

Fondamenti di Informatica e Laboratorio T-AB e Fondamenti di Informatica T1
Ingegneria Elettronica e Telecomunicazioni e
Ingegneria dell'Automazione

Lab 08

Stringhe e strutture

Esercizio 1

- Scrivere una funzione che data una stringa A calcoli la sua lunghezza:

```
int lunghezza(char A[]);
```

- Scrivere una funzione che date due stringhe A e B copi il contenuto di B in A e restituisca il numero di caratteri copiati:

```
int copiastr(char dst[], char src[]);
```

- Scrivere una funzione che date tre stringhe A, B e C concateni in A il contenuto di B e C e restituisca il numero di caratteri copiati:

```
int conc(char dest[], char s1[], char s2[]);
```

NON UTILIZZARE FUNZIONI DI LIBRERIA

Esercizio 1 - Soluzione

```
int lunghezza(char A[]) {
    int result = 0;
    while (A[result] != '\0')
        result++;
    return result;
}
```

```
int copiastr(char dest[], char src[]) {
    int result = 0;
    while (src[result] != '\0') {
        dest[result] = src[result];
        result++;
    }
    dest[result] = '\0';
    return result;
}
```

```
int conc (char dest[], char s1[], char s2[]){
    int l1, l2;
    l1 = copiastr(dest, s1);
    l2 = copiastr(&dest[result], s2);
    return (l1 + l2);
}
```

Esercizio 1 - Soluzione

```
int main()
{
    char s1[] = "Ciao";
    char s2[] = "Mondo";
    char result[15];

    printf("Lunghezza di s1 = %d\n", lunghezza(s1));
    copiastr(result, s1);
    printf("Copia di s1 = %s\n", result);
    conc(result, s1, s2);
    printf("Concatenazione di s1 e s2 = %s\n", result);

    return 0;
}
```

Esercizio 2

- Al fine di stampare degli indirizzi su delle buste, è necessario comporre la prima parte dell'indirizzo come “Cognome Nome” o “Cognome N.”
- Si realizzi una funzione che riceva come parametri:
 - il cognome
 - il nome
 - una stringa che conterrà la prima parte dell'indirizzo
 - la lunghezza massima della stringa indirizzo

Esercizio 2

- La funzione deve copiare/concatenare nell'indirizzo il cognome seguito dal nome, avendo cura di rispettare le dimensioni della stringa indirizzo. Qualora la stringa indirizzo sia troppo piccola per contenere entrambi, la funzione provi a comporre la stringa come "Cognome N."
- Qualora neanche ciò sia possibile, la funzione ritorni un codice di errore opportuno (esempio - 1)
- Se non si verifica nessun errore la funzione deve restituire il numero di caratteri nella stringa Indirizzo

Esercizio 2

- Si realizzi una funzione che riceva come parametri:
 - il cognome
 - il nome
 - una stringa che conterrà la prima parte dell'indirizzo
 - la lunghezza massima della stringa indirizzo

```
int indirizzo(char cognome[], char nome[],  
char indirizzo1[],int dim);
```

Esempio: Se il cognome è Rossi e il nome è Mario e la dimensione dim = 15 allora la stringa indirizzo1 sarà "Rossi Mario". Se invece la dimensione fosse 8 allora la stringa indirizzo1 sarebbe "Rossi M.". Se la dimensione fosse 5 allora verrebbe restituito un codice di errore -1 senza modificare indirizzo1

Esercizio 2 - Soluzione

```
#include <stdio.h>
#include <string.h>

int componiIndirizzo(char * cognome, char * nome, char * indirizzo1, int dim) {
    int requiredChars;
    int compressNome = 0, size;

    requiredChars = strlen(cognome) + 1 + strlen(nome);
    if (requiredChars > dim) {
        requiredChars = strlen(cognome) + 3;
        if (requiredChars > dim)
            return -1;
        else
            compressNome = 1;
    }
    strcpy(indirizzo1, ""); // inizializzazione
    strcat(indirizzo1, cognome);
    strcat(indirizzo1, " ");
    if (!compressNome)
        strcat(indirizzo1, nome);
    else {
        size = strlen(indirizzo1);
        indirizzo1[size] = nome[0];
        indirizzo1[size+1] = '.';
        indirizzo1[size+2] = '\0';
    }
    return strlen(indirizzo1);
}
```

Esercizio 2 - Soluzione

```
#define MAX 20

int main()
{
    char indirizzo[MAX];
    int result;

    result = componiIndirizzo("Rossi", "Federico", indirizzo, MAX-1);
    if (result > 0)
    {
        printf("%s\n", indirizzo);
        printf("Lunghezza indirizzo: %d\n", result);
    }
    else
        printf("Impossibile generare un indirizzo\n");

    return 0;
}
```

Esercizio 3

Si realizzi un programma C che legga da utente i dati relativi ad alcuni corsi. In particolare, per ogni corso vengono dati:

- **denominazione** del corso: una stringa di 20 caratteri che riporta il nome del corso;
- **cognome** del docente: una stringa di 15 caratteri che rappresenta il cognome del docente del corso;
- **iscritti**: un intero che indica il numero di studenti che frequentano il corso.

