**Storage Classes**

**auto, register, extern and static** defines the variables storage class, storage duration, scope and linkage specificasions.

scope is where the identifier can be referenced in a program.

linkage (security-hiding variable)determines for a multiple-source-file program whether the identifier is known only in the current source file or in any source file with proper declarations.

**Auto**: only available in its block (default storage class)

**extern and static**: available in all the program-accessible only in its block

**Scope**

function scope(Labels are the only ones), file scope(declared outside all functions), block scope(defined inside a block), and function-prototype scope(only prototype parameters)

**const qualifier**

const variables can not be modified (not even the pass by value)

if used like “const int\*”: value can not be modified

if used like “int \* const”:pointer can not be modified

if used like “const int \* const”:neither pointer or value can be modified

**bubble sort**

for(i=0;i<size-1;i++)

for(j=0;j<size-1;j++)

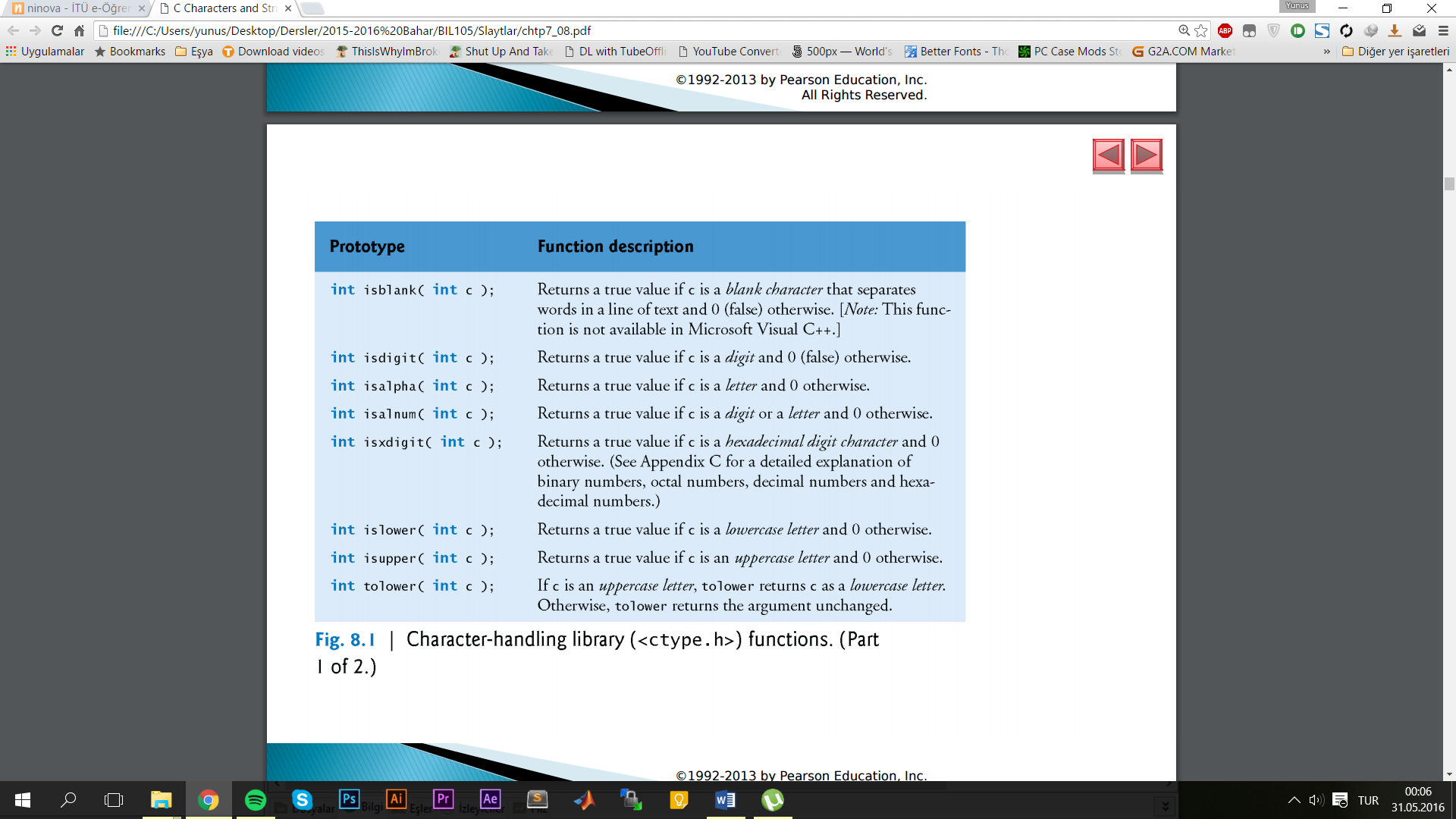
if(array[j]>array[j+1])

