
PROGRAMMAZIONE I

A.A. 2020 - 2021
CORSO DI LAUREA IN INFORMATICA
DOCENTI: G. DE MARCO R. ZIZZA

PROVA UNICA - 8 SETTEMBRE 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 dell'8 Settembre 2021.

Indirizzo di consegna

`proglresto02@gmail.com`

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

Attenzione

Per lo svolgimento degli esercizi è proibito usare strumenti di programmazione che non sono stati studiati durante il corso.

Teoria

Esercizio 1 *Indicare cosa fa la seguente funzione e cosa viene stampato dal main, giustificando la risposta.*

```
#include <stdio.h>

void funz(int n, int A[ ]){
    int m, i, j;
    for (i=0; i<n; i++) {
        m=A[i];
        for (j=i+1; j<n; j++) {
            if (A[j]<m) {
                m=A[j];
                A[j]=A[i];
                A[i]=m;
            }
        }
    }
}

int main(void) {
    int a[]={2,45,1,14,7,23,13,5,6,50};
    funz(4,&a[2]);
    int i;
    for (i=0; i<10;i++) printf("%d \n", a[i]);
    return 0;
}
```

. ~ .

Esercizio 2 *La seguente funzione contiene un errore. Indicare qual è e come correggerlo, giustificando la risposta.*

```
int* f (int a[], int n) {
    if (n-2 <= 0)
        return NULL;
    int b[n-2];
    int i = 0;
    while (i < n-2) {
        b[i] = a[i];
        i++;
    }
    return b;
}
```

. ~ .

Esercizio 3 *Il seguente programma contiene un errore. Indicare qual è, giustificando la risposta. Una volta corretto, indicare cosa fa la funzione e cosa stampa il main.*

```
#include <stdio.h>
char* f(char *p1, char *p2, char *p3) {
    char* tp1 = p1;
    char* tp2 = p2;
    char* tp3 = p3;
    while (*tp1) tp1++;
    while (*tp2) tp2++;
    while (tp1>p1) *tp3++ = *--tp1;
    while (tp2>p2) *tp3++ = *--tp2;
    *tp3=0;
    return p3;
}

int main(void) {
    char* string1="abcd";
    char* string2="efg";
    char* string3="\0";
    char* ris=f(string1,string2,string3);
    printf("%s\n", ris);
}
```

. ~ .

Programmazione

Esercizio 4 Si scriva una funzione `void create (char *s1, char *s2, int a, int b)` che prenda in input una stringa `s1`, una stringa `s2` e due interi `a` e `b`, e formi la stringa `s2` copiando in essa la parte di `s1` è lunga `b` caratteri cominciando dal suo carattere `a`-esimo. Se la stringa `s1` non è sufficientemente lunga allora `s2` sarà la stringa nulla.

Si scriva la funzione in un programma che prenda i parametri `s1`, `a` e `b` dalla linea di comando e stampi la stringa `s2` generata (`s2` dovrà essere dichiarata nel `main`). Esempi:

```
> ./a.out buongiorno 5 6
giorno
```

```
> ./a.out buongiorno 5 7
```

Si noti che nel secondo esempio il programma non stampa nulla perché `buongiorno` non contiene 7 caratteri dopo il quinto. In questo caso la funzione avrà formato `s2` con la sola stringa nulla.

Attenzione: Il programma può usare solo le librerie `<stdio.h>` e `<stdlib.h>`.

. ~ .

Esercizio 5 Scrivere un programma `treni` che permetta di esaminare l'elenco delle partenze dei treni dalla stazione di *Paperopoli Centrale* a quella di *Topolinia Porta Nuova*.

L'elenco è memorizzato in un file `orari.txt`. Ogni linea del file corrisponde ad un treno e presenta i seguenti elementi separati da uno o più spazi: orario di partenza, orario di arrivo, numero di cambi. Gli orari sono nel formato `hh:mm`.

Il programma può essere usato in due modi.

1) Prende in input da linea di comando la lettera `q` seguita da un orario `hh:mm`. Stampa l'orario di partenza e l'orario di arrivo dei treni che partono all'ora `hh:mm` indicata.

```
> ./treni q 12:57
12:57 16:25
12:57 16:35
```

2) Prende in input da linea di comando la lettera `p` seguita da un orario `hh:mm` e da un numero `x`. Stampa l'elenco dei treni che partono entro le ore `hh:mm` e che fanno non più di `x` cambi. Per ciascun treno dovrà stampare, nell'ordine: orario di partenza, orario di arrivo e numero di cambi.

```
> ./treni p 13:50 1
12:30 16:05 0
12:57 16:25 1
12:57 16:35 0
13:23 17:00 0
```

Il file `orari.txt` ha il seguente formato. Si noti che l'elenco **non è ordinato** e che il numero di spazi fra gli elementi è indefinito.

```
12:30 16:05 0
12:31 19:48 2
12:57 16:25 1
12:57 16:35 0
13:23 17:00 0
...
```

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

. ~ .