

Apellido y Nombres	Legajo	Calificación

1. Implemente una función que contenga el algoritmo de cifrado **ROT13**.

El algoritmo consiste en sustituir cada letra por una que se encuentra trece (13) posiciones por delante o por detras según corresponda.

Por ejemplo la A se reemplaza por la N, la B por la O y así sucesivamente. Para las últimas trece letras la secuencia se invierte.

Se debe mostrar el mensaje original, el mensaje codificado y la cantidad de caracteres que fueron sustituidos. También se debe utilizar la menor cantidad de memoria posible y el mensaje no puede exeder 30 caracteres.

A continuación se muestra la tabla de las equivalencias entre las letras.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

a	b	c	d	e	f	g	h	i	j	k	l	m
n	o	p	q	r	s	t	u	v	w	x	y	z

Ejemplo:

Texto sin cifrar	H	o	L	a		m	u	N	d	O
Texto cifrado	U	b	Y	n		z	h	A	q	B

Prototipo de la función:

`int rot_13 (char *dataPtr , char * dataPtrCodificada)`

Donde:

- **dataPtr:** es el puntero al mensaje a cifrar (string).
- **dataPtrCodificada:** es el puntero al mensaje cifrado (string) donde se colocará el mensaje cifrado.

Devuelve:

- **-1** Si el mensaje contiene un caracter **distinto** de una **letra** o un **espacio**.
- Un **número positivo** indicando la cantidad de caracteres convertidos sin tener en cuenta los espacios ni el **\0**

Donde se ve (.....) completar con el dato necesario.

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #define TAM 30
5
6 int rot_13(char* , char *);
7
8 int main(void)
9 {
10     char dataPtr[TAM], *dataPtrCodificada;
11     int devuelve, longitud, i;
12
13     printf("Ingrese texto a codificar\n");
14     fgets(dataPtr, TAM, stdin);
15     for (i=0; dataPtr[i]!='\0'; i++){
16     }
17     dataPtr[i-1]='\0';
18     longitud = strlen(dataPtr);
19
20     dataPtrCodificada = (char *) malloc ( (longitud+1) * sizeof(char));
21
22     devuelve = rot_13( (.....) , (.....));
23     if (devuelve == -1){
24         printf("el mensaje contiene un caracter distinto de una letra o un espacio\n");
25     }else{
26         printf("el mensaje %s fue codificado a %s\n", (.....), (.....));
27         printf("la cantidad intercambiada fue de: %d\n", (.....));
28     }
29     free ({ (.....) });
30     return (0);
31 }
```