Il programma deve stampare la denominazione del corso e il cognome del docente relativi a tutti i corsi che hanno il numero di iscritti maggiore o uguale alla media aritmetica degli iscritti (calcolata su tutti i corsi).

Esercizio 3

Attenzione: abbiamo bisogno di un ARRAY di strutture !!!!!

Esempio: l'utente inserisce i seguenti dati per 3 corsi

analisi

obrecht

55

fond.inf

milano

40

geometria

ferri

37

<i>analisi</i>	<i>fond.inf</i>	<i>geometria</i>
<i>obrecht</i>	<i>milano</i>	<i>ferri</i>
<i>55</i>	<i>40</i>	<i>37</i>

La media e' di 44 quindi il programma stampa:

analisi

obrecht

Esercizio 3

Si suddivida il programma appena creato in 3 diverse funzioni:

- Funzione 1: lettura dei corsi da input
- Funzione 2: stampa del singolo corso
- Funzione 3: calcolo della media e stampa dei corsi con numeri di iscritti superiore o uguale ad essa

Si crei la libreria corsi (corsi.h/corsi.c) contenente le 3 funzioni di cui sopra all'interno della libreria

Esercizio 3 – Soluzione (corsi.h)

```
#ifndef _CORSI_H_
#define _CORSI_H_

#define N 30

typedef struct {
    char denominazione[20];
    char cognome_docente[15];
    int studenti;
} Corso;

int leggi_corsi(Corso corsi[]);
void stampa_corso(Corso c);
void mostra_corsi_multi_iscritti(Corso corsi[]);

#endif
```

Esercizio 3 – Soluzione (corsi.c)

```
#include <stdio.h>
#include "corsi.h"

int leggi_corsi(Corso corsi[], int dim_fis) {
    int i, nc;
    printf("Inserisci il numero dei corsi: ");
    scanf("%d", &nc);
    if(nc > dim_fis) nc = dim_fis;

    /* inserimento dati */
    for (i=0; i<nc; i++) {
        printf("Inserisci il nome del corso ");
        scanf("%s",corsi[i].denominazione);
        printf("Inserisci il cognome del docente ");
        scanf("%s",corsi[i].cognome_docente);
        printf("Inserisci il numero degli iscritti");
        scanf("%d",&corsi[i].studenti);
    }
    return nc;
}
```

Continua...

Esercizio 3 – Soluzione (corsi.c)

```
void stampa_corso(Corso c) {  
    printf("Corso: %s; docente: %s; n. iscritti: %d",  
          c.denominazione, c.cognome_docente, c.studenti);  
}
```

```
void mostra_corsi_multi_iscritti(Corso corsi[], int nc) {  
    float somma, media;  
    somma=0;  
    for (i=0; i< nc; i++)  
        somma = somma + corsi[i].studenti;  
    media= somma/nc;  
  
    for (i=0; i< nc; i++)  
        if (corsi[i].studenti>=media)  
            printf("%s %s\n",corsi[i].denominazione,  
                  corsi[i].cognome_docente);  
}
```

Esercizio 3 – Soluzione (main.c)

```
#include <stdio.h>
#include "corsi.h"

int main() {
    int i, nc;
    corso corsi[N];
    /* inserimento dati */
    nc = leggi_corsi(corsi, N);
    /* controlla correttezza inserimento */
    for (i=0; i<nc; i++)
        stampa_corso(corsi[i]);
    /* calcola media e stampa corsi numerosi */
    mostra_corsi_molti_iscritti(corsi, nc);
}
```

Esercizio 4

- Scrivere una funzione C che, data una stringa A ed una stringa B, calcoli il numero di occorrenze della stringa A in B.
- `int occorrenze(char A[], char B[]);`
- Ad esempio, se B="tre tigri contro tre tigri" ed A="tr", deve restituire 3.

Esercizio 4 - Soluzione

```
int occorrenze(char a[], char b[]) {
    int result = 0;
    int trovato = 0;
    int i=0, j=0, temp;

    while (b[j] != '\0') {
        if (b[j] == a[i]) {
            trovato = 1;
            temp = j;
            while (a[i] != '\0' && trovato) {
                if (a[i] == b[j]) {
                    i++;
                    j++;
                }
                else {
                    trovato = 0;
                }
            }
            if (trovato)
                result++;
            j = temp;
            i=0;
        }
        j++;
    }
    return result; }
```

