funzione, dopo aver copiato ciascuna parte della stringa di ingresso in una opportuna stringa temporanea, può anche essere utile l'utilizzo della funzione di libreria `atoi` che riceve come argomento una stringa rappresentante un numero intero decimale e ne restituisce il valore.
È anche possibile sfruttare la funzione `sscanf` con una opportuna stringa di formato, ma l'esercizio così diventerebbe troppo semplice e quindi poco interessante… 😊

Si scriva il relativo main per provarla. Il formato dei dati di ingresso e di uscita può essere dedotto dagli esempi riportati più oltre Per le funzioni da usare per leggere e scrivere le stringhe di test si faccia riferimento all'esercizio 1.

Esempi:

```
#formati_data
2006-2-14
14 febbraio dell'anno 2006

#formati_data
2000-1-1
1 gennaio dell'anno 2000
```