

**Pemrograman Deklaratif**  
**Semester Genap 2014/2015**  
**Fakultas Ilmu Komputer, Universitas Indonesia**  
**- Operasi & Fungsi terkait List, Soal A -**

Ingat bahwa pada *functional programming*, seandainya kita melakukan modifikasi pada List, maka kita sebenarnya akan mengembalikan **List baru** yang dibangun kembali dari list kosong.

**Generating List**

1. Deklarasikan fungsi **downto1 : int -> int list** sehingga

**downto n = [n; n-1; n-2; ...;1]**

2. Deklarasikan fungsi **upto : int -> int list** sehingga

**upto n = [1; 2; 3; ...;n]**

3. Deklarasikan fungsi **evenN : int -> int list** sehingga **evenN n** akan menghasilkan list yang berisi “n bilangan genap pertama yang bukan negatif”.

Contoh : **evenN 5** akan menghasilkan **[2;4;6;8;10]**

4. Deklarasikan fungsi **doubleList : int list -> int list** sehingga

**doubleList [x1; x2; x3; ...; xn] = [2\*x1; 2\*x2; 2\*x3; ...; 2\*xn]**

Contoh : **doubleList [1;2;3;4]** akan mengembalikan **[2;4;6;8]**

**Menghitung properti dari List**

5. Deklarasikan fungsi **numOdd : int list -> int** yang mengembalikan nilai banyaknya bilangan ganjil dalam sebuah list of integers.

Contoh : **numOdd [2;3;4;5;6;7;8]** akan mengembalikan 3

6. Deklarasikan fungsi **jumOdd : int list -> int** yang menjumlahkan semua elemen-elemen ganjil pada sebuah list of integers.

Contoh : **jumOdd [2;3;4;5;6;7;8]** akan mengembalikan 15

7. Deklarasikan fungsi **multiplicity (x, ls)** yang bertipe **int \* int list -> int**. Fungsi ini mengembalikan nilai banyaknya kemunculan elemen **x** di dalam list **ls**.

Contoh : **multiplicity** (2, [2;3;4;2;5;2;6;2]) akan mengembalikan 4

8. Deklarasikan fungsi **listParity** **ls** : **'a list -> bool** yang mengembalikan **True** jika panjang list adalah genap dan mengembalikan **False** jika sebaliknya. **Hint**: agar lebih elegan, coba gunakan operasi **not** dalam hal ini.

Contoh : **listParity** [3;2;4;5] akan mengembalikan **True**

### Removing and Filtering

9. Deklarasikan fungsi **removeEven** : **int list -> int list** yang membuang semua elemen genap pada sebuah list.

Contoh : **removeEven** [1;2;3;4;5;6] akan mengembalikan [1;3;5]

10. Deklarasikan fungsi **throwLastElmt** : **'a list -> 'a list** yang membuang elemen terakhir pada list. Perhatikan bahwa list masukan harus **minimal** punya 1 elemen.

Contoh : **throwLastElmt** [1;2;3;4;5] akan mengembalikan [1;2;3;4]

11. Deklarasikan fungsi **getLastElmt** : **'a list -> 'a** yang mengambil elemen terakhir dari sebuah list.

Contoh : **getLastElmt** [1;2;3;4;5] akan mengembalikan 5

12. Deklarasikan fungsi **removeAll** (**x**, **ls**) yang mempunyai tipe **int \* int list -> int list**. Fungsi ini akan menghapus semua kemunculan **x** di dalam list **ls**.

Contoh : **removeAll** (2, [1;2;3;2;4;2;5;2]) akan mengembalikan [1;3;4;5]

13. Deklarasikan fungsi **removeFirst** (**x**, **ls**) yang mempunyai tipe **int \* int list -> int list**. Fungsi akan menghapus kemunculan pertama **x** dalam list **ls**.

Contoh : **removeFirst** (2, [1;2;3;2;4;2;5;2]) akan mengembalikan [1;3;2;4;2;5;2]

### Inserting Element

14. Deklarasikan fungsi **addLast** (**x**, **ls**) yang bertipe **int \* int list -> int list**. Fungsi ini akan menambahkan elemen **x** pada posisi terakhir di list **ls**, jika elemen **x** belum ada pada list. Jika **x** sudah ada pada list, maka **x** tidak akan ditambahkan ke dalam list.

Contoh :

**addLast** (4, [1;2;3]) akan mengembalikan [1;2;3;4]

**addLast (4, [1;4;2;3])** akan mengembalikan **[1;4;2;3]**

15. Deklarasikan fungsi **insert (x, ls)** yang bertipe **int \* int list -> int list**. Kita asumsikan **list ls sudah terurut menaik**. Fungsi insert akan menyisipkan elemen x ke dalam list dengan tetap menjaga keterurutan dari list tersebut.

Contoh : **insert (5, [1;2;3;7;8;9])** akan mengembalikan **[1;2;3;5;7;8;9]**