Esercizio 4 - Soluzione

```
/* Variante con la strcmp/strlen/strcpy
 * e copia di sottoarray.
 */

int occorrenze(char a[], char b[])
{
    int i, result = 0;
    int alen = strlen (a);
    int blen = strlen (b);
    int iters = blen - alen;

    for (i = 0; i < iters; i++)
    {
        int k;
        char c[MAXDIM];      /* creo una stringa temporanea c */

        strcpy (c, &b[i]); /* ci copio la sottostringa che inizia a b[i] */
        c[alen] = '\\0';    /* e la termino dopo alen caratteri */

        if (!strcmp (c, a)) /* ora c e a hanno la stessa lunghezza */
            result++;
    }

    return result;
}
```

Esercizio 4 - Soluzione

```
/* Seconda variante con la strcmp/strlen/strcpy
 * e passaggio per riferimento di sottoarray.
 */

int occorrenze(char a[], char b[])
{
    int i, result = 0;
    int alen = strlen (a);
    int blen = strlen (b);
    int iters = blen - alen;

    for (i = 0; i < iters; i++)
    {
        char *tmp = &b[i]; /* puntatore alla sottostringa che inizia a b[i] */
        char s = tmp[alen]; /* memorizzo il carattere in posizione b[i+alen] */
        tmp[alen] = '\\0'; /* e lo sovrascrivo col terminatore di stringa */

        if (!strcmp (tmp, a)) /* ora tmp e a hanno la stessa lunghezza */
            result++;

        tmp[alen] = s; /* ripristino il valore originale in posizione b[i+alen] */
    }

    return result;
}
```

Esercizio 5

1) Si scriva un programma C che legga una serie di dati e li memorizzi primo vettore **SQUADRE** (di dimensione 3) contenente strutture (**struct squadra**) del tipo:

-**nome squadra** (*stringa di lunghezza 20*)

-**codice squadra** (*intero*)

-**goal fatti** (*intero*)

-**goal subiti** (*intero*)

2) Stampi a terminale tutti i nomi e codici delle squadre che hanno fatto un numero di goal maggiore del numero dei goal subiti.

3) Letto a terminale un codice di una squadra stampi a video il nome della squadra, i goal fatti e i goal subiti.

Esercizio 5

Attenzione: abbiamo bisogno di un ARRAY di strutture !!!!!

Esempio: l'utente inserisce i seguenti dati per 3 squadre

juventus

1

10

12

milan

2

7

6

inter

3

13

11

<i>juventus</i>	<i>milan</i>	<i>inter</i>
1	2	3
10	7	13
12	6	11

2) Viene stampato a video

milan 2

inter 3

3) Se l'utente digita 1 viene stampato

juventus 10 12

Esercizio 5 – Soluzione (squadre.h)

```
#define N_SQUADRE 3

typedef struct {
    char nome[21];
    int codice;
    int goal_fatti, goal_subiti;
} squadra;

int leggi_squadre(squadra sq[], int dimF);
void stampa_sq_forti(squadra sq[], int dimL);
void stampa_sq_id(squadra sq[], int dimL);
```

Esercizio 5 – Soluzione (squadre.c)

```
#include <stdio.h>
#include "squadre.h"

int leggi_squadre(squadra sq[], int dimF){
    int i;

    printf("Inserisci il numero delle squadre");
    scanf("%d", &ns);
    if(ns>dimF) ns = dimF;

    /* inserimento dati */
    for (i=0; i<ns; i++) {
        printf("Inserisci nome, codice, goal fatti e subiti \n");
        scanf("%s",sq[i].nome);
        scanf("%d",&sq[i].codice);
        scanf("%d",&sq[i].goal_fatti);
        scanf("%d",&sq[i].goal_subiti);
    }

    return ns;
}
```

Esercizio 5 – Soluzione (squadre.c)

```
void stampa_sq_forti(squadra sq[], int dimL){
    int i;
    for (i=0; i<dimL; i++)
        {
            if(sq[i].goal_fatti > sq[i].goal_subiti){
                printf("%s\n",sq[i].nome);
                printf("%d\n",sq[i].codice);
            }
        }
}
```

Esercizio 5 – Soluzione (squadre.c)

```
void stampa_sq_id(squadra sq[], int dimL){
    int i, cod, T;
    printf("Inserisci un codice ");
    scanf("%d", &cod);
    i=0; T=0;
    while ((i < dimL)&& (T==0))
    {
        if (sq[i].codice == cod){
            printf("%s\n",sq[i].nome);
            printf("%d\n",sq[i].goal_fatti);
            printf("%d\n",sq[i].goal_subiti);
            T=1;
        }
        i++;
    }
    if(T==0)
        printf("codice non trovato");
}
```

Esercizio 5 – Soluzione (main.c)

```
#include "squadre.h"

int main() {
    int dimL;
    squadra squadre[N_SQUADRE];

    dimL = leggi_squadre(squadre, N_SQUADRE);

    stampa_sq_forti(squadre, dimL);

    stampa_sq_id(squadre, dimL);

    return 0;
}
```