C Programming:

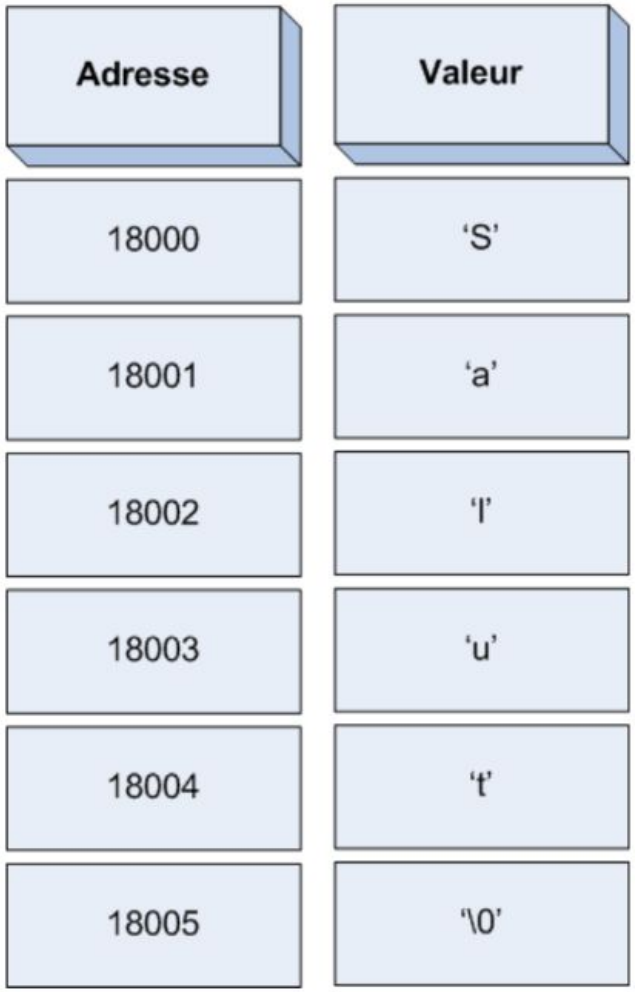
Cours 11:

Strings

Dr. Chénche

Strings (Chaine de characters) ?

- A string is nothing more than an array of type char.
- A string must always contain a special character at the end of the sequence, called the *null terminator*.
- This character is written as '\0'.



Strings

Declaration:

Syntax :

```
char name[size];
```

Where :

- **char** → the type (each element is a character)
- **name** → the identifier (variable name)
- **size** → (value) the total number of characters the array can hold **including the null terminator \0**

Strings

Declaration:

Examples :

- char ch [6]=‘’Hello”;

H	e	l	l	o	\0

Index	ch[0]	ch[1]	ch[2]	ch[3]	ch[4]	ch[5]
Value	h	e	l	l	o	\0

Strings

Initialization and display of string :

```c
int main()
{
    char chaine[6];

    chaine[0] = 'S';
    chaine[1] = 'a';
    chaine[2] = 'l';
    chaine[3] = 'u';
    chaine[4] = 't';
    chaine[5] = '\0';
    // Affichage de la chaine grace au %s du printf
    printf("%s", chaine);
    return 0;
}
```

Strings

Declaration:

Examples :

- char ch [6]; ✓
- char ch [6] = {'H', 'e', 'l', 'l', 'o', '\0'}; ✓
- char ch [6] = "Hello"; ✓
- char ch [6]; ch= "Hello"; ✗
- ch [4] = "x"; ✗
- ch [4] = 'x'; ✓

Strings

Reading a string :

1. Using scanf :

```
char ch[20];
scanf("%s", ch);
```

Limit :

scanf **cannot read spaces (blanks)** in a string (eg. "Hello World" → only "Hello" is read).

Strings

Reading a string :

2. Using fgets :

char ch[20];

Syntax :

fgets(String_name, size, stdin);

fgets can read **spaces(blanks) and entire lines**.

Strings

Functions :

Function	Purpose	Example	Header
fgets	Reads a line of text from input (stdin or file)	fgets(str, size, stdin);	<stdio.h>
getchar	Reads a single character from stdin	c = getchar();	<stdio.h>
sizeof	Returns the size in bytes of a variable or array	sizeof(ch)	
strlen	Returns the length of a string (number of characters without counting \0)	strlen(str);	<string.h>