


# Repaso parcial

1c-2021



Legolas se cansó de cortar le la barba a Gimli y armó el siguiente algoritmo que lo hace por él ....





```
void cortar_la_barba_de_gloin(int largo_actual_de_la_barba_de_gimli) {  
    char gimli_esta_conforme = 'N';  
    while(gimli_esta_conforme != 'S') {  
        largo_actualde_la_barba_de_gimli--;  
        print("Estás conforme? (S para terminar)");  
        scanf(" %c", &gimli_esta_conforme);  
    }  
}
```

1. ¿Qué buenas prácticas no se están cumpliendo en el siguiente algoritmo?
2. ¿Cómo las corregiría?

# Discutamos...



```
void cortar_la_barba_de_gloin(int largo_actual_de_la_barba_de_gimli) {  
    char gimli_esta_conforme = 'N';  
    while(gimli_esta_conforme != 'S') {  
        largo_actualde_la_barba_de_gimli--;  
        print("Estás conforme? (S para terminar)");  
        scanf(" %c", &gimli_esta_conforme);  
    }  
}
```




```
void cortar_la_barba_de_gimli(int largo_actual_barba) {  
    char gimli_conforme = 'N';  
    while( gimli_conforme != 'S') {  
        largo_actual_barba--;  
        print("Estás conforme? (S para terminar)");  
        scanf(" %c", &gimli_esta_conforme);  
    }  
}
```

Ojo, este tipo de ejercicios no tienen una única solución...



Discutamos lo mismo pero un poco más rápido...



```
int cortar_pelo(int largo_actual) {  
    char esta_conforme = 'N';  
    while(true) {  
        largo_actual--;  
        print("Estás conforme? (S para terminar)");  
        scanf(" %c", &esta_conforme);  
        if (esta_conforme == 'S'){  
            return largo_actual;  
        }  
    }  
}
```




```
int cortar_pelo(int largo_actual) {  
    char esta_conforme = 'N';  
    while(esta_conforme != 'S') {  
        largo_actual--;  
        print("Estás conforme? (S para terminar)");  
        scanf(" %c", &esta_conforme);  
    }  
    return largo_actual;  
}
```

Lo mismo, esto no es absoluto....

Ahora sigamos con otro tema !!!







Los elfos no son un pueblo de muchos amigos, pero cuando alguien es merecedor, le hacen regalos extraordinarios.

Se armó un vector con los amigos de los elfos (actuales o que alguna vez lo fueron) y los regalos que les hicieron, de cada amigo se tiene la siguiente información:

```
typedef struct elfamigo {  
    char nombre[MAX_NOMBRE];  
    char nombre_mejor_regalo[MAS_NOMBRE_REGALO];  
    bool amigo_actualmente;  
    int cantidad_regalos;  
} elfamigo_t;
```

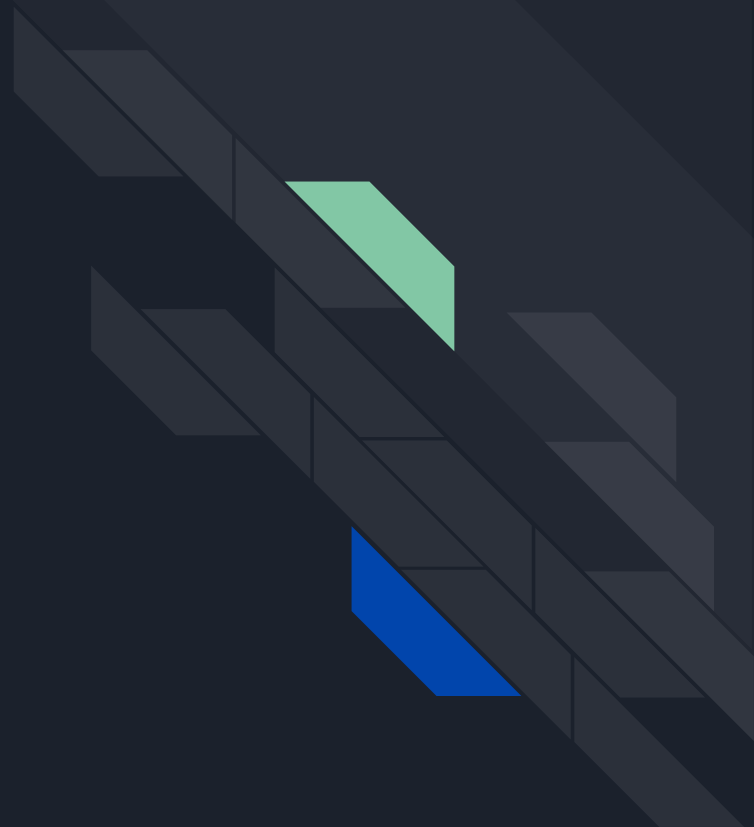
Cree una rutina recursiva, que muestre los nombres de los que son amigos actualmente.

¿Qué parámetros recibe y por qué?  
¿Dónde se ven los componentes principales de la rutina recursiva?

# Spoiler :

Cómo recorreremos un arreglo recursivamente?

Necesitamos 3 argumentos.



# Ejemplo de implementación :



```
void recorrer_arreglo(int arreglo [], int principio, int tope) {  
    if (principio < tope){  
        //Lo que queremos hacer  
        recorrer_arreglo(arreglo, principio+1, tope);  
    }  
    return;  
}
```

Link a c tutor : [shorturl.at/auzQS](https://shorturl.at/auzQS)

Ahora llevémoslo a  
nuestro ejercicio...



# Y luego tenemos las variantes :

Cree una rutina recursiva, que muestre los nombres de aquellos en los que el nombre empieza con C.

Cree una rutina recursiva, que muestre los nombres de aquellos en los que el mejor regalo tiene más de 10 letras

Cree una rutina recursiva, que muestre los nombres de aquellos que recibieron más de 5 regalos