

# Esercizi su stringhe e struct in C

Stefano Cherubin\*

03/11/2016

[**Informatica A**] Esercitazione #7

corso per Ing. Gestionale a.a. 2016/17

---

\*<nome>.<cognome>@polimi.it

## Indice

<b>1</b>	<b>Parole al contrario</b>	<b>3</b>
1.1	Soluzione C . . . . .	4
<b>2</b>	<b>Parole palindrome</b>	<b>5</b>
2.1	Soluzione C . . . . .	5
<b>3</b>	<b>Alla ricerca dello Jedi</b>	<b>6</b>
3.1	Soluzione C . . . . .	7
<b>4</b>	<b>Prodotti scaduti</b>	<b>8</b>
4.1	Soluzione C . . . . .	9

## 1 Parole al contrario

Presa in input una stringa che rappresenta una frase di più parole, stampare in output la stessa frase con ogni parola scritta al contrario. Le parole sono separate da uno spazio.

Esempio:

**Input** non ho studiato informatica a sufficienza

**Output** non oh otaiduts acitamrofni a azneiciffus

**Nota** per poter acquisire da tastiera una stringa `str` contenente 'spazi' e terminata dal carattere 'invio' l'istruzione da utilizzare é la seguente: `scanf ("%^\n]s", str);`

## 1.1 Soluzione C

Listato 1: Parole al contrario

```
1 #include <stdio.h>
2 #define LEN 50
3 int main() {
4     int inizio_parola, i, j;
5     char str[LEN + 1];
6     printf("Inserisci una sequenza di parole: ");
7     scanf("%[^\n]s", str);
8     inizio_parola = 0;
9     i = 0;
10    /* fino alla fine della stringa */
11    while (str[i] != '\0') {
12        /* lo spazio è la fine della parola */
13        if (str[i] == ' ') {
14            /* stampa la parola al contrario */
15            for (j = i - 1; j >= inizio_parola; j--) {
16                printf("%c", str[j]);
17            }
18            inizio_parola = i + 1;
19            printf(" ");
20        }
21        i++;
22    }
23    /* ultima parola (che termina con '\0' ) */
24    for (j = i - 1; j >= inizio_parola; j--) {
25        printf("%c", str[j]);
26    }
27    printf("\n");
28    return 0;
29 }
```

## 2 Parole palindrome

Scrivere un programma in linguaggio C che letta dallo standard input una parola, determini se essa è palindroma.

### 2.1 Soluzione C

Una parola è palindroma se letta al contrario è identica a sé stessa.

Listato 2: Palindrome

```
1 #include <stdio.h>
2 #include <string.h> /* libreria per strlen */
3 #define MAX_LEN 50
4 int main( ) {
5     char parola[MAX_LEN + 1];
6     int len, i, flag;
7     /* acquisizione */
8     printf("\nInserire una parola per verificare se e'
9         palindroma ");
10    scanf("%s", parola);
11    len = strlen(parola);
12    flag = 1;
13    /* Controlla i caratteri partendo dalle estremità
14       verso la metà. Interrompe al primo carattere non
15       corrispondente. */
16    for (i = 0; i < len / 2 && flag == 1; ++i) {
17        /* parola[len] = '\0'
18           parola[len-1] = ultimo carattere */
19        if (parola[i] != parola[len - 1 - i]) {
20            flag = 0;
21        }
22    }
23    if (flag == 1) {
24        printf("\nLa parola e' palindroma.");
25    } else {
26        printf("\nLa parola NON e' palindroma.");
27    }
28    return 0;
29 }
```

### 3 Alla ricerca dello Jedi

Siete alla ricerca di nuovi Jedi per le diverse colonie della Repubblica. I Jedi hanno caratteristiche straordinarie, nelle loro cellule si ha un'elevata concentrazione di midi-chlorian. Per semplicità si supponga che questa elevata concentrazione di midi-chlorian (indicatore del potenziale di uno Jedi) derivi da un particolare gene (*key*) nel DNA dello Jedi.

Dato il DNA di un individuo, espresso come sequenza di caratteri, stabilire se esso contiene o meno la stringa del gene Jedi.

