

2. Ariketa: Erakusleak, Arrayak, Karaktere Kateak

Ejercicio 2: Apuntadores, Arrays, Cadenas de Caracteres

Helburua:

Ikasleak Erakusle , Array eta Karaktere Kateen kontzeptuak ikastea.

Teoria:

- Erakusleak, Arrayak eta Karaktere kateak

Kontsultatzeko bibliografia:

Klaseko apunteak

Objetivos:

Que el alumno aprenda los conceptos de Apuntadores, Arrays y Cadenas de Caracteres .

Teoria:

- Apuntadores, Arrays, Cadenas de Caracteres

Bibliografía de consulta:

Apuntes de clase

El Lenguaje de Programación C

Brian W. Kernighan, Dennis M. Ritchie

C/C++ Curso de Programación

Fco. Javier Ceballos

Enuntziatuak / Enunciados

1. Idatzi, gehienez 1024 karaktere izango dituen katea tekladutik irakurri ondoren, hau kapikua den edo ez esango digun programa. Bukatzeko, karaktere katea zuzenean eta alderantziz pantailaratu.
2. Idatzi, karaktere kate bat tekladutik irakurri ondoren, kate honen karaktere kopurua pantailan idatziko digun programa. Ezingo da **strlen** funtzioa erabili. Zeuk idatzi beharko duzu honen lana egiten duen funtzioa, **nire_strlen**.
3. Idatzi, 100 zenbaki erreal baino gutxiago irakurri ondoren, berauek txikitik handira ordenatzen dituen programa. Erabili array bakar bat. Behin array-a ordenatutakoan, idatzi emaitza pantailan.
4. *Karaktere kate bi* irakurri eta bietatik alfabetikoki *txikiena* dena *idatzi*. Berdinak badira zirkunstantzia hau adierazi. **strcmp** funtzioa erabili beharrean funtzio baliokide bat gauzatu, **nire_strcmp**.
5. *Karaktere kate bi* irakurri eta biak kateatuz string berria idatzi. **strcat** funtzioa erabili beharrean funtzio baliokide bat gauzatu, **nire_strcat**.
6. *Karaktere kate bi* irakurritik batek bestea *barne* badauka, pantailaratu irakurritako katea eta lehenengo agerpen karakteretik aurrerakoa. **strstr** funtzioa erabili beharrean funtzio baliokide bat gauzatu, **nire_strstr**.

2. Ariketa: Erakusleak, Arrayak, Karaktere Kateak

Ejercicio 2: Apuntadores, Arrays, Cadenas de Caracteres

`size_t strlen(const char *s);`

Description:

Calculates the length of a string.

strlen calculates the length of s.

Return Value:

strlen returns the number of characters in s, not counting the null-terminating character.

size_t → Defined in: `stddef.h`

Description:

Type which is an unsigned integer returned by the sizeof operator.

`int strcmp(const char *s1, const char *s2);`

Description:

Compares one string to another.

strcmp performs an unsigned comparison of s1 to s2, starting with the first character in each string and continuing with subsequent characters until the corresponding characters differ or until the end of the strings is reached.

Return Value:

If s1 is... strcmp returns a value that is...

less than s2 < 0

the same as s2 == 0

greater than s2 > 0

`char * strcat(char *dest, const char *src);`

Description:

Appends one string to another.

strcat appends a copy of **src** to the end of **dest**. The length of the resulting string is `strlen(dest) + strlen(src)`.

Return Value:

strcat returns a pointer to the concatenated strings.

`char * strstr(const char *s1, const char *s2);`

Description:

Scans a string for the occurrence of a given substring.

strstr scans s1 for the first occurrence of the substring s2.

Return Value:

strstr returns a pointer to the element in s1, where s2 begins (points to s2 in s1). If s2 does not occur in s1, strstr returns null.