

Parziale di Programmazione I - Bioinformatica

1 febbraio 2021, turno delle 13:00 (tempo disponibile: 2 ore)

Esercizio 1 (20 punti) (si consegnì `hex.c` e `hex.h`)

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

```
// inizializza arr, lungo length, con numeri interi casuali pari tra 0 a 1000,  
// usando srand() e rand()  
void init_random(int arr[], int length);  
  
// stampa su un'unica riga, in esadecimale, gli elementi  
// dell'array arr, lungo length, poi va a capo;  
// si ricorda che esiste il formato %x di printf per stampare  
// un intero in esadecimale  
void print(int arr[], int length);  
  
// determina se il numero n non negativo e' esalelettere,  
// cioe', tradotto in esadecimale, usa solo le cifre a, b, c, d, e, f  
int is_hexletter(int n);  
  
// modifica l'array, lungo length, in modo da spostare al suo inizio i suoi elementi  
// esalelettere e alla sua fine i suoi elementi non esalelettere  
void hexletters_first(int arr[], int length);
```

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

Esercizio 2 (12 punti) (si consegnì `main.c`)

Si scriva un programma `main.c` che include le funzioni dell'Esercizio 1 tramite il file `hex.h`. Il programma `main.c` deve contenere una funzione iniziale `main` che esegue le seguenti operazioni:

1. legge da tastiera la lunghezza `length` di un array, richiedendola ad oltranza se fosse inserita negativa;
2. crea un array `elements` di `length` interi;
3. chiama la funzione `init_random` per inizializzare `elements` in modo casuale;
4. chiama la funzione `print` per stampare `elements`;
5. chiama la funzione `hexletters_first` con l'array `elements` come parametro;
6. chiama la funzione `print` per stampare `elements`.