

# SHUFFLE

---

In de opgave gaan we een klasse maken voor gebruik in een simpel spel.

## Het spel :

De opdracht van het spel is om in zo min mogelijk stappen een reeks van getallen te ordenen.

Stel de reeks bestaat uit 6 getallen ( 1 t/m 6 in willekeurige volgorde ).

Bij elke stap kunnen we een getal opgeven tussen 1 en 6 (inclusief) .

Stel we geven 3 op.

Het resultaat is nu dat de 1<sup>e</sup> 3 cijfers in volgorde worden omgekeerd, de rest van de reeks blijft gelijk.

Bij zowel 1 als 6 wordt de gehele reeks omgekeerd.

Voorbeeld : **6 , 4 , 1 , 5 , 2 , 3** met getal 3 geeft : **1 , 4 , 6 , 5 , 2 , 3** .

De speler kan :

1. bij aanvang een lengte kiezen. ( N )  
Het programma moet nu een reeks genereren van de cijfers 1 t/m N in willekeurige volgorde.
2. De speler kan een cijfer opgeven. Tussen 1 en N (inclusief).  
Het programma zal de bestaande reeks manipuleren aan de hand van het cijfer, zoals boven beschreven.
3. Bij een geordende reeks worden geen cijfers meer geaccepteerd, en dus geen manipulaties meer uitgevoerd.
4. Op elk moment kan de score opgevraagd worden. Dit is het aantal manipulaties dat uitgevoerd is.

## Codeer instructies :

WE maken hier niet het complete spel. Alleen de 'shuffle' klasse.

Je kunt zelf ( in main in een aparte file ) het spel eenvoudig realiseren.

Je krijgt een header file **shuffle.h** .

De klasse met alle methoden moet geïmplementeerd worden in een code file **shuffle.cpp** .

1. Constructor  
met de constructor kun je een getal opgeven dat aangeeft hoe lang de reeks van getallen moet worden waarmee het spel gespeeld gaat worden.  
De maximale waarde is 20.  
Bij een ongeldige waarde ( groter dan max of kleiner dan 2) dan wordt een default waarde van 8 gebruikt!  
De constructor moet de structuur maken die de reeks getallen kan vastleggen.  
(je mag zelf bedenken hoe of wat!)
2. Destructor  
als je in de constructor geheugen alloceert dan moet je het in de destructor vrijgeven !
3. De methode manipuleer krijgt een getal opgegeven.  
Ligt dit getal buiten de range (1 .. N) dan wordt de reeks niet aangepast en een **false** waarde geretourneerd.

# SHUFFLE

---

Anders wordt de parameter gebruikt om de reeks te manipuleren en wordt **true** gereturned.

4. De methode isGeordend() geeft **true** als de reeks geordend is (1,2,...,N ). Anders **false**.
5. De methode geefScore() geeft het aantal uitgevoerde manipulaties tot dat moment.  
Merk op (punt 3) dat bij een geordende reeks de score niet meer kan veranderen.

**Gegeven is de volgende code :**

**shuffle.h:**

```
#ifndef SHUFFLE_H
#define SHUFFLE_H

#include <string>

class shuffle {
private:
    // de private data en methoden mag je aanpassen !

public:
    // deze public methoden mogen NIET worden gewijzigd !!
    // constructor met de parameter geeft aan hoe lang de reeks getallen moet zijn.
    shuffle( uint );
    // de destructor
    ~shuffle();
    // de functie die de reeks getallen manipuleert volgens de opgave.
    bool manipuleer(uint);
    // geeft true als de reeks 1 ... N is , anders false
    bool isGeordend();
    // geeft de getallen , door een komma gescheiden. v.b. : 4,5,2,1,3 (geen spaties !)
    std::string toString();
    // geef de score (= het aantal manipulaties)
    uint geefScore();
};

#endif // einde SHUFFLE_H
```

**Conditie:**

1. Voorkom memory-leaks !

**Merk op:**

1. De opdracht betreft alleen de header file en code file van de 'shuffle' klasse.  
Het 'spel' zelf hoeft niet gerealiseerd te worden.

# SHUFFLE

---

Je kunt een testprogramma maken met daarin (delen van) het spel. Je mag hier test-code opnemen voor de gemaakte klasse / methoden, geheel naar eigen inzicht.

2. De 'signatuur' van de public shuffle-methoden mag niet worden aangepast.  
Wijzigingen geven fouten in de test.

## Upload instructies:

Omdat je de gegeven header file mag aanpassen, moet je deze wel uploaden. Anders zal de compiler jouw code niet kunnen verwerken.

Upload de volgende 2 bestanden:

1. **shuffle.cpp**
2. **shuffle.h**