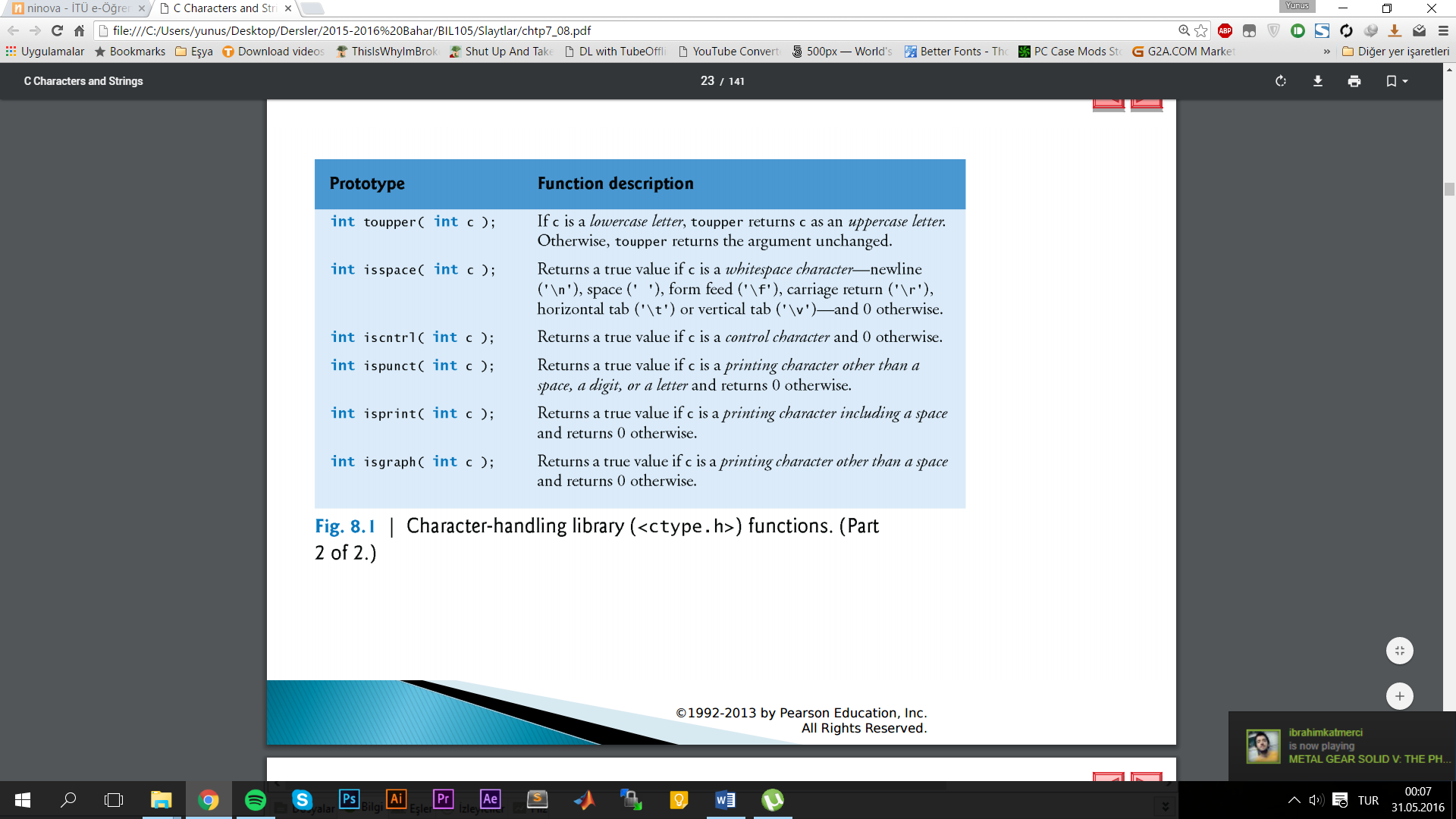
swap(array[j],array[j+1])

**note**

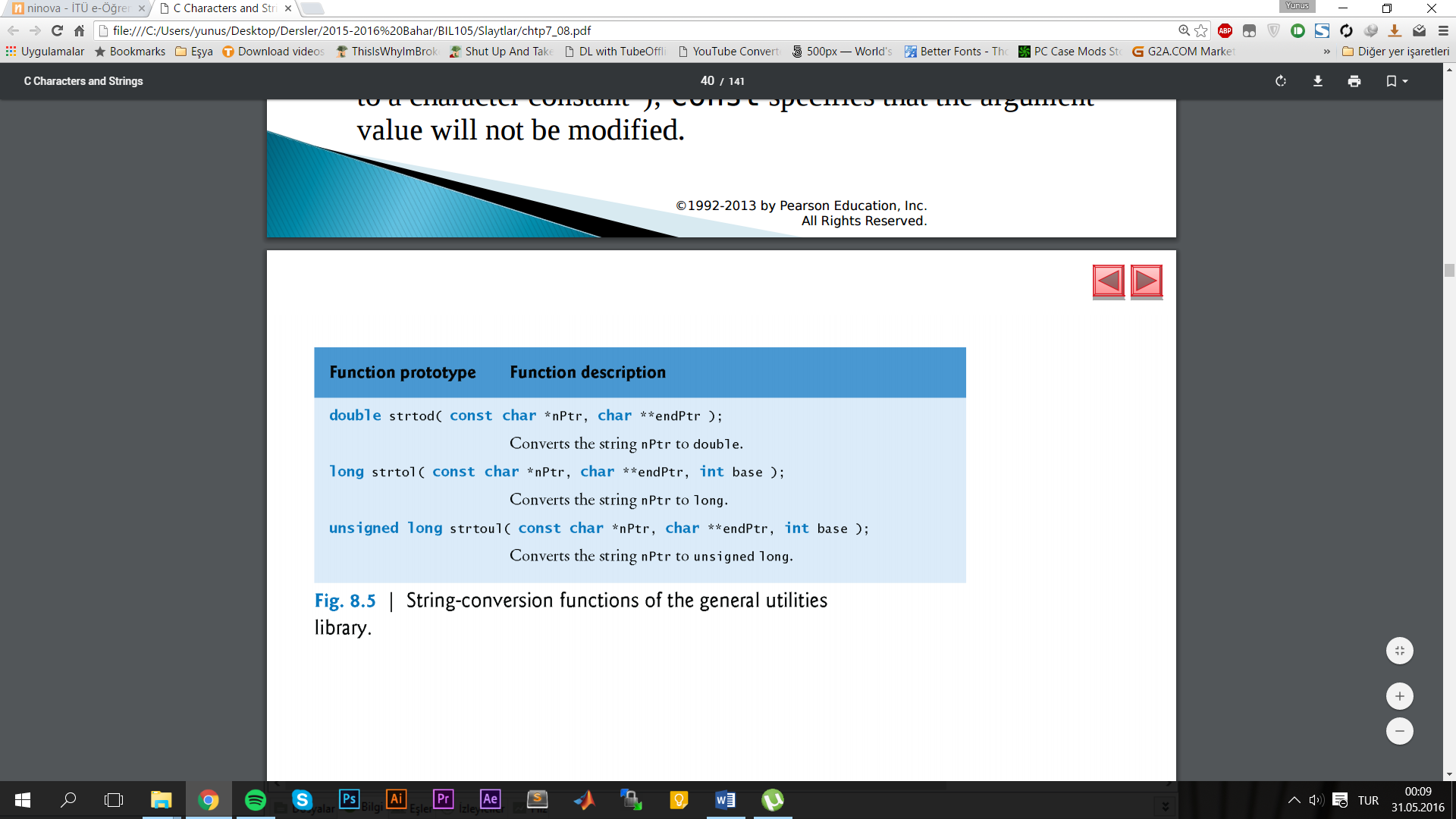
pointer aritmetik işlemleri pointer’ın tipindeki değerin boyut kadar olur (sizeof a gerek yok)

