

ALUMNO:

EPS – Escuela Politécnica Superior

Asignatura: Programación I (Grado de Ingeniería Informática – 1^{er} curso)

Curso: 2020/2021

Examen: Final

Fecha: 21-06-2021

Semestre: 1º

Convocatoria: Extraordinaria

Parte Practica [10 puntos, 70% nota final]

Para esta parte del examen se deben entregar 3 archivo a través del Campus Virtual: ejercicio1.cpp, ejercicio2.cpp y ejercicio3.cpp

Ejercicio 1: [3 puntos]

Implementar una función que reciba como parámetros de entrada dos `std::string` y devuelva un `std::vector<char>` con los caracteres que tengan en común las dos variables `std::string` de entrada sin que estén repetidos.

Puedes implementar tantas funciones adicionales como necesites.

Ejemplo:

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
<code>std::string in1</code>	u	n		p	e	r	r	o		c	o	m	e

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
<code>std::string in2</code>	u	n	a		p	u	e	r	t	a

	[0]	[1]	[2]	[3]	[4]	[5]
<code>std::vector<char> out</code>	u	n		p	e	r

Ejercicio 2: [3 puntos]

Implementar una función que reciba como parámetros de entrada dos `std::string` y devuelva un `std::vector<string>` con las palabras que tengas en común.

Puedes implementar tantas funciones adicionales como necesites.

Ejemplo:

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
<code>std::string in1</code>	p	e	p	e		t	o	m	a		t	e

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
<code>std::string in2</code>	e	l		t	o	m	a		u	n		t	e

	[0]	[1]
<code>std::vector<string> out</code>	toma	te

Ejercicio 3: [3 puntos]

Implementar una función que reciba como parámetro de entrada un `std::string`, de tal forma que muestre por terminal cuantos signos de puntuación tiene de cada tipo de este grupo que os indicamos a continuación ('.', ';', ':', '!', '?'), así como su porcentaje respecto al número total de caracteres del `std::string` de entrada.

Adicionalmente dicha función debe devolver el carácter que haya sido más repetido, así como su %.

Enlace de ayuda: <https://es.stackoverflow.com/questions/112394/c-problema-al-dividir-2-3>

Ejemplo:

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
<code>std::string in1</code>	H	o	l	a	!	T	o	m	a	s		u	n

[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]
c	a	f	e	?	S	o	l	o	?	A	q	u	i		t

[29]	[30]	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]	[26]	[27]	[28]
i	e	n	e	s	.									

Carácter más repetido: ?

Porcentaje: $(2/35) \cdot 100 = 5.71428$

Ejercicio 4: [1 punto]

Prueba todas las funciones de los tres apartados anteriores en el main de cada una.

Rúbrica de evaluación

El programa no compila o no se asemeja a lo que se pide	0%
El programa se aproxima suficientemente a lo pedido	60%
El programa funciona correctamente y las estrategias y elementos de código elegidos son adecuados.	100%