

Laboratorio di Algoritmi e Strutture Dati

Esercitazioni del 16 Ottobre 2012

Esercizio preliminare

Creare con un editor qualsiasi un file di nome `prova.c`.
Scrivere nel file `prova.c` il programma:

```
#include <stdio.h>

int main(void)
{
    printf("Salve Mondo\n");
    return 0;
}
```

Salvare il file `prova.c`.
Compilare il programma: Aprire un terminale e digitare
`gcc -o ciao prova.c`
Eseguire il programma: digitare `ciao`.

Esercizio 0: input/output, organizzazione del codice

Si tratta di un esercizio di riscaldamento su input/output con `scanf` e `printf`, e sull'organizzazione del codice.

Prima parte

In prima istanza si richiede di implementare l'algoritmo descritto:

1. Stampa del messaggio: `Ciao, come ti chiami?`
2. Lettura di una parola (i.e. sequenza di caratteri consecutivi non di spaziatura) e sua memorizzazione nell'array `char nome[MAXDIM]` dove `MAXDIM` è la seguente macro:


```
#define MAXDIM    80
```
3. Stampa del messaggio: `Quanti anni hai?`
4. Lettura di un intero nella variabile `int anni`.
5. Stampa del messaggio: `Dove vivi? Immetti nome via e numero civico (nell'indicare il nome della via non utilizzare spazi)`
6. Lettura di una parola da memorizzare nell'array `char indirizzo[MAXDIM]`, e di un intero da memorizzare nella variabile `int civico`.
7. Stampa del messaggio: `Dunque ti chiami nome, hai anni anni e vivi in via indirizzo al civico.`, dove le parole in corsivo vanno sostituite col valore delle corrispondenti variabili.

8. Stampa del messaggio: `Il tuo nome e' lungo lunghezza caratteri.`, dove *lunghezza* è la lunghezza in caratteri della stringa `char nome`, senza contare il carattere di terminazione.

Per il calcolo della lunghezza si può usare la funzione di libreria `int strlen(char *s)`, il cui prototipo è in `string.h`.

Si ricordi:

- Sullo schermo non appare alcun output fino a quando la linea di output non è terminata dal carattere `'\n'`.
- La specifica di formato da fornire a `printf` per scrivere una stringa sull'output è `%s`, per scrivere un intero è `%d`.
- La specifica di formato da fornire a `scanf` per leggere una parola dall'input è `%s`, per leggere un intero è `%d`.

Non devono essere presenti variabili *globali*. Tutte le variabili devono quindi essere dichiarate nel corpo di qualche funzione.

Seconda parte

Si distribuisca il codice realizzato nella prima parte su più file in questo modo:

- il file `salvemain.c` deve eseguire l'algoritmo descritto nella prima parte, richiamando le seguenti funzioni:

```
void chiedinome(char *),  
int chiedianni(void),  
int chiediindirizzo(char *),  
void messaggiofinale(char *,int,char *,int),
```

tutte definite altrove.
- Il file `datipersonali.c` deve contenere le definizioni di
 - `void chiedinome(char *)`
che deve implementare i punti 1 e 2 dell'algoritmo.
 - `int chiedianni(void)`
che deve implementare i punti 3 e 4 dell'algoritmo.
- Il file `indirizzo.c` deve contenere la definizione di
 - `int chiediindirizzo(char *)`
che deve implementare i punti 5 e 6 dell'algoritmo.
- Il file `messaggio.c` deve contenere la definizione di
 - `void messaggiofinale(char *,int,char *,int)`
che deve implementare i punti 7 e 8 dell'algoritmo.

Si completi il codice fornendo tutte le informazioni necessarie alla compilazione del progetto.

Anche in questo caso, non si usino variabili *globali*.

Facoltativo

Scrivere il *makefile* del progetto.

Terza parte

Si fornisca una implementazione alternativa di **strlen** (chiamiamola **mstrlen**).

Se invece nella prima parte dell'esercizio non si è usata la funzione di libreria **strlen**, allora si modifichi il codice in modo da utilizzarla.