

Parziale di Programmazione I - Bioinformatica

31 gennaio 2022, turno delle 11:30 (tempo disponibile: 2 ore)

Esercizio 1 (18 punti) (si consegnino `max.c` e `max.h`)

Si scriva un programma `max.c` che implementa le seguenti funzioni:

```
// inizializza l'array, lungo length, con numeri casuali tra 0 e 9 inclusi
void init(int array[], int length) { ... }

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

// restituisce l'elemento dell'array, lungo length, il cui valore,
// scritto in binario, contiene piu' '1'. Se non esistesse, ritorna -1.
// In caso di parita' fra due elementi, restituisce quello piu' a sinistra.
// Si dia per scontato che gli elementi dell'array siano non negativi.
// L'array non deve venire modificato
int max(int array[], int length) { ... }
```

Si scriva quindi un file di header `max.h` che dichiara le precedenti funzioni.

Per esempio, il seguente file `main.c` (già fornito e da non modificare):

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

int main(void) {
    int array1[8];
    init(array1, 8);
    printf("array1: ");
    print(array1, 8);
    printf("Elemento con piu' '1' in binario dentro array1: %i\n", max(array1,
        8));
    int array2[] = { 4, 4, 8, 1 };
    printf("array2: ");
    print(array2, 4);
    printf("Elemento con piu' '1' in binario dentro array2: %i\n", max(array2,
        4));
    int array3[] = { };
    printf("array3: ");
    print(array3, 0);
    printf("Elemento con piu' '1' in binario dentro array3: %i\n", max(array3,
        0));
    return 0;
}
```

stampa qualcosa del tipo:

```
array1: 6 7 2 8 0 7 7 5
Elemento con piu' '1' in binario dentro array1: 7
array2: 4 4 8 1
Elemento con piu' '1' in binario dentro array2: 4
array3:
Elemento con piu' '1' in binario dentro array3: -1
```

Esercizio 2 (13 punti) (si consegna bar.c)

Si completi il seguente programma `bar.c` in modo che la funzione `bar` stampi una riga fatta dai caratteri da `c` fino a `'z'` incluso e poi da `'z'` indietro fino a `c` incluso. **La funzione `bar` deve essere ricorsiva:**

```
#include <stdio.h>

void bar(char c) {
    // si dia per scontato che c sia un carattere fra 'a' e 'z' inclusi
    // DA COMPLETARE, DEVE ESSERE RICORSIVA
}

int main(void) {
    // DA COMPLETARE:
    // 1) legge il carattere c con scanf(" %c", &c) ad oltranza
    //   finche' non e' tra 'a' e 'z'
    // 2) chiama bar(c)
    // 3) va a capo
    return 0;
}
```

Per esempio, un utilizzo del programma potrebbe essere il seguente:

```
Inserisci un carattere alfabetico minuscolo: d
defghijklmnopqrstuvwxyzzyxwvutsrqponmlkjihgfed
```