

Laboratorio di programmazione strutturata (STM) - A.A. 2014-2015 -

Secondo appello - luglio 2015

ESERCIZI

Esercizio 1 (18 punti)

Siano $S1$ e $S2$ due liste di interi tali che all'interno di una stessa lista non siano presenti elementi ripetuti.

1. Si definisca il nodo lista
2. Si scriva una funzione che, prende in input le liste $S1$ e $S2$ e restituisce una nuova lista SD contenente gli elementi in $S1$ che non sono presenti anche in $S2$. Gli elementi di SD devono trovarsi nello stesso ordine che hanno in $S1$.
3. Si scriva una funzione che prende in input le liste $S1$ e $S2$ e restituisce una nuova lista SI contenenti gli elementi di $S1$ che sono presenti anche in $S2$. Gli elementi di SI dovranno essere in ordine inverso rispetto a quello in cui si trovano in $S1$.
4. Opzionale (+4 punti) Si scriva una funzione ricorsiva che verifica se la lista $S2$ è completamente contenuta nella lista $S1$.

Esercizio 2 (6 punti)

Si riscriva il seguente segmento di programma eliminando l'istruzione **break** e usando delle istruzioni **while** al posto delle istruzioni **for**.

```
for (i=1; i<=26; i++){
    for (j=16; j>0; j--){
        num = num * i+j;
        if(num == 64)
            break;
    }
}
```

Esercizio 3 (6 punti)

Si scriva la funzione `void ordina(int vet[], int s)` che ordina in modo decrescente un vettore **vet** di dimensione **s**, utilizzando l'algoritmo di ordinamento selectionSort.