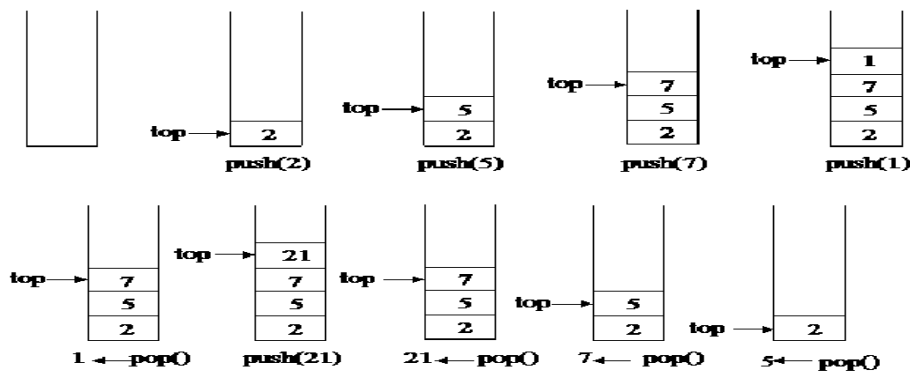


EXAMEN PRÁCTICAS
ARQUITECTURA DE COMPUTADORES
3º ST, TT, T y Teleco-ADE - URJC
Fuenlabrada, 11 de Diciembre de 2015

Notas Importantes:

- En el escritorio (\$HOME/Escritorio) deberás crear un fichero en el que implementarás el programa solicitado. Nombre del fichero: **Apellido1_Apellido2_Nombre_titulación.asm**, donde "titulación" será ST, TT, T o TADE según corresponda.
- Dentro del programa debes escribir también tu nombre y tus apellidos a modo de comentario.
- Avisa al profesor para proceder a la recogida. **NO apagues** el ordenador ni salgas de la sesión.

Se pide implementar (respetando el convenio de llamada a subrutina) el programa **Apellido1_Apellido2_Nombre_titulación.