
PROGRAMMAZIONE I

A.A. 2019 - 2020
CORSO DI LAUREA IN INFORMATICA
DOCENTI: G. DE MARCO R. ZIZZA

PROVA UNICA A DISTANZA - 4 SETTEMBRE 2020

Consegna

Fase 1 (carta e penna). Creare un file pdf (denominato `cognome_nome.pdf`) contenente la scansione di tutti i fogli su cui sono stati svolti gli esercizi e inviarlo come allegato all'indirizzo di consegna.

Fase 2 (implementazione). L'implementazione dell'esercizio di programmazione svolto durante la prima fase va scritta in un file C (denominato `cognome_nome.c`) che dovrà essere inviato all'indirizzo di consegna entro le ore 18:00 del 4 settembre 2020.

Indirizzo di consegna

`gianluca.demarco@gmail.com`

Indicare “esame programmazione” come oggetto e inserire nome, cognome e matricola nel testo del messaggio.

Teoria

Esercizio 1 Il seguente frammento di codice è corretto? Se lo è, indicare cosa fa, altrimenti dire qual è l'errore e come correggerlo.

```
int *h(int *a, int length) {
    int b[length];
    for (int i=0; i<length; i++)
        b[i] = 2*a[i];
    return b;
}
```

. ~ .

Esercizio 2 Quale sarà l'output del seguente programma? Giustificare la risposta.

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

```
void g(int y[], int n) {
    int i;
    for (i=0; i<n; i++) {
        y[i] = y[i]/2;
    }
}
```

```
void h(int a, int *b) {
    while (a > *b) {
        a--;
        (*b)++;
    }
}
```

```
int main(void) {
    int x = 4;
    int a[5] = {1,2,3,4,5};
    g(a,3);
    printf("a[1] = %d\n",a[1]);
    printf("a[4] = %d\n",a[4]);
    h(10, &x);
    printf("valore di x = %d\n", x);
    return 0;
}
```

. ~ .

Esercizio 3 Cosa stampa il seguente programma? Spiegare il compito svolto dalla funzione *f*.

```
#include<stdio.h>

int f(char s[], char t[]){
    char *p=s;
    char *q=t;
    int i = 0;
    while (*p) p++;
    p--;
    while (*p == *q && p>=s && *q){
        p--;
        q++;
        i++;
    }
    *q = '\\0';
    return i;
}

int main(void) {
    char a[]="giocare";
    char b[]="erario";
    printf("Valore della chiamata: %d\\n", f(a,b));
    printf("Stringa a: %s\\n", a);
    printf("Stringa b: %s\\n", b);
    return 0;
}
```

. ~ .

Programmazione

Esercizio 4 Scrivere un programma *laghi* che permetta di esaminare la lista dei laghi italiani memorizzata in un file *laghi.txt*. Il programma può essere usato in tre modi.

1) Prende in input da linea di comando il nome di un lago. Stampa: il nome del lago seguito dalla sua profondità massima e la sua altitudine. Esempio:

```
> ./laghi Garda
Garda 346 65
```

2) Prende in input da linea di comando due interi x e y . Stampa: l'elenco dei laghi la cui altitudine, in metri, è maggiore di x e la cui profondità massima, in metri, è inferiore a y . Esempio:

```
> ./laghi 180 40
Trasimeno 258 6
Varese 238 26
Annone 224 11
Pusiano 257 24
Chiusi 243 8
```

3) Prende in input da linea di comando il carattere d seguito da un intero. Stampa: tra tutti i laghi aventi un'altitudine in *m.s.l.m* non inferiore a quella indicata, quello che ha la minore profondità massima. Esempio:

```
> ./laghi d 400
Caldonazzo 449 49
```

Il file *laghi.txt* contiene l'elenco dei laghi italiani ed ha il seguente formato. Ogni riga corrisponde ad un lago e presenta nell'ordine: il suo nome, la sua area in km^2 , la sua profondità massima in *m*, la sua altitudine in *m.s.l.m* e infine, le regioni e/o stati in cui si trova. Si noti che il numero di spazi fra gli elementi è indefinito e che ciascun nome di regione o stato è costituito da una sola stringa.

```
Garda 370 346 65 Lombardia Veneto Trentino-Alto-Adige
Maggiore 212 372 193 Lombardia Piemonte Svizzera
Como 146 410 197 Lombardia
Trasimeno 128 6 258 Umbria
Bolsena 114 146 305 Lazio
Iseo 65 251 185 Lombardia
Varano 60 6 0 Puglia
...
```

Informazioni utili per l'implementazione

Attenzione: Per la stampa dell'output, ci si attenga strettamente alle indicazioni date sopra e illustrate negli esempi, evitando di introdurre frasi non richieste, come: "I laghi ... sono ..., la profondità ... è ..., ecc. **Soluzioni che ignorino tali indicazioni, rischiano di essere penalizzate o di non essere prese in considerazione!**

Si ricordi che lo specificatore di formato `% [^ set]` consente di far corrispondere ogni sequenza di caratteri non presente in *set*.

. ~ .