

## FEJLETT PROGRAMOZÁSI NYELVEK, 2009

### 9. GYAKORLAT – Tárolók és algoritmusok

#### Cél:

- A `string` és a `stringstream` használata
- Permutációk

#### I. A `stringstream`

Bontunk fel egy karakterláncot egységekre!

- `s`: a bemenő karakterlánc, ezt kell felbontani
- `delim`: ezzel a karakterrel vannak elválasztva az egységek
- `elems`: ide teszi az egységeket

```
vector<string> &mysplit(const string &s, char delim, vector<string> &elems) {  
    stringstream ss(s);  
    //Kiegészíteni  
}
```

#### II. Általánosítsuk az I. feladatot és készítsünk egy `StringTokenizer` osztályt!

```
#ifndef STRING_TK_  
#define STRING_TK_  
  
#include <string>  
#include <vector>  
#include <fstream>  
  
using namespace std;  
  
class StringTokenizer  
{  
    private:  
        vector<string> v;  
        string delims;  
        int poz;  
  
    public:  
        StringTokenizer(string line="", string delims=" ");  
        int numberOfTokens();  
        string nextToken();  
        bool hasNext();  
};  
#endif
```

#### Segítség:

Használjuk fel a `string` osztály metódusait.

Ld. *nyomtatott jegyzet*: 129-138 o.

Ld. *elektronikus jegyzet*: 120 -128 o.

## FEJLETT PROGRAMOZÁSI NYELVEK, 2009

### 9. GYAKORLAT – Tárolók és algoritmusok

#### 3. Permutációk I.

Olvasson be a standard bemenetről egy szóhalmazt. Írassa ki a szóhalmaz összes permutációját.

Megoldási vázlat:

- beolvassuk a szavakat egy sorozat típusú tárolóba
- rendezzük a szavakat a sort algoritmussal azért, hogy előállítsuk lexikografikus sorrendben a legkisebb permutációt
- kiíratjuk a legkisebb permutációt
- felhasználva a `next_permutation` algoritmust előállítjuk és kiíratjuk az összes permutációt

#### 4. Permutációk II.

Az A, E, K, M, T betűkből, minden betűt egyszer felhasználva, felírjuk az összes értelmes és értelmetlen szót, ábécé sorrendben.

- Készítsen programot, amely egy adott szóra eldönti, hogy hányadik ábécé sorrendben.
- Hányadik helyen áll a MATEK szó?

#### 5. EXTRA

Készítsen programot, amely adott helyes zárójelezésre eldönti, hogy hányadik lexikografikus sorrendben. A ( kisebb lexikografikus sorrendben, mint a ) ). Készítsen gyors algoritmust!

Példa n=2:

1. ( ( ) )
2. ( ) ( )

Bemenet	Kimenet
( ) ( )	2
(( ( ) ))	1
(( ) ( ) )	2
(( ) ) ( )	3
( ) (( ) )	4
( ) ( ) ( )	5