

Secondo parziale di Programmazione I - Bioinformatica

14 giugno 2022 (tempo disponibile: 2 ore)

INDICAZIONI GENERALI

- Utilizzare il comando `ulimit -v 500000` per limitare l'utilizzo delle risorse al terminale su cui viene eseguito il comando ed evitare spiacevoli inconvenienti dovuti ad eccessive allocazioni di memoria.
 - Scaricare il file di ogni esercizio e riconsegnarlo senza modificarne il nome.
 - I file non consegnati o consegnati con errori di compilazione non verranno presi in considerazione.
 - Si possono utilizzare funzioni aggiuntive non presenti nei file modello e aggiungere linee di commento alle funzioni già implementate nel modello.
 - I file possono essere consegnati più volte. Per ogni esercizio, solo l'ultimo file consegnato sarà considerato valido.
 - Compilare con l'opzione `-Wall` per avere tutti i warning.
-

Esercizio 1 (10 punti) (si consegni `sinh1.c`)

Completare il programma `sinh1.c` che ha la seguente funzione `main()`:

```
int main(){
    printf("Valore di sinh1: %lf\n", sinh1(10));
    return 0;
}
```

con le funzioni:

- `f()` che calcola, preferibilmente in modo ricorsivo, il fattoriale di un numero `n`;
- `sinh1()` che calcola, preferibilmente in modo ricorsivo, la somma dei primi `n` termini della serie sequente che rappresenta lo sviluppo di Taylor-Mc Laurin del seno iperbolico:

$$\frac{1}{(2 \cdot 0 + 1)!} + \frac{1}{(2 \cdot 1 + 1)!} + \frac{1}{(2 \cdot 2 + 1)!} + \frac{1}{(2 \cdot 3 + 1)!} + \frac{1}{(2 \cdot 4 + 1)!} + \dots$$

in modo che produca il seguente output:

Valore di sinh1: 1.175201

Esercizio 2 (14 punti)

(si consegna odd_element_list.c)

Si completi il programma odd_element_list.c, la cui funzione main() crea e poi stampa una lista concatenata con i soli numeri dispari presenti in un file di testo contenente dei numeri interi. Per esempio, se viene fornito in input il file numbers.txt seguente:

```
8 3 1 3
9 4
6 7 7
```

il programma stampa nella console:

```
3 1 3 9 7 7
```

Esercizio 3 (9 punti)

(si consegna matrix_density.c)

Completare il file matrix_density.c che ha la seguente funzione main():

```
int main() {
    int matrix1[SIZE][SIZE] = {{0, 2, 0}, {1, 0, 0}, {4, 0, 8}};
    int matrix2[SIZE][SIZE] = {{0, 2, 0}, {4, 0, 6}, {0, 8, 9}};

    print_matrix(matrix1, SIZE);
    printf("Matrice densa? %i\n---\n", is_dense(matrix1, SIZE));

    print_matrix(matrix2, SIZE);
    printf("Matrice densa? %i\n---\n", is_dense(matrix2, SIZE));

    return 0;
}
```

in modo che produca il seguente output:

```
0 2 0
1 0 0
4 0 8
Matrice densa? 0
---
0 2 0
4 0 6
0 8 9
Matrice densa? 1
---
```