

Esame Completo di Programmazione I - Bioinformatica

21 luglio 2021 (tempo disponibile: 2 ore)

Esercizio 1 (15 punti) (si consegna `ordina.c`)

Si completi il seguente programma `ordina.c` che implementa le seguenti funzioni:

```
// inizializza l'array lungo length con numeri casuali tra 0 e 4 inclusi
void init_random(int arr[], int length) {
    // DA COMPLETARE
}

// stampa su una riga l'array lungo length e poi va a capo
void print(int arr[], int length) {
    // DA COMPLETARE
}

// calcola quante volte value e' presente dentro l'array lungo length
int how_many(int arr[], int length, int value) {
    // DA COMPLETARE
}

// determina se arr[pos] e' in ordine errato col successore arr[pos + 1],
// cioe' arr[pos] e' meno frequente di arr[pos + 1]
// oppure entrambi hanno la stessa frequenza ma arr[pos] e' maggiore di arr[pos + 1];
// si assuma che 0 <= pos < length - 1
int must_swap(int arr[], int length, int pos) {
    // DA COMPLETARE
}

// mette in ordine l'array in modo che gli elementi piu' frequenti siano
// messi al suo inizio; a parita' di frequenza, mette prima gli elementi minori
void more_frequent_first(int arr[], int length) {
    // DA COMPLETARE
}

int main(void) {
    int arr[20];
    init_random(arr, 20);
    print(arr, 20); // stampa prima dell'ordinamento
    more_frequent_first(arr, 20);
    print(arr, 20); // stampa dopo l'ordinamento
    return 0;
}
```

L'esecuzione del programma dovrà stampare qualcosa del tipo:

```
1 4 3 3 0 3 0 4 3 4 2 4 4 2 4 4 1 2 2
4 4 4 4 4 4 4 4 2 2 2 2 3 3 3 3 0 0 1 1
```

In questo esempio, il valore più frequente nell'array è 4, presente otto volte, e infatti viene spostato all'inizio. Poi seguono il 2 e il 3, che sono presenti quattro volte ciascuno, e si mette prima il 2 perché è minore. Poi seguono lo 0 e l'1, che sono presenti due volte ciascuno, e si mette prima lo 0 perché è minore.

Esercizio 2 (16 punti)

(si consegna calcoli.c)

Si completi il seguente programma `calcoli.c`, la cui funzione `main()` calcola la percentuale di equazioni corrette presenti in un file il cui contenuto è una sequenza (di lunghezza ignota) di uguaglianze tra somme di interi positivi, ciascuna terminata da un punto e virgola, e senza spazi bianchi. Il file di esempio potrebbe essere

```
2+3+10=7+8;
1+2+3+4+5=15;
33=3+4+5+6+7+8;
3+5+4=1+18;
```

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

int myerror(char *message);
int myfclose(FILE *f);
FILE * myfopen(char *name, char *mode);
float calcola(FILE* fp);

int main()
{
    char filename[6] = "in.txt";
    FILE* fp;
    fp = myfopen(filename,"r");
    float risultato = calcola(fp);
    myfclose(fp);
    printf("Le equazioni corrette sono il %.2f %% \n", (risultato*100));
    return 0;
}

int myerror(char *message)
{
    fputs( message, stderr );
    exit(1);
}

/**
 * chiude lo stream su file controllando se ci sono errori,
 * invoca myerror sia che f sia null sia che il risultato delle
 * chiusura sia diverso da zero
 */
int myfclose(FILE *f)
{
    // DA COMPLETARE
}

/**
 * apre lo stream su file controllando se ci sono errori,
 * invoca myerror se f \'{e} null
 */
FILE * myfopen(char *name, char *mode)
{
    // DA COMPLETARE
}

/**
 * legge un file identificato da FILE * f, distingue la singole equazioni
 * grazie al ";" che ne determina la fine,
 * restituisce la frazione di equazioni corrette sul totale
 */
float calcola(FILE* fp)
{
    int parte_sin = 1;
    // DA COMPLETARE
}
```

La sua esecuzione, con il file di esempio sopra specificato, dovrà stampare sul video:

Le equazioni corrette sono il 75.00 %

Si osservi che:

- La soluzione richiesta ignora la divisione in righe e si basa soltanto sui caratteri che seguono i numeri (cioè +, = e ;)
- Suggerimento: la soluzione richiesta può utilizzare una variabile booleana, chiamata `parte_sin`, che tiene traccia del fatto che si sta analizzando la parte a sinistra o a destra del segno di uguaglianza.