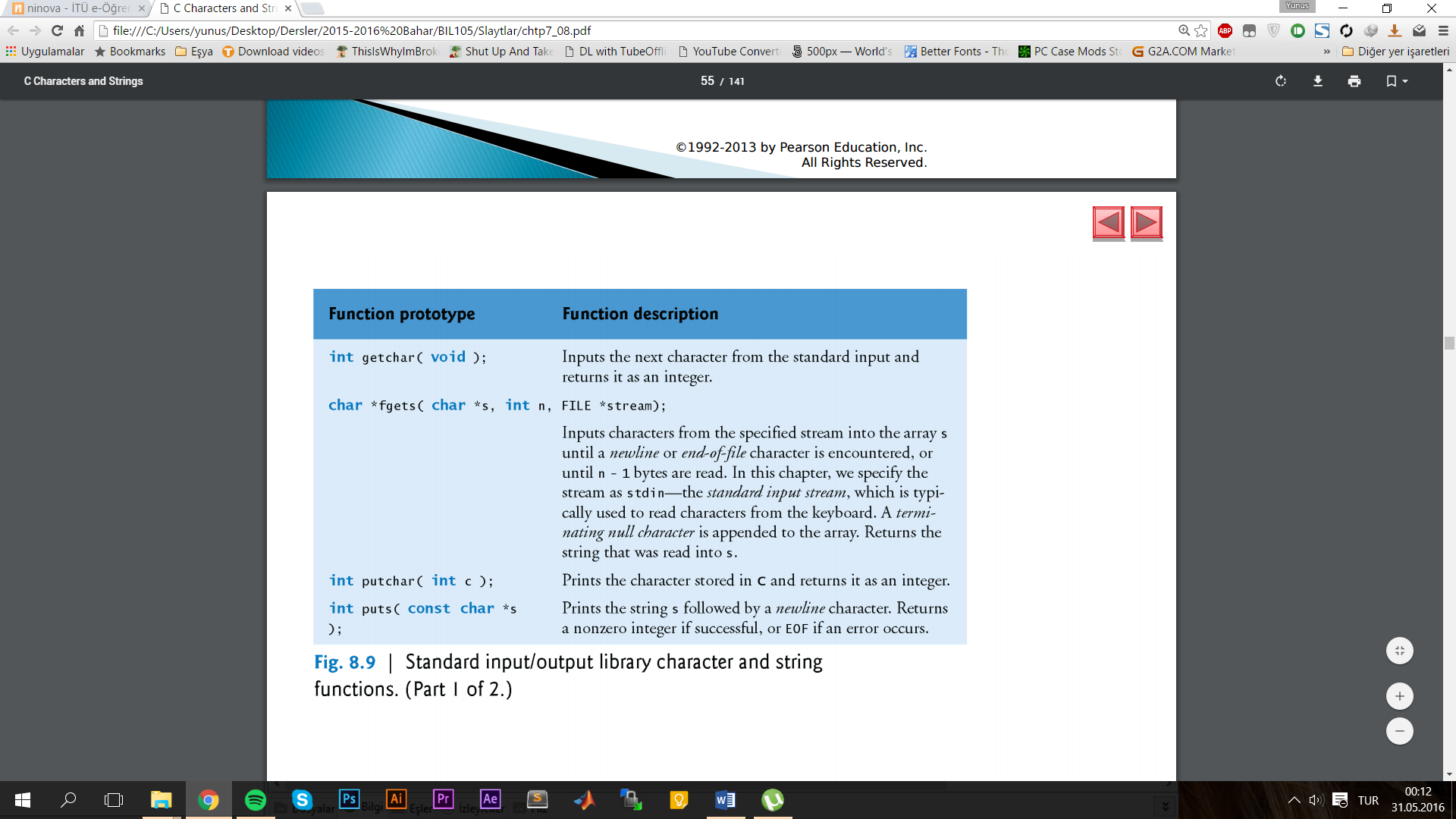
**Char-String Libraries**

**<ctype.h>**

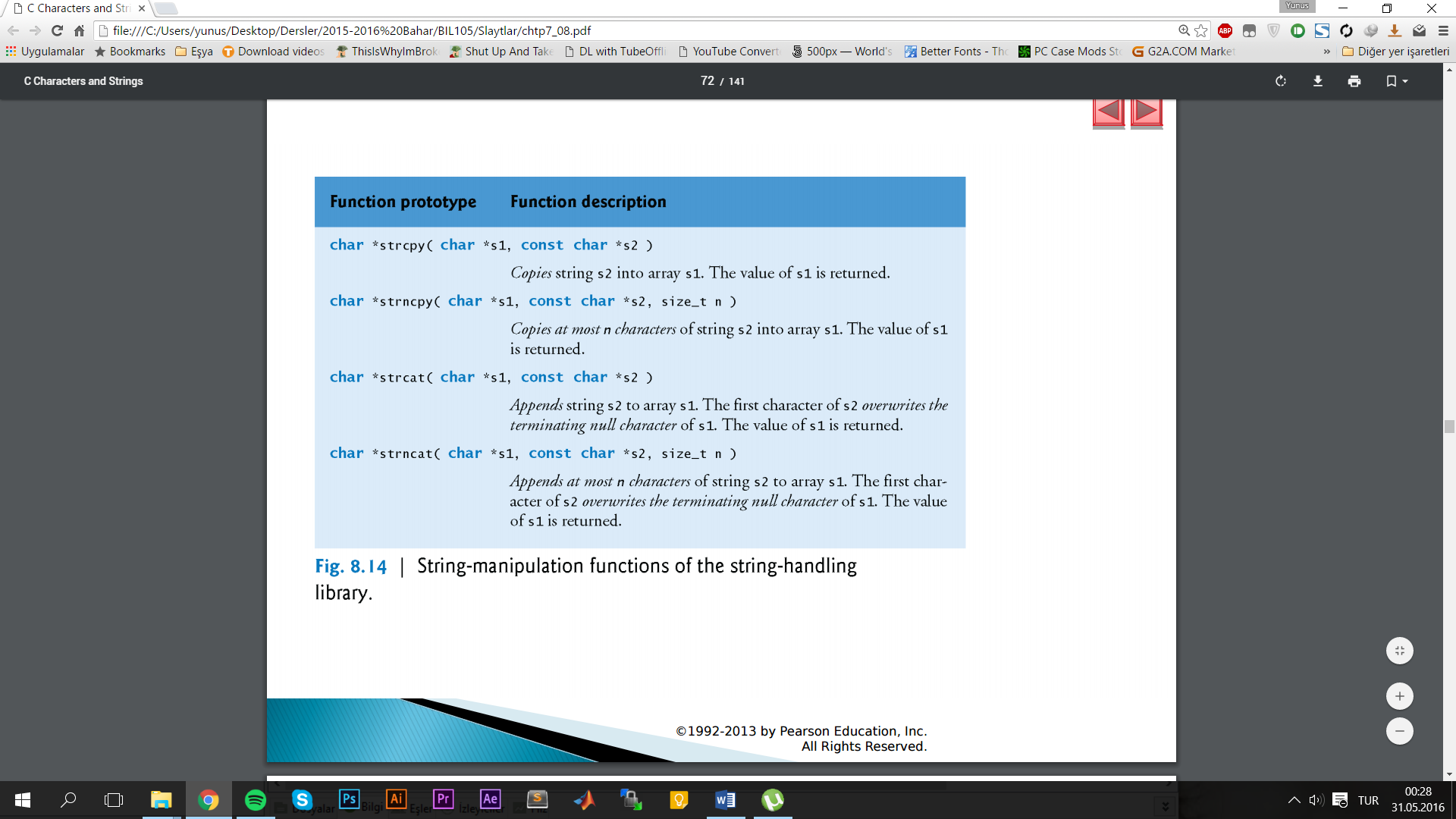


