Ejercicio obligatorio 2

Fecha de entrega: Domingo 14 de abril

Introducción

Un programa utiliza un protocolo de texto donde comunica sus mensajes con cadenas de caracteres.

Estas cadenas constan de varios campos, separados por espacios. Ahora bien, además del carácter espacio en estas cadenas juega un rol especial el carácter dos puntos. Si el mismo está al comienzo de la cadena debe ser ignorado. Luego de ese primer carácter (que puede o no estar) la próxima vez que el carácter dos puntos aparezca será después de un espacio y significa que todo lo que sigue a continuación es un único campo, sin importar si tiene dentro espacios o no.

Algunos ejemplos de cadenas y sus campos:

```
"hola que tal te va" => {"hola", "que", "tal", "te", "va"}
":hola que tal te va" => {"hola", "que", "tal", "te", "va"}
"hola que :tal te va" => {"hola", "que", "tal te va"}
":hola que tal :te va" => {"hola", "que", "tal", "te va"}
":hola que :tal te :va" => {"hola", "que", "tal te :va"}
```

En el primer ejemplo hay 5 campos, lo mismo en el segundo, el que le sigue tiene 3, 4 y el último 3.

Se puede asumir que no hay múltiples espacios consecutivos.

Trabajo

Comparar cadenas

Implementar una función `bool son_iguales(const char a[], const char b[]);` que devuelva `true` si las cadenas `a` y `b` son iguales, de forma insensible a las mayúsculas. Es decir, la cadena `"HoLa"` es idéntica a la cadena `"hOlA"`.

Cantidad de campos

Implementar una función `size_t cantidad_campos(const char s[]);` que dada una cadena `s` devuelva la cantidad de campos que contiene.

Lectura de línea

Implementar una función `bool leer_linea(char s[]);` que lea de `stdin` caracteres de a uno por vez hasta encontrar el `'\n'` y los guarde en la cadena `s`. Al encontrar el carácter `'\n'` el mismo debe ser descartado y debe terminar la lectura (y la cadena). En caso de falla en la lectura de caracteres debe devolverse `false`, `true` en caso contrario.

Extraer campo

Implementar una función `bool extraer_campo(char campo[], const char s[], size_t c);` que extraiga el campo `c` de la cadena `s` y lo almacene en `campo`. Debe devolver `true` si puede realizar la tarea.

Comparar cadenas

Implementar una función `size_t buscar_cadena(const char s[], char opciones[][MAX_CADENA], size_t n_opciones);` que busque si la cadena `s` es alguna de las `n_opciones` cadenas del vector de cadenas `opciones` (de forma insensible a las mayúsculas). La función debe devolver el índice de `opciones` en el cual se encontró la ocurrencia con `s`. En cambio de que `s` no sea ninguna de las cadenas de `opciones` entonces se debe devolver `n_opciones`.

Aplicación

Se pide implementar un programa que lea líneas de `stdin` de a una por vez hasta que se termine la entrada.

Para cada línea:

- Si el primer campo es `"PING"` la aplicación debe imprimir `"PONG"` seguido por los demás campos que se hayan envíado, ejemplo: `"ping hola :estas ahi?"` debería imprimir `"PONG hola estas ahi?"`
- Si el primer campo es `"MENSAJE"` el segundo campo es un destinatario y el tercero es un mensaje. Se debe imprimir el nombre del destinatario seguido del mensaje, ejemplo: `":mensaje Juan :Como estas?"` debería imprimir `"Juan: Como estas?"`
- Si el primer campo es `"SALUDAR"` el segundo campo es un destinatario. Se debe saludar al destinatario, ejemplo: `"saludar :Juan Carlos"` debería imprimir `"Hola Juan Carlos!"`.
- Si el primer campo es `"NOMBRE"` el segundo debería ser un nombre y el tercero un apellido, ejemplo: `":nombre Juan :Perez Garcia"` debería responder `"Bienvenido Perez Garcia, Juan"`.
- Si el primer campo es `"SALIR"` el programa debería terminar.

Si el primer campo es inválido o no se recibe la cantidad de campos esperados el programa debe imprimir un mensaje de error y seguir.

El programa sólo terminará si se recibe `"SALIR"` como primer campo o si se termina la entrada.

Entrega

Deberá entregarse el código fuente del programa desarrollado.

El programa debe compilar correctamente con los flags:

```
-Wall -Werror -std=c99 -pedantic
```

y validar los ejemplos dados.

La entrega se realiza a través del sistema de entregas.

El ejercicio es de entrega individual.