
PROGRAMMAZIONE I

ANNO DI PREPARAZIONE E CLASSE 1
CORSO DI LAUREA IN INFORMATICA
DOCENTI: G. DE MARCO R. ZIZZA

PROVA UNICA - 6 LUGLIO 2021

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 23:59 del 6 luglio 2021.

Indirizzo di consegna

`proglresto02@gmail.com`

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

Teoria

Esercizio 1 Indicare se il seguente main è corretto e cosa viene stampato dalle seguenti printf, motivando brevemente la risposta.

```
1 #include <stdio.h>
2
3 int main(void) {
4     int i=6, j=0;
5     printf("valore espressione 1: %d\n", (j--) && (i--));
6     printf("valore i= %d, j= %d\n", i, j);
7
8     char *s="Dieci piccoli indiani";
9     *(s+5)='\n';
10    printf("Risposta stringa: %s\n", s+4);
11 }
```

. ~ .

Esercizio 2 Spiegare quale operazione svolge la funzione f . Indicare l'output del seguente programma e il significato della variabile x . L'output è coerente con il significato della funzione?

```
#include <stdio.h>

int x=2;

int f(int n){
    if (n < 2)
        return 0;
    while (x*x <= n){
        if (n%x == 0)
            return 0;
        x++;
    }
    return 1;
}

int main () {
    printf("%d %d\n", f(13), f(8));
    return 0;
}
```

. ~ .

Esercizio 3 *Il seguente main contiene un errore. Spiegare qual è.*

```
#include <stdio.h>
#define N 6

int main(void) {
    int i, a[]={2,3,4,5,6,7};
    for (i=0; i<N; i++)
        *a++=1;
    return 0;
}
```

. ~ .

Programmazione

Esercizio 4 Si scriva una funzione `void subreverse (char *s1, char *s2)` che prenda in input due stringhe `s1` e `s2`, e le modifichi (sul posto, cioè senza l'ausilio di stringhe di appoggio) rispettando la seguente regola.

Sia `p` il prefisso comune alle due stringhe (che può eventualmente essere anche vuoto), la funzione inverte:

1) la sottostringa di `s1` che comincia con il primo carattere successivo a `p` e finisce con l'ultimo carattere di `s1`.

2) la sottostringa di `s2` che comincia con il primo carattere successivo a `p` e finisce con l'ultimo carattere di `s2`.

Ad esempio se le due stringhe input sono `buongiorno` e `buonasera`, la funzione le modificherà in `buononroig` e `buonaresa` rispettivamente (il prefisso comune è `buon` in questo caso). Se le due stringhe sono `buongiorno` e `arrivederci`, la funzione le modificherà in `onroignoub` e `icredevirra` rispettivamente (il prefisso comune è vuoto in questo caso). Chiaramente, se le due stringhe sono uguali, la funzione dovrà lasciarle inalterate.

Attenzione: Si usi la funzione in un programma che prenda le due stringhe da linea di comando e stampi le stringhe modificate. Il programma può usare solo le librerie `<stdio.h>` e `<stdlib.h>`.

. ~ .

Esercizio 5 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 due modi.

1) Prende in input da linea di comando il nome di un lago. Stampa: il nome del lago seguito dalle regioni/stati in cui si trova. Esempio:

```
> ./laghi Garda
Garda Lombardia Veneto Trentino-Alto-Adige
```

2) Prende in input da linea di comando un intero x e un segno $+/-$. Stampa il nome del lago, seguito dalla sua area in km^2 , che ha la seguente proprietà:

ha la maggiore area tra tutti quelli che hanno un'altitudine superiore a x m.s.l.m se l'utente ha passato il segno $+$, inferiore a x m.s.l.m se l'utente ha passato il segno $-$.

Esempio:

```
> ./laghi 50 -
Varano 60

> ./laghi 60 +
Garda 370
```

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 | | |
| Santa Croce | 8 | 44 | 386 | Veneto | | |
| ... | | | | | | |

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.

. ~ .