

# Parziale di Programmazione I - Bioinformatica

1 febbraio 2023, turno delle 11:30 (tempo disponibile: 2 ore)

## Esercizio 1 (20 punti)

(si consegnino `pari_dispari.c` e `pari_dispari.h`)

Si completi il seguente file `pari_dispari.c`:

```
// aggiungete #include se servono

// inizializza arr, lungo length, con numeri casuali tra -5 e 25 inclusi
// in modo da non avere mai ne' due numeri pari in posizioni consecutive
// ne' due numeri dispari in posizioni consecutive; non randomizza
// la sequenza casuale con srand(time(NULL)): ci pensa il main
void init(int arr[], int length) {
    // completare
}

// stampa arr, lungo length, separando gli elementi
// con uno spazio, e va a capo
void print(int arr[], int length) {
    // completare
}

// determina se in arr, lungo length, la quantita' dei numeri dispari
// e' maggiore o uguale alla quantita' dei numeri pari; si assuma
// che arr non abbia mai ne' due numeri pari in posizioni consecutive
// ne' due numeri dispari in posizioni consecutive
int dispari_almeno_quanto_i_pari(int arr[], int length) {
    // completare
}
```

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

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

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "pari_dispari.h"

int main(void) {
    srand(time(NULL));
    int array1[20];
    init(array1, 20);
    printf("array1: ");
    print(array1, 20);
    printf("In array1 dispari >= pari ? %i\n", dispari_almeno_quanto_i_pari(
        array1, 20));
}
```

```

int array2[15];
init(array2, 15);
printf("array2: ");
print(array2, 15);
printf("In array2 dispari >= pari ? %i\n", dispari_almeno_quanto_i_pari(
    array2, 15));
int array3[17];
init(array3, 17);
printf("array3: ");
print(array3, 17);
printf("In array3 dispari >= pari ? %i\n", dispari_almeno_quanto_i_pari(
    array3, 17));
return 0;
}

```

stampa qualcosa del tipo:

```

array1: 7 2 5 0 11 18 15 24 19 20 -1 2 13 2 17 16 13 2 21 8
In array1 dispari >= pari ? 1
array2: 5 20 5 0 11 12 3 16 9 10 23 22 1 4 11
In array2 dispari >= pari ? 1
array3: 2 7 6 23 0 3 6 -1 -2 7 6 13 24 13 4 9 8
In array3 dispari >= pari ? 0

```

## Esercizio 2 (11 punti)

(si consegna triangular.c)

Si completi il seguente programma triangular.c. La funzione triangular deve essere ricorsiva:

```

#include <stdio.h>

// stampa i volte c, poi i-1 volte il carattere precedente a c
// poi i-2 volte il carattere precedente al carattere precedente a c, ecc.
void triangular(char c, int i) { // completare
}

int main(void) {
    triangular('j', 5);
    printf("\n");
    triangular('s', 8);
    printf("\n");
    return 0;
}

```

Se tutto è corretto, tale programma dovrebbe stampare:

```

jjjjjjiiihhhggf
ssssssssrrrrrrrrqqqqqqppppppooooonnnmml

```