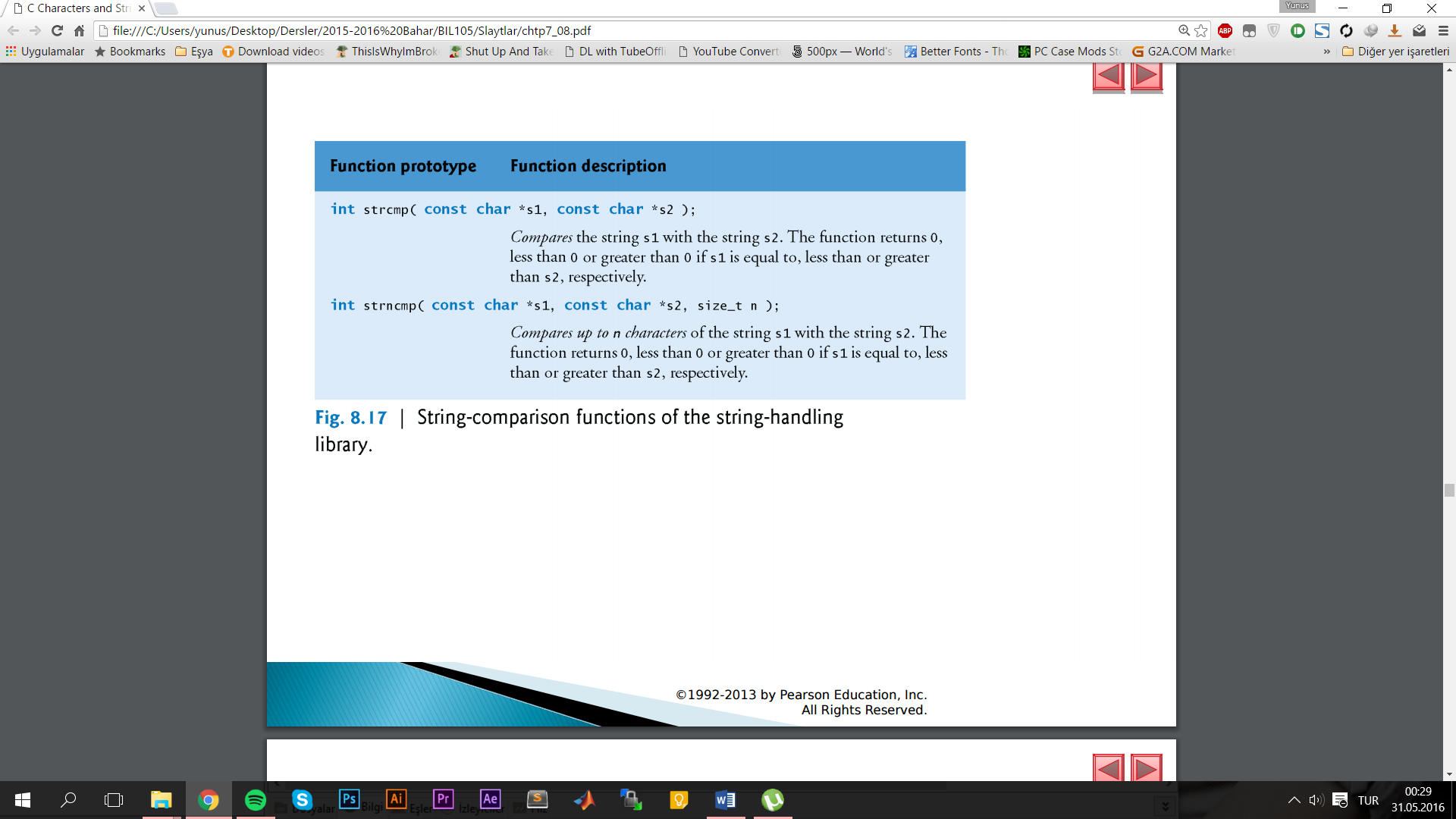
**<stdlib.h>**

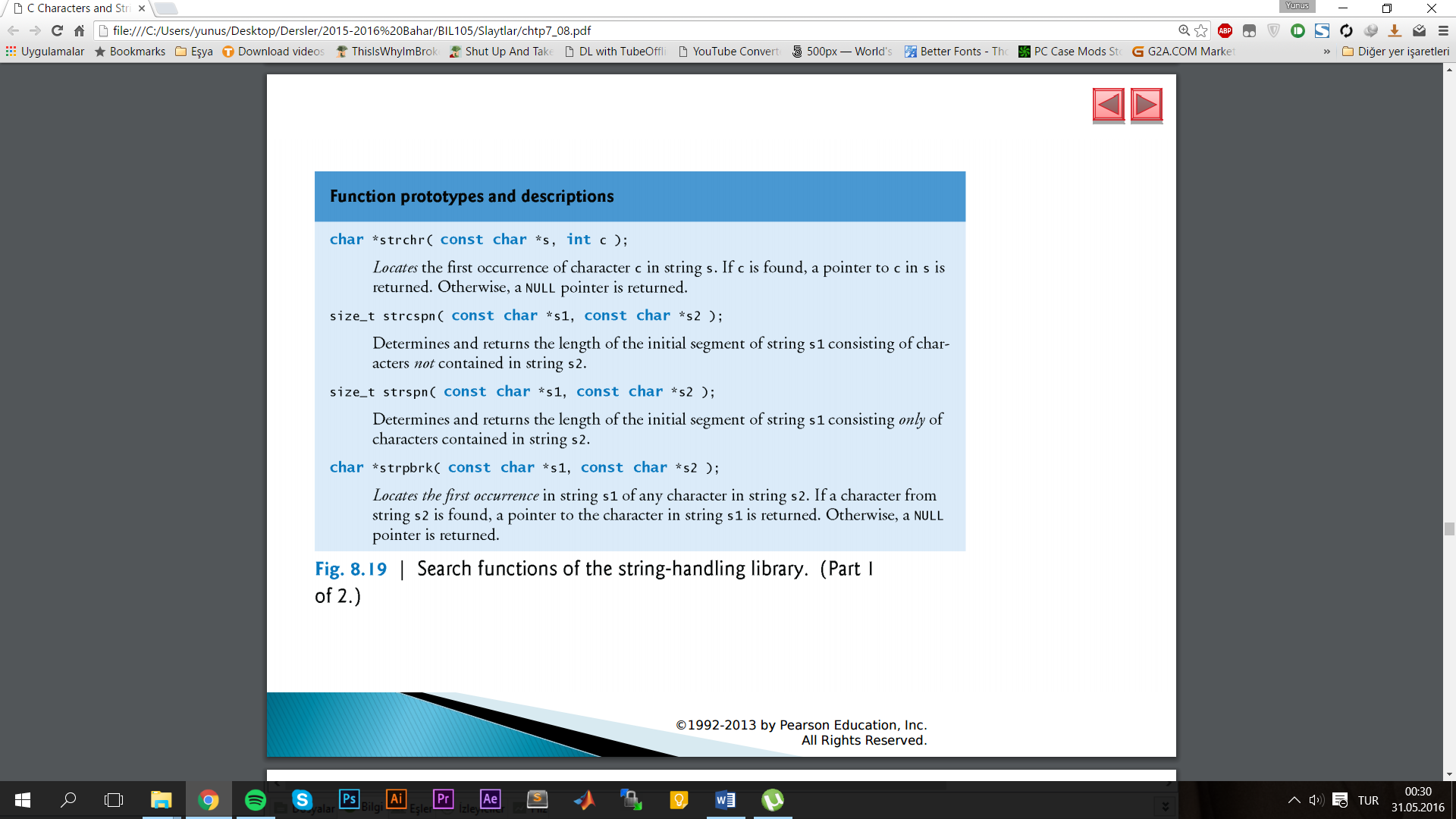


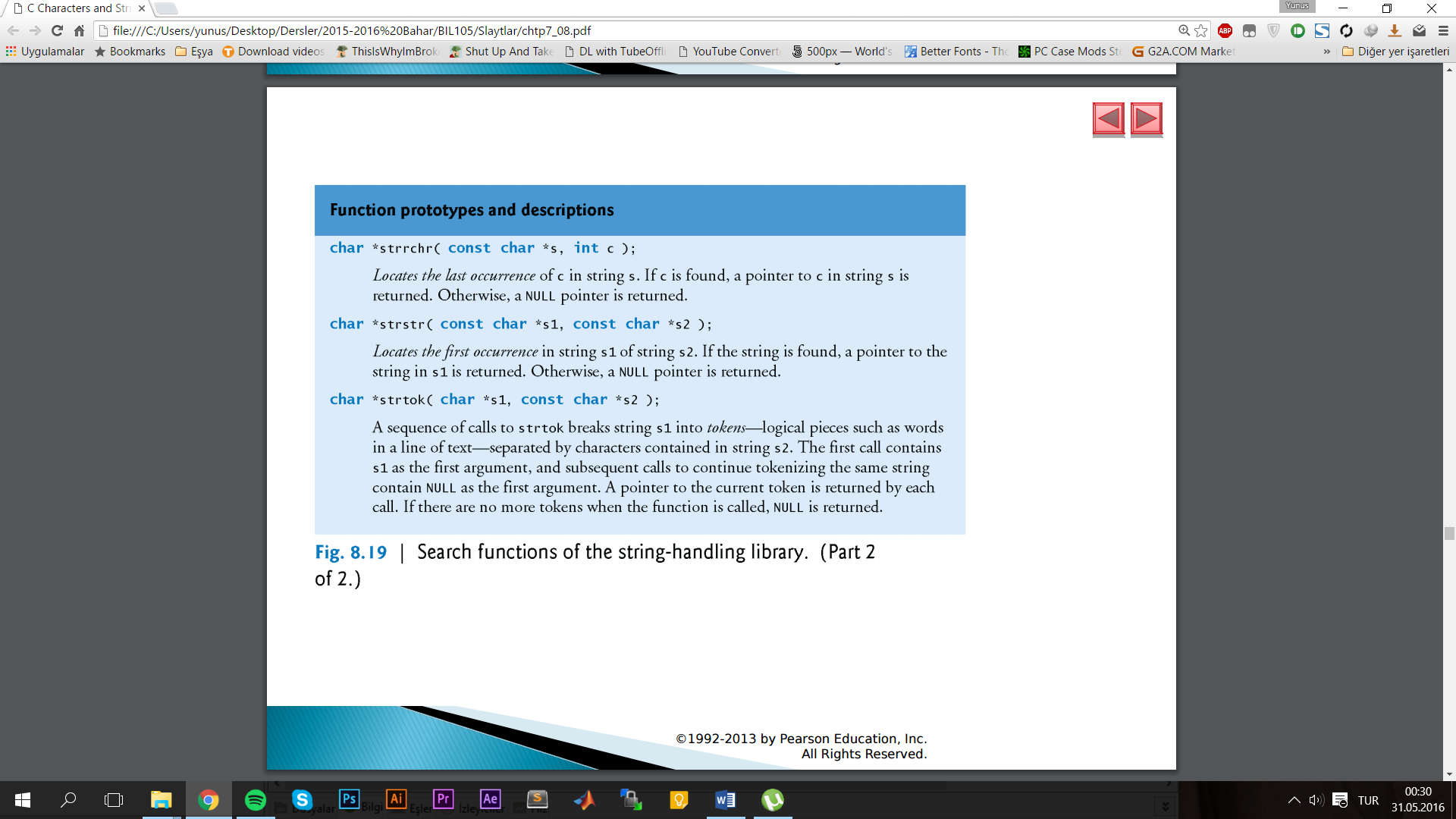
