

Esercizi su stringhe e struct in C

Stefano Cherubin*



26/10/2017

[Informatica A] Esercitazione #7

corso per Ing. Gestionale a.a. 2017/18

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

Indice

1	Parole al contrario	3
1.1	Soluzione C	4
2	Una storia scalda-ossa	5
2.1	Soluzione C	6
3	Alla ricerca dello Jedi	7
3.1	Soluzione C	8
4	Prodotti scaduti	9
4.1	Soluzione C	10

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 Una storia calda-ossa

Lasciate che vi racconti una storia per scaldarvi le ossa
È una cosa che mi è successa davvero
Una notte passeggiavo per le Everglades
Avevo bevuto un drink ma non di più
Vagavo godendomi la luce della luna piena
Camminavo col naso all'insù per guardare le stelle
Non avevo notato una presenza così vicina
Che osservava ogni mio movimento
Preso dallo spavento sono caduto a terra
Quando qualcosa mi precipitò addosso dagli alberi
Mi portò in un luogo profano
Ed è qui che caddi in disgrazia
Mi convocarono perché mi unissi a loro
Nella danza dei defunti
Nel cerchio di fuoco li seguii
Nel centro fui condotto
Come se il tempo si fosse fermato, ero intorpidito dalla paura
Ma non volevo tirarmi indietro
E le vampate del fuoco non mi ferivano
Mentre camminavo sulle braci
Le loro voci erano un coro confuso senza inizio, senza fine
Ogni parola finiva come iniziava.
In questa altalena di suoni palindromi la mia voce si perdeva
Il rito continuò finché non persi anche la cognizione di me
E i dannati mi sostituirono nel mio corpo
Ed io fui dannato con loro

Nell'eventualità di trovarvi in una situazione simile, scrivere un programma in linguaggio C che, data una parola in input, determini se essa è palindroma.

Quando vi troverete nelle Everglades e riconoscete troppe parole palindrome, iniziate a correre!

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 MAX_DATA_SIZE 100
5 #define MAX_KEY_SIZE 25
6
7 int main ( )
8 {
9     char dna_data[MAX_DATA_SIZE + 1];
10    char key[MAX_KEY_SIZE + 1];
11    int dna_len, key_len;
12    int i, j, jedi, equal;
13
14    printf("please insert data ");
15    scanf("%s", dna_data);
16    printf("please insert key ");
17    scanf("%s", key);
18    dna_len = strlen(dna_data);
19    key_len = strlen(key);
20
21    jedi = 0;
22    for (i = 0; i < (dna_len - key_len) && !jedi; ++i) {
23        /* un'altra flag per controllare l'uguaglianza della
24           sottostringa carattere per carattere */
25        equal = 1;
26        for (j = 0; j < key_len && equal; ++j) {
27            if (dna_data[i + j] != key[j]) {
28                equal = 0;
29            }
30        }
31        /* jedi trovato se equal rimane 1 per tutta la
32           lunghezza di key - era anche possibile utilizzare
33           un'unica flag per entrambi i controlli */
34        jedi = equal;
35    }
36
37    if (jedi)
38        printf("Jedi was found!\n");
39    else
40        printf("Not the Jedi you were looking for...\n");
41    return 0;
42 }
```


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 beerware¹

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.

¹<http://people.freebsd.org/~phk/>