

Compito di Programmazione I - BioInformatica

13 settembre 2018 (tempo disponibile: 2 ore)

Esercizio 1 (6 punti)

Che numero stampa l'esecuzione del seguente programma?

```
#include <stdio.h>

int foo(int arr[], int length) {
    int counter = 0, i;
    for (i = 0; i < length - 1; i += 2)
        if (arr[i] > arr[i + 1])
            counter++;

    return counter;
}

int main(void) {
    int original[] = { 6, 6, -6, -5, 8, 1, 4, 4, 3 };
    printf("%d\n", foo(original, 9)); // cosa stampa?

    return 0;
}
```

Esercizio 2 (9 punti)

Si scriva una funzione `divide` che riceve tre parametri: un array `original` di `int`, un array `destination` di `double` e la lunghezza dei due array (che si assume essere la stessa per entrambi). La funzione deve modificare `destination` in modo che ogni suo elemento in posizione i diventi l'elemento di `original` in posizione i diviso per $i + 1$ (risultato con la virgola). Per esempio, l'esecuzione del seguente programma:

```
#include <stdio.h>

// la funzione divide va aggiunta qui

int main(void) {
    int i, original[] = { 5, 4, 3, 2, 1, 0 };
    double destination[6];
    divide(original, destination, 6);
    for (i = 0; i < 6; i++)
        printf("%.2f ", destination[i]);
    return 0;
}
```

dovrà stampare 5.00 2.00 1.00 0.50 0.20 0.00.

Esercizio 3 (11 punti)

Si scriva un file `registro.h` che definisce una struttura `registro` che implementa un registro per una campagna raccolta punti. I partecipanti alla campagna sono al massimo 10 e sono identificati per cognome. La struttura `registro` sarà fatta da un array di 10 stringhe (i cognomi dei partecipanti) e da un array di 10 `double` (i punti di ciascun partecipante) e non dovrà avere altre componenti. Si scriva il file `registro.c` che implementa le seguenti funzioni, dichiarandole in `registro.h`:

- `struct registro *construct_registro()` che restituisce un nuovo registro vuoto;
- `void destruct_registro(struct registro *this)` che dealloca il registro `this`;
- `void aggiungi_punti(struct registro *this, char *cognome, int punti)`, che aumenta di `punti` i punti del partecipante `cognome`, se si trova in `this`. Se il partecipante non è in `this` e non si è ancora arrivati al massimo di 10 partecipanti, questa funzione lo aggiunge come nuovo partecipante con punti iniziali pari a `punti`. Se il partecipante non è in `this` e si è già arrivati a 10 partecipanti, questa funzione non fa nulla;
- `void bonus(struct registro *this, double percent)`, che aumenta i punti di ciascun partecipante del `percent` per cento.

Esercizio 4 (6 punti)

Cosa stampa il seguente programma C?

```
#include <stdio.h>

void f(int *p){
    *p = 5;
}

void g(int *p){
    int i = 12;
    p = &i;
}

int main(void){
    int n = 8;
    int *p = &n;
    printf("Il valore puntato da p e' uguale a %d.\n", *p);
    f(p);
    printf("Il valore puntato da p e' uguale a %d.\n", *p);
    g(p);
    printf("Il valore puntato da p e' uguale a %d.\n", *p);
}
```