

Parziale di Programmazione I - Bioinformatica

1 febbraio 2023, turno delle 15:00 (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 non negativi divisibili
// per 5 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 dichiari 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: 1390741700 1729062575 476161820 466897965 399090190 26084305 251372970 799201515
1094977550 670565265 1743740570 186827875 1663476810 1017154705 904737510 597427485
2044555980 678609605 212097830 807774915
In array1 dispari >= pari ? 1
array2: 13150390 1810090715 1792295540 466060805 1560237680 283637865 247794490 855049935
1862099380 1077922515 524959800 369117805 1744420990 630703485 418630810
In array2 dispari >= pari ? 0
array3: 2007145705 1903055230 1870673145 2008279780 1306383695 60525760 1518577195
1129157370 1796086945 38722370 2093968385 274308190 899586285 1623976460 1024382915
434370140 530517665
In array3 dispari >= pari ? 1

```

Esercizio 2 (11 punti)

(si consegnì 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:

```

jjjjjjiiiihhhgff
ssssssssrrrrrrrqqqqqqppppppooooonnnmml

```