

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 (18 punti) (si consegnì `letter_list.c`)

Si completi programma `letter_list.c`, in modo che la funzione `main()` riportata di seguito:

```
int main() {
    int n = 10;
    struct letter_t *head = NULL;
    srand(0);
    for (int i = 0; i < n; i++) {
        char ch = 'A' + rand() % NUM_OF_LETTERS;
        head = insert_letter_in_tail(head, ch);
    }
    print_list(head);
    printf("Occorrenze di A: %d\n", count_letter(head, 'A'));
    return 0;
}
```

produca il seguente output:

```
B B A B C B B A A B
Occorrenze di A: 3
```

Esercizio 2 (14 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
---
```