**<stdio.h>**

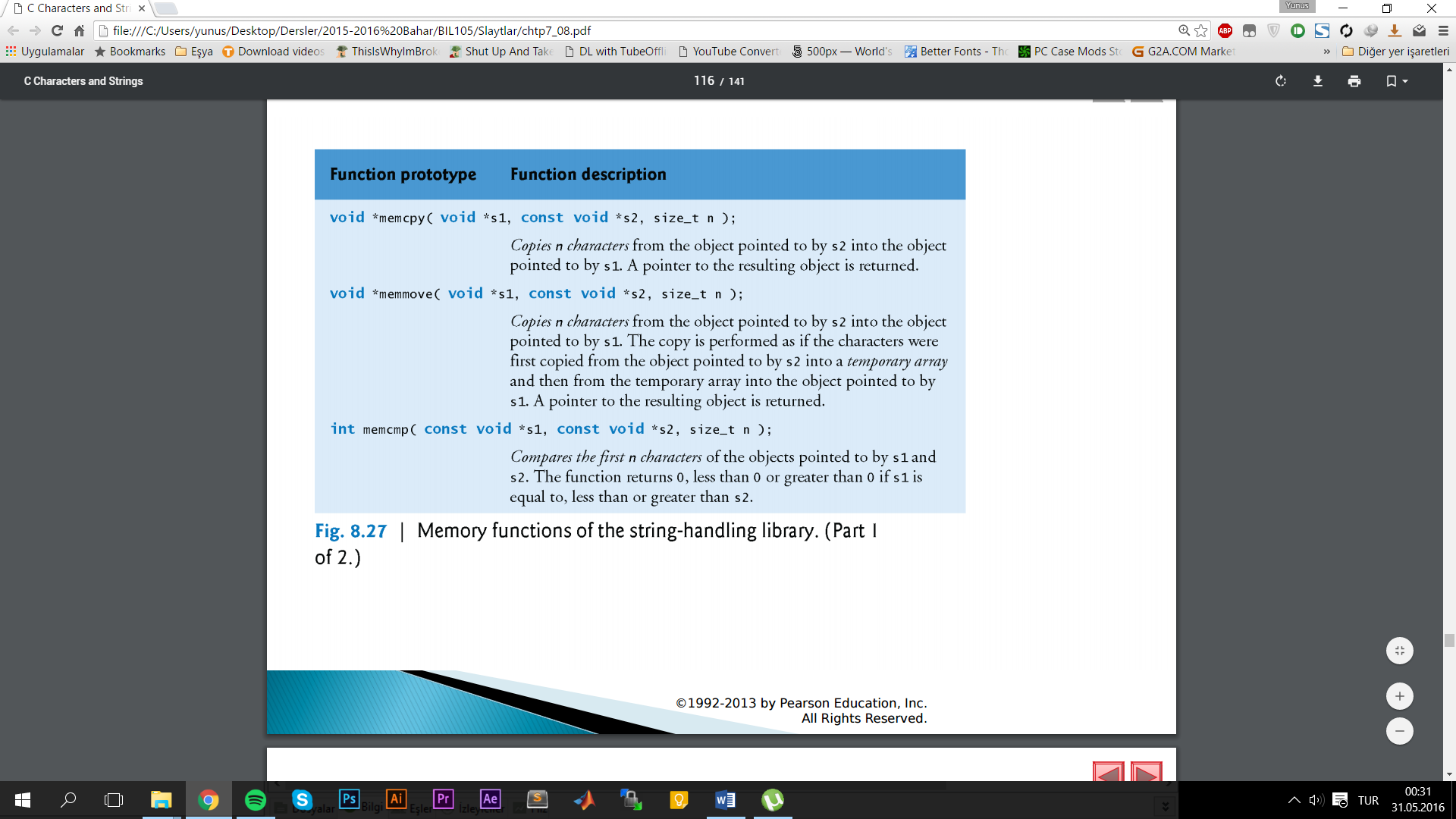
**<string.h>**

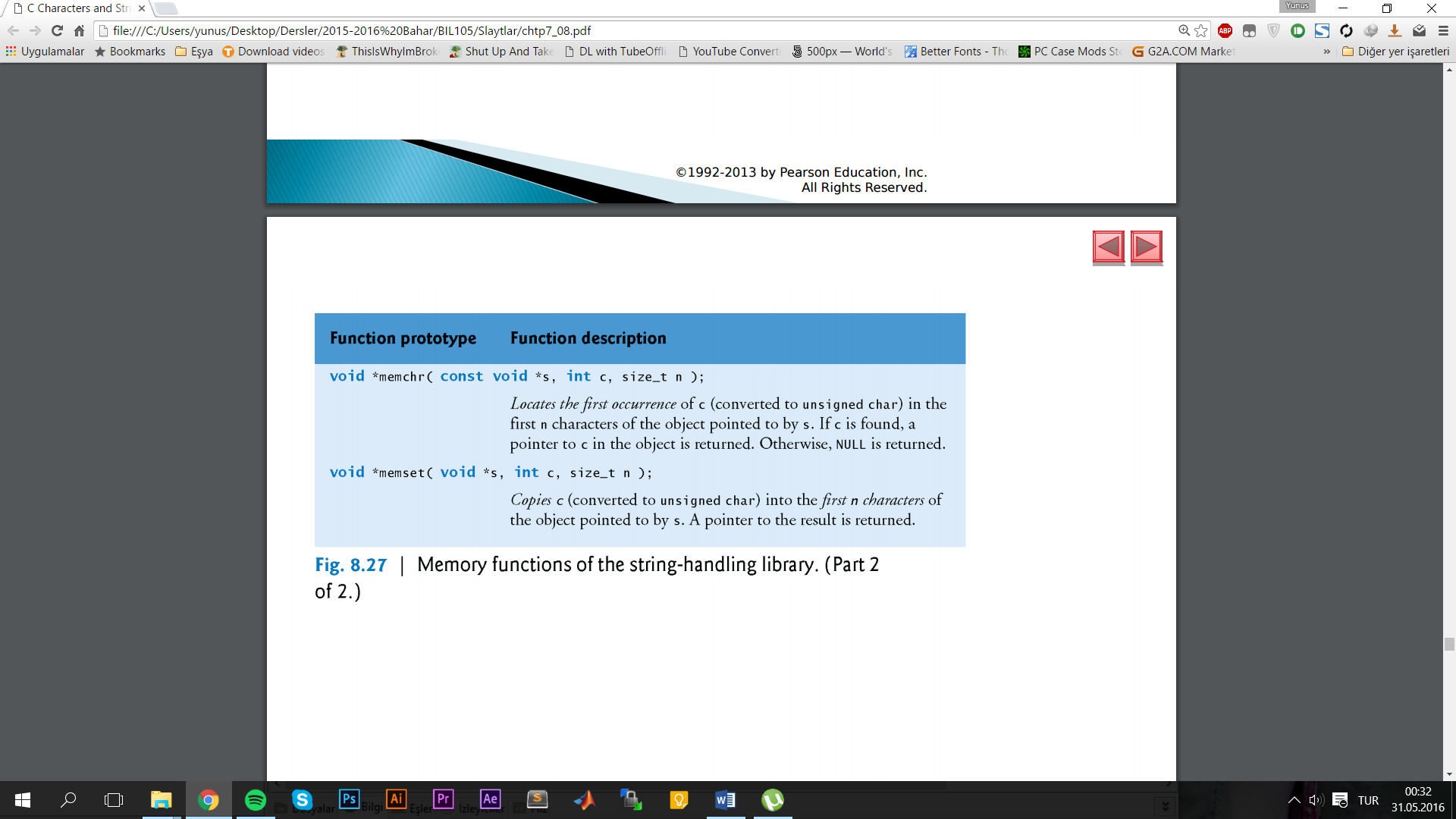