### 3.1 Soluzione C

Listato 3: Alla ricerca dello Jedi

```
1 #include <stdio.h>
2 #include <string.h>
3
4 #define DATA_SIZE 100
5 #define KEY_SIZE 25
6
7 int main ( )
8 {
9     char data[DATA_SIZE + 1];
10    char key[KEY_SIZE + 1];
11    int data_len, key_len;
12    int i, j, found;
13
14    printf("please insert data ");
15    scanf("%s", data);
16    printf("please insert key ");
17    scanf("%s", key);
18    data_len = strlen(data);
19    key_len = strlen(key);
20
21    found = 0;
22    for (i = 0; i < (data_len - key_len) && !found; ++i)
23    {
24        found = 1;
25        for (j = 0; j < key_len && found; ++j) {
26            if (data[i + j] != key[j]) {
27                found = 0;
28            }
29        }
30
31        if (found)
32            printf("Key was found!\n");
33        else
34            printf("key not found...\n");
35
36        return 0;
37    }
```

## 4 Prodotti scaduti

Caricare un vettore di 5 elementi **Prodotto** rappresentanti degli ingredienti presenti nel frigorifero. Al termine del caricamento si deve acquisire in input una data e verificare per ogni prodotto se a quella data il prodotto sarà o meno scaduto.

Per rappresentare un **Prodotto** si utilizzi un tipo strutturato che abbia come elementi il nome del prodotto e una data di scadenza.

Si utilizzi per rappresentare una data un tipo così definito

```
typedef struct {  
    int giorno;  
    int mese;  
    int anno;  
} Data;
```



## 4.1 Soluzione C

Listato 4: Prodotti scaduti

```
1 #include <stdio.h>
2 #define MAX_LEN 100 /* lunghezza nome */
3 #define N_PROD 5 /* numero prodotti */
4
5 typedef struct {
6     int giorno;
7     int mese;
8     int anno;
9 } Data;
10
11 typedef struct {
12     char nome[MAX_LEN];
13     Data scadenza;
14 } Prodotto;
15
16 int main() {
17     Prodotto frigo[N_PROD];
18     Data controllo;
19     int i, flag;
20     /* input frigo */
21     for(i = 0; i < N_PROD; i++) {
22         printf("\nInserisci il nome del prodotto: ");
23         scanf("%s", frigo[i].nome);
24         do { /* input blindato sulla data */
25             printf("\nInserisci la data di scadenza di %s ",
26                 frigo[i].nome);
27             scanf("%d %d %d", &frigo[i].scadenza.giorno, &
28                 frigo[i].scadenza.mese, &frigo[i].scadenza.
29                 anno);
30             } while (frigo[i].scadenza.giorno <= 0 || frigo[i]
31                 .scadenza.giorno > 31 || frigo[i].scadenza.
32                 mese <= 0 || frigo[i].scadenza.mese > 12 ||
33                 frigo[i].scadenza.anno < 2015);
34     }
35     /* input data di controllo */
36     do {
37         printf("\nInserisci la data a cui controllare la
38             scadenza ");
39         scanf("%d %d %d", &controllo.giorno, &controllo.
40             mese, &controllo.anno);
41     } while (controllo.giorno <= 0 || controllo.giorno >
42         31 || controllo.mese <= 0 || controllo.mese >
```

```

12);
34  /* controllo */
35  for (i = 0; i < N_PROD; ++i) {
36      /* flag -1 -> scaduto
37         flag 1 -> buono
38         flag 0 -> scade oggi */
39      if (frigo[i].scadenza.anno > controllo.anno)
40          flag = 1;
41      else if (frigo[i].scadenza.anno < controllo.anno)
42          flag = -1;
43      else if (frigo[i].scadenza.mese > controllo.mese)
44          flag = 1;
45      else if (frigo[i].scadenza.mese < controllo.mese)
46          flag = -1;
47      else if (frigo[i].scadenza.giorno > controllo.
48                giorno)
49          flag = 1;
50      else if (frigo[i].scadenza.giorno < controllo.
51                giorno)
52          flag = -1;
53      else
54          flag = 0;
55      /* output */
56      switch (flag) {
57          case 1:
58              printf("\nIl prodotto %s sara' ancora buono!",
59                    frigo[i].nome);
60              break;
61          case 0:
62              printf("\nIl prodotto %s scade esattamente in
63                    quella data.", frigo[i].nome);
64              break;
65          case -1:
66              printf("\nIl prodotto %s sara' ormai da
67                    buttare!", frigo[i].nome);
68              break;
69      }
70  }
71  return 0;
72 }

```

## **Licenza e crediti**

### **Licenza beerware<sup>1</sup>**

Quest'opera è stata redatta da Stefano Cherubin. Mantenendo questa nota, puoi fare quello che vuoi con quest'opera. Se ci dovessimo incontrare e tu ritenessi che quest'opera lo valga, in cambio puoi offrirmi una birra.

---

<sup>1</sup><http://people.freebsd.org/~phk/>