

Pregunta 1

Programa la función *extraerBits* con el siguiente encabezado:

```
typedef unsigned int uint;
uint extraerBits(uint *px, int n);
```

Sea $x = *px$. Considere que x se descompone en sus 32 bits: $x_{31} \dots x_0$. La función *extraerBits* debe retornar los n bits menos significativos de x , es decir $0 \dots 0x_{n-1} \dots x_0$. Además debe extraer esos bits de $*px$, es decir dejando en $*px$: $0 \dots 0x_{31} \dots x_n$. El siguiente es un ejemplo de uso:

```
uint x= 0b101001110;
uint a= extraerBits(&x, 3); // a=0b110, x=0b101001
uint b= extraerBits(&x, 2); // b=0b01, x=0b1010
uint c= extraerBits(&x, 4); // c=0b1010, x=0
```

Restricciones: No use los operadores de multiplicación, división o módulo ($*$ / $\%$). Use eficientemente los operadores de bits, sumas y restas.

Pregunta 2

Programa la siguiente función:

```
void minusculas(char *str, char *minusc);
```

Esta función recibe un string en *str* y debe entregar en el string *minusc* la primera palabra formada solo por letras alfabéticas minúsculas. Ejemplos:

```
char minusc[80];
minusculas("-24*vel+dis", minusc); // minusc es "vel"
minusculas("Pedro picapiedra", minusc); // minusc es "edro"
minusculas("25*300-5", minusc); // minusc es ""
```

Restricciones: No use el operador de subindicación de arreglos $[]$ ni su equivalente $*(p+i)$. Use aritmética de punteros. No puede pedir memoria adicional usando *malloc* o declarando un arreglo de caracteres. Necesitará declarar punteros adicionales.

Ayuda: busque el primer caracter que sea una letra alfabética minúscula (o el fin del string), luego copie todas las letras alfabéticas minúsculas al string *minusc*, hasta que encuentre una letra que no sea alfabética minúscula. Finalmente termine el string.

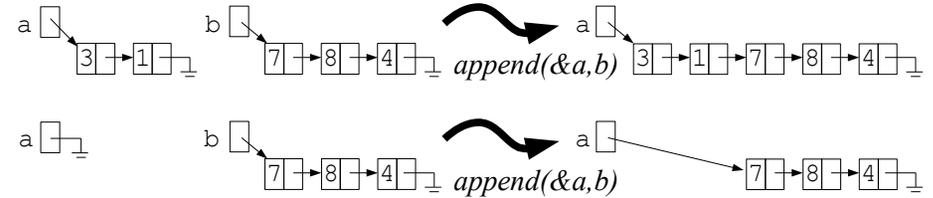
Pregunta 3

Programa la función *append* que adjunta la lista enlazada en *b* a la lista

en **pa*. El encabezado de la función es el siguiente:

```
typedef struct nodo {
    int x;
    struct nodo *prox;
} Nodo;
void append(Nodo **pa, Nodo *b);
```

Estos son 2 ejemplos de uso en donde el tipo de *a* y *b* es *Nodo**:



Restricciones: No puede usar *malloc*. Debe reutilizar los nodos que recibe en **pa* y *b*. Los nodos del resultado deben seguir el mismo orden en que aparecían originalmente **pa* y *b*.

Pregunta 4

Programa la función: *int main(int argc, char *argv[])*. Esta función es para el comando *./extracto* que recibe como parámetros el nombre de un archivo, la posición *k* de una línea en el archivo y un número de líneas *n*. El comando debe desplegar *n* líneas del archivo a partir de la línea *k*. Todas las líneas del archivo son siempre de 81 caracteres de largo (incluyendo el terminador de líneas $\backslash n$), y se enumeran como 0, 1, 2, 3, etc. Una línea contendrá muchos espacios en blanco para completar los 81 caracteres. No necesita considerar el caso en que al archivo le quedan menos de *n* líneas, ni casos de error como que el archivo no exista. El siguiente es un ejemplo de uso con el archivo *loco.txt* de la derecha:

```
$ ./extracto loco.txt 5 3
3:
el norte
y
```

```
los
4
puntos
cardinales
son
3:
el norte
y
el
sur
:-p
loco.txt
```

Restricciones: debe usar las funciones *fopen*, *fclose*, *fseek* para posicionarse en la primera línea que desplegará y *fread* para leer las líneas del archivo.