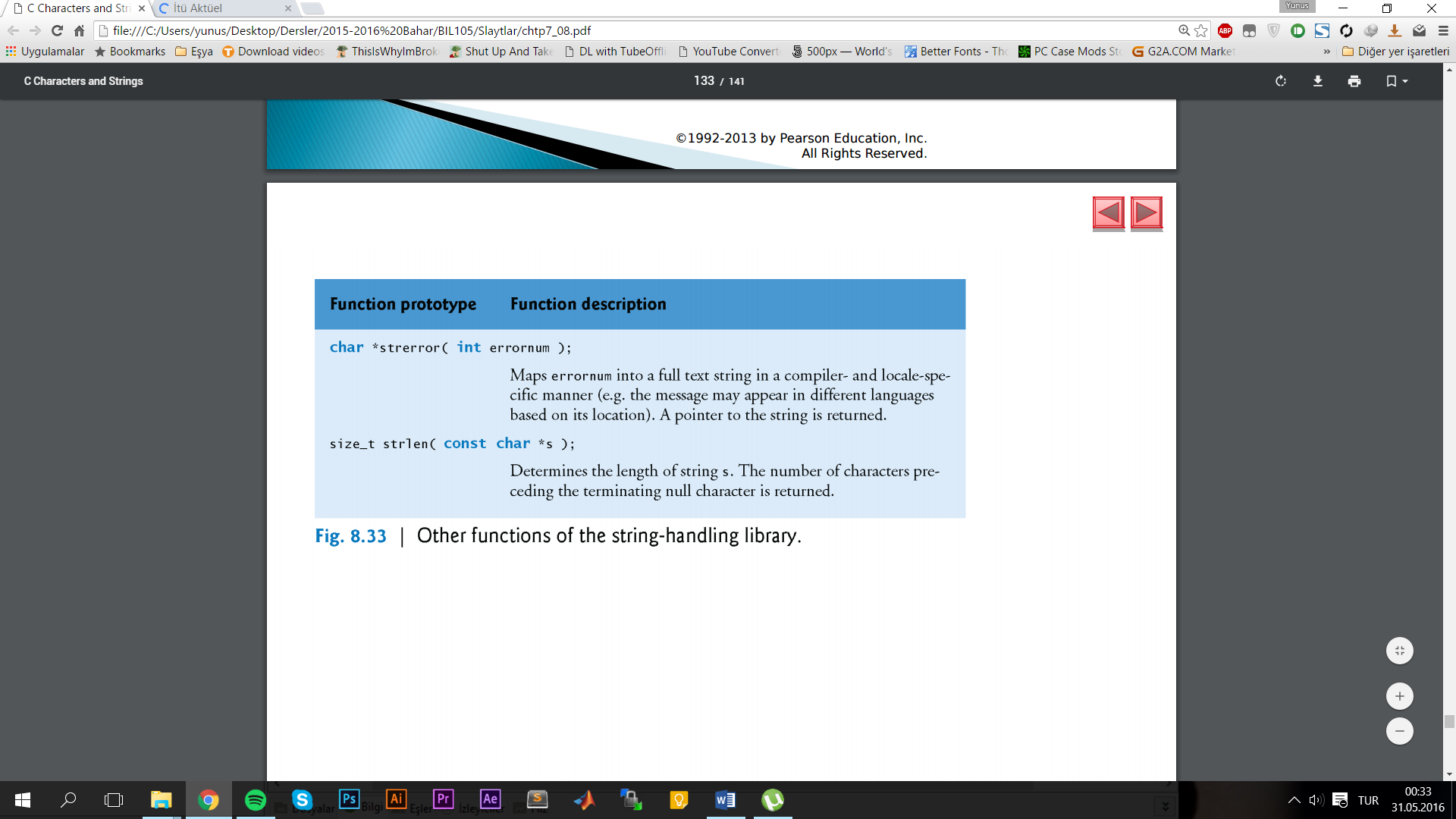




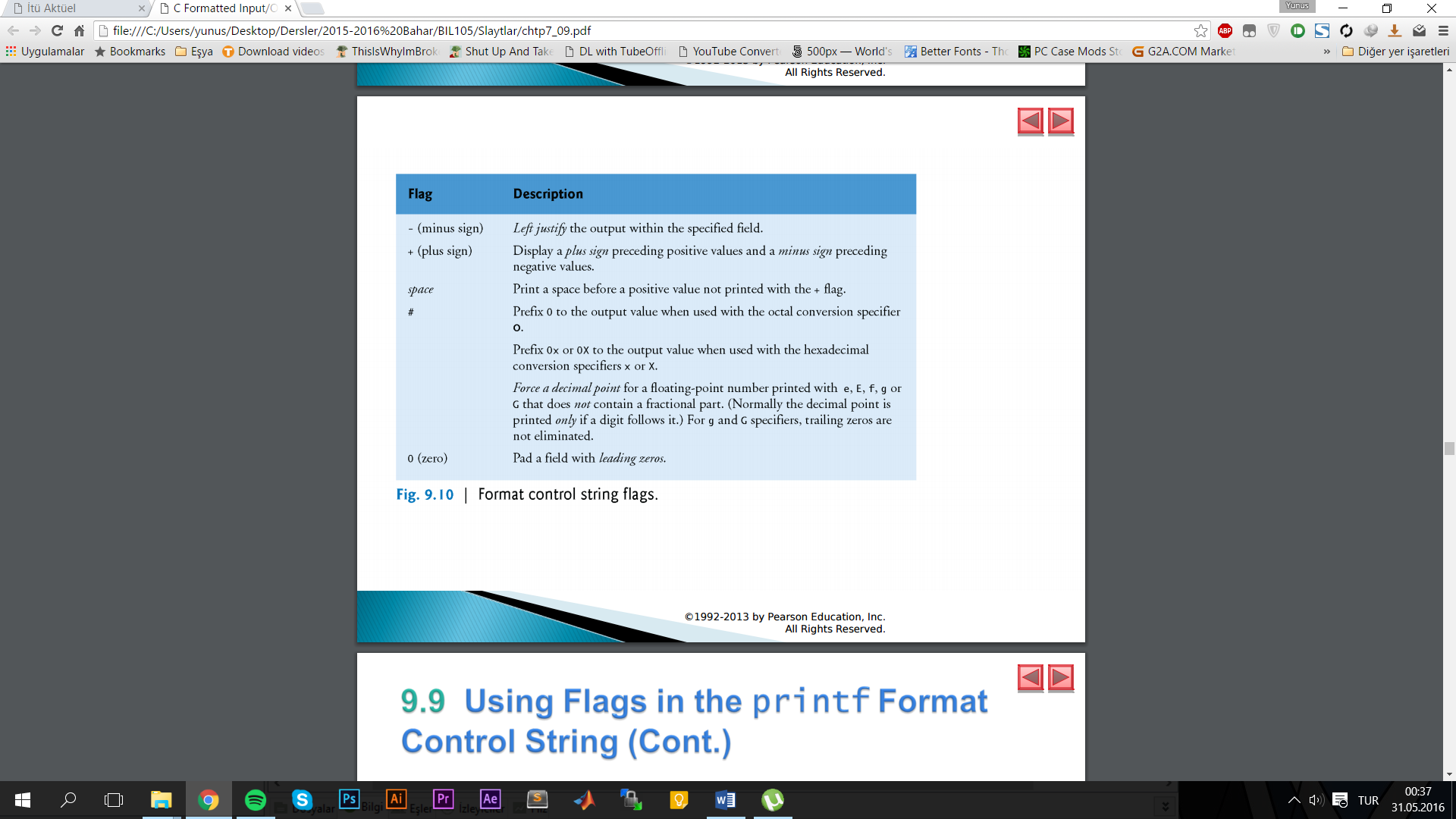








**Printf flags**



**Union-Structure-Enum**

**Structure**

cardPtr->suit is equivalent to (\*cardPtr).suit

typedef struct { char \*face; char \*suit; } Card;

Card deck[ 52 ];

Union:like structure but shares same space(birinden birini kullanabiliyosun)(bitleri okuma şekli değişiyor)