

Compito di Programmazione I - Bioinformatica

5 luglio 2022 (tempo disponibile: 2 ore)

Esercizio 1 (15 punti) (si consegnì max_subarray.c)

Si completi il programma max_subarray.c in modo che la funzione:

```
void max_subarray(int arr[], int length, int *start, int *end)
```

identifichi un sottoarray di `arr` (cioè una porzione contigua di `arr`) i cui elementi abbiano somma massima. Per indicare dove si trova tale sottoarray, la funzione scriverà gli estremi del sottoarray nelle variabili puntate da `start` (incluso) ed `end` (escluso). L'array `arr` è lungo `length`. Per esempio, nel `main` già fornito con il compito:

```
int main(void) {  
    int arr[] = {-2, 1, -3, 4, -1, 2, 1, -5, 4};  
    int start, end;  
    max_subarray(arr, 9, &start, &end);  
    printf("Un sottoarray di somma massima e': ");  
    for (int pos = start; pos < end; pos++)  
        printf("%i ", arr[pos]);  
    printf("\n");  
    return 0;  
}
```

la funzione `sub_array` dovrà identificare che un sottoarray massimo è 4, -1, 2, 1 e quindi memorizzerà 3 dentro `start` (posizione di inizio, inclusa) e 7 dentro `end` (posizione di fine, esclusa). L'esecuzione di tale `main` stamperà quindi:

```
Un sottoarray di somma massima e': 4 -1 2 1
```

Esercizio 2 (16 punti)

(si consegna odd_element_list.c)

Si completi il programma odd_element_list.c, la cui funzione main() seguente:

```
int main() {  
    char filename[] = "numbers.txt";  
    struct element_t *list = create_odd_list(filename);  
    print_list(list);  
    return 0;  
}
```

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 fornito in allegato:

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

il programma stampa nella console:

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