

Verkefni 1: Klasinn DoubleArray

Þið eigið að búa til klasann DoubleArray og notast við aðskilda þýðingu. Klasinn á að vera ADT, þ.e. breytur og annað er private, fyrir utan þau föll sem eiga að vera sýnileg utan klasans. Klasinn á að innihalda eftirfarandi atriði:

- Kvikt fylki sem geymir double tölur
- Stærð fylkis
- Búa til færibreytulausan smið sem upphafsstillir stærð fylkisins sem 0 og fylkið sjálft sem NULL
- Smið sem tekur færibreytur fyrir stærð og gildi. Smiðurinn býr til þá stærð af kviku fylki og setur gildið í hvert einasta stak í fylkinu
- Destructor sem skilar öllu minni
- Fall sem reiknar meðaltal stakanna í fylkinu

$$\bar{x} = \frac{\sum x}{n}$$

- Fall sem reiknar staðalfrávik stakanna í fylkinu

$$\sigma = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}}$$

- operator+ sem leggur saman meðaltöl tveggja DoubleArray hluta
- operator> sem ber saman meðaltöl tveggja DoubleArray hluta
- operator<< sem prentar út fjölda staka, meðaltal og staðalfrávik
- operator>> sem tekur við fjölda staka og les svo inn hvert stak

Klasinn á að virka með eftirfarandi main falli (athugið að output þarf ekki endilega að hafa nákvæmlega sama fjölda aukastafa og sýnt er hér):

```
1  int main() {
2      DoubleArray DA1, DA2(5, 3.7);
3      cin >> DA1;
4      cout << "DA1: " << DA1 << endl;
5      cout << "DA2: " << DA2 << endl;
6
7      cin >> DA2;
8      cout << "DA1 + DA2 = " << DA1 + DA2 << endl;
9
10     if (DA1 > DA2) {
11         cout << "DA1 has the highest average!" << endl;
12     } else {
```

```
13     cout << "DA2 has the highest average!" << endl;  
14 }  
15  
16     return 0;  
17 }  
18  
19 Output:  
20     Array length: 3  
21     4 7 1  
22     DA1: n = 3, mean = 4, stddev = 2.45  
23     DA2: n = 5, mean = 3.7, stddev = 0  
24     1.4 2.7 8.9  
25     DA1 + DA2 = 8.3333  
26     DA2 has the highest average!
```

Listing 1: Example

Verkfni 2: Tengdur listi

Nú eigið þið að gera eintengdann lista af heiltölum. Notið structið Node til að geyma gögnin.

```
1 struct Node {  
2     int data;  
3     Node* next;  
4 };
```

Listing 2: Example

Skrifið forrit sem:

- Tekur inn n fjölda af tölum og býr til listann
- Prentar út listann
- Sækir aftasta stakið í listanum
- Athugar hvort tala sem notandi slær inn sé til staðar í listanum

Forritið á að hafa eftirfarandi föll:

- Fall sem býr til nýtt Node
- Fall sem bætir Node við listann
- Fall sem prentar út öll stök í listanum
- Fall sem athugar hvort einhver tala sé til í listanum
- Fall sem finnur síðasta stak í listanum
- Fall sem skilar öllu minni

```
1 Input :  
2 8  
3 1 5 3 4 6 4 2 6  
4 7  
5 Output:  
6 Data in list: 1 5 3 4 6 4 2 6  
7 Data in back: 1  
8 The number 7 is not in the list
```

Listing 3: Dæmi um input og output

```
1 Input :  
2 4  
3 1 5 6 1  
4 5  
5 Output:  
6 Data in list: 1 5 6 1  
7 Data in back: 1  
8 The number 5 is in the list
```

Listing 4: Dæmi um input og output