Pregunta 1

Programe la función *desplazarDer* con el siguiente encabezado:

```
void desplazarDer(char *s, int n);
```

Esta función desplaza el string *s* en *n* caracteres hacia la derecha agregando espacios en blanco al principio y eliminando los últimos *n* caracteres. Por simplicidad puede considerar que *s* tiene al menos *n* caracteres. Está permitido usar *strlen*. Este es un ejemplo de uso:

```
char s[]= "quien vive";
desplazarDer(s, 3); // s es el string "   quien v" con 3 espacios al inicio
```

Restricciones: No use el operador de subindicación de arreglos [ ] ni su equivalente *(p+i), use aritmética de punteros. Sí puede usar *p+i*. No puede pedir memoria con *malloc* ni declarar arreglos.

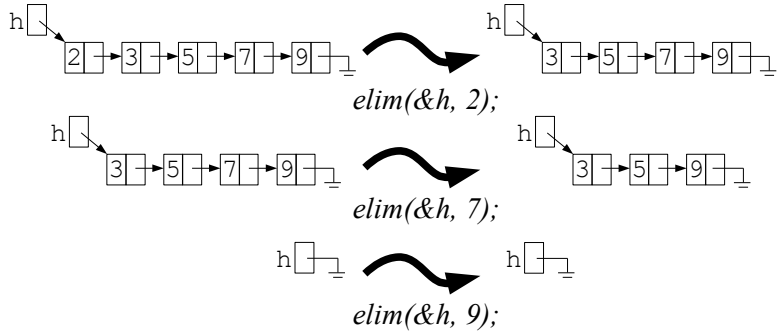
Ayuda: Si *L* es el largo del string, Ud. *debe* recorrer los caracteres desde la posición *L-n-1* hacia la posición 0, copiándolos en su posición de destino. Si recorre recorre de principio a fin sobreescribirá incorrectamente caracteres que no ha copiado todavía.

Pregunta 2

Programe la siguiente función:

```
typedef struct nodo {
  int x; // la etiqueta
  struct nodo *prox;
} Nodo;
int elim(Nodo **pcabeza, int x);
```

Esta función recibe en **pcabeza* una lista simplemente enlazada. Si existe un nodo con etiqueta *x*, debe eliminar ese nodo de la lista, liberar la memoria que ocupa (con *free*) y retornar 1. En caso contrario debe retornar 0. En los siguientes ejemplos de uso el tipo de *h* es *Nodo* *.



Restricción: No puede usar *malloc*. Debe reutilizar los nodos que recibe en **pcabeza*.

Pregunta 3

El programa en assembler Risc-V de más abajo es el resultado de compilar la función *incognito*. Programe la función *equivalente* a *incognito* en C sin usar la instrucción **goto** de C. Preocúpese de *reproducir* en C todos los aspectos de la función original en assembler, en particular el valor retornado. El encabezado de *incognito* es:

```
int *incognito(int a[], int x);
```

```
incognito:
        mv      a4,a0
.L1:
        lw      a5,0(a4)
        addi    a4,a4,4
        bge     a5,a1,.L2     # salta si a5>=a1
        sw      a5,0(a0)
        addi    a0,a0,4
.L2:
        bne     a5,zero,.L1   # salta si a5!=0
        ret
```

Pregunta 4

La función *f* de más abajo invoca las funciones *g* y *h*. Las funciones *g* y *h* son dadas y toman mucho tiempo en calcularse.

```
double g(double **mat, double x);
double h(double **mat, double x);
double f(double **mat, double x) {
  return g(mat, x) + h(mat, x);
}
```

Reprograme la función *f* de manera que el cálculo de *g* y *h* se haga en paralelo. Para ello Ud. debe invocar *fork* para crear un proceso que calcule *h(mat, x)*. El proceso padre calcula *g(mat, x)* y retorna el resultado final.