

## 1 Esercizio 3<sub>1</sub>

Scrivere nel file `esercizio3.cc` la dichiarazione e la definizione della funzione `creaLista` che prende come argomenti un array di interi `arrayDiInteri` e la dimensione dell'array `dimensioneArray`. La funzione `creaLista` deve creare una lista doppiamente concatenata contenente gli elementi dell'array e ritornare il primo nodo di tale lista. Dopo aver creato la lista, scrivere la dichiarazione e la definizione della funzione `rimuoviNodiAlternati` che prende come argomento il nodo iniziale della lista `nodoIniziale`. La funzione `rimuoviNodiAlternati` deve rimuovere dalla lista ogni coppia di nodi alternati (cioè inframezzati da un altro nodo) in cui il valore del primo nodo è uguale al valore del secondo nodo, e infine ritornare il nodo iniziale della lista così modificata. Quando una coppia di nodi alternati vengono rimossi, i nodi rimanenti della lista devono essere correttamente ricollegati. Notare che questa operazione porta nodi della lista, precedentemente staccati, ad essere adesso alternati.

Questi sono due esempi di esecuzione:

```
computer > ./a.out
Lista iniziale: 1 1 2 0 1 0 0 2
Risultato: 1 0
```

```
computer > ./a.out
Lista iniziale: 1 0 2 0 1 1 0 2
Risultato: 2 1 0 2
```

**Note:**

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione delle funzioni `creaLista` e `rimuoviNodiAlternati` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito. Ricordarsi di deallocare la memoria;
- Le coppie di nodi alternati in cui il valore del primo nodo è uguale al valore del secondo nodo devono essere rimosse in ordine. Per esempio, data la lista "0 1 0 1", il risultato dovrà essere "1 1" e non "0 0";
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema.
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.

## 2 Esercizio 3<sub>2</sub>

Scrivere nel file `esercizio3.cc` la dichiarazione e la definizione della funzione `creaLista` che prende come argomenti un array di interi `arrayDiInteri` e la dimensione dell'array `dimensioneArray`. La funzione `creaLista` deve creare una lista doppiamente concatenata contenente gli elementi dell'array e ritornare il primo nodo di tale lista. Dopo aver creato la lista, scrivere la dichiarazione e la definizione della funzione `rimuoviNodiAlternati` che prende come argomento il nodo iniziale della lista `nodoIniziale`. La funzione `rimuoviNodiAlternati` deve rimuovere dalla lista ogni coppia di nodi alternati (cioè inframezzati da un altro nodo) in cui il valore del primo nodo è diverso del valore del secondo nodo, e infine ritornare il nodo iniziale della lista così modificata. Quando una coppia di nodi alternati vengono rimossi, i nodi rimanenti della lista devono essere correttamente ricollegati. Notare che questa operazione porta nodi della lista, precedentemente staccati, ad essere adesso alternati.

Questi sono due esempi di esecuzione:

```
computer > ./a.out
Lista iniziale: 0 2 1 0 1 2 1 0
Risultato: 2 0

computer > ./a.out
Lista iniziale: 0 1 0 1 2 1 2 0
Risultato: 1 0
```

### Note:

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione delle funzioni `creaLista` e `rimuoviNodiAlternati` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito. Ricordarsi di deallocare la memoria;
- Le coppie di nodi alternati in cui il valore del primo nodo è diverso dal valore del secondo nodo devono essere rimosse in ordine. Per esempio, data la lista "0 1 1 0", il risultato dovrà essere "1 0" e non "0 1";
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema.
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.

### 3 Esercizio 3<sub>3</sub>

Scrivere nel file `esercizio3.cc` la dichiarazione e la definizione della funzione `creaLista` che prende come argomenti un array di interi `arrayDiInteri` e la dimensione dell'array `dimensioneArray`. La funzione `creaLista` deve creare una lista doppiamente concatenata contenente gli elementi dell'array e ritornare il primo nodo di tale lista. Dopo aver creato la lista, scrivere la dichiarazione e la definizione della funzione `rimuoviNodiAlternati` che prende come argomento il nodo iniziale della lista `nodoIniziale`. La funzione `rimuoviNodiAlternati` deve rimuovere dalla lista ogni coppia di nodi alternati (cioè inframezzati da un altro nodo) in cui il valore del primo nodo è maggiore del valore del secondo nodo, e infine ritornare il nodo iniziale della lista così modificata. Quando una coppia di nodi alternati vengono rimossi, i nodi rimanenti della lista devono essere correttamente ricollegati. Notare che questa operazione porta nodi della lista, precedentemente staccati, ad essere adesso alternati.

Questi sono due esempi di esecuzione:

```
computer > ./a.out
Lista iniziale: 2 1 1 0 0 0 1 2
Risultato: 0 0 1 2
```

```
computer > ./a.out
Lista iniziale: 1 1 0 2 0 2 1 0
Risultato: 2 0
```

**Note:**

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione delle funzioni `creaLista` e `rimuoviNodiAlternati` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito. Ricordarsi di deallocare la memoria;
- Le coppie di nodi alternati in cui il valore del primo nodo è maggiore del valore del secondo nodo devono essere rimosse in ordine. Per esempio, data la lista "1 2 0 0", il risultato dovrà essere "2 0" e non "1 0";
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema.
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.

1 2 0 0

## 4 Esercizio 3<sub>4</sub>

Scrivere nel file `esercizio3.cc` la dichiarazione e la definizione della funzione `creaLista` che prende come argomenti un array di interi `arrayDiInteri` e la dimensione dell'array `dimensioneArray`. La funzione `creaLista` deve creare una lista doppiamente concatenata contenente gli elementi dell'array e ritornare il primo nodo di tale lista. Dopo aver creato la lista, scrivere la dichiarazione e la definizione della funzione `rimuoviNodiAlternati` che prende come argomento il nodo iniziale della lista `nodoIniziale`. La funzione `rimuoviNodiAlternati` deve rimuovere dalla lista ogni coppia di nodi alternati (cioè inframezzati da un altro nodo) in cui il valore del primo nodo è minore del valore del secondo nodo, e infine ritornare il nodo iniziale della lista così modificata. Quando una coppia di nodi alternati vengono rimossi, i nodi rimanenti della lista devono essere correttamente ricollegati. Notare che questa operazione porta nodi della lista, precedentemente staccati, ad essere adesso alternati.

Questi sono due esempi di esecuzione:

```
computer > ./a.out
Lista iniziale: 0 1 1 0 2 2 1 0
Risultato: 2 0

computer > ./a.out
Lista iniziale: 1 2 0 1 1 0 2 0
Risultato: 1 2 0 0
```

**Note:**

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione delle funzioni `creaLista` e `rimuoviNodiAlternati` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito. Ricordarsi di deallocare la memoria;
- Le coppie di nodi alternati in cui il valore del primo nodo è minore del valore del secondo nodo devono essere rimosse in ordine. Per esempio, data la lista "0 0 1 2", il risultato dovrà essere "0 2" e non "0 1";
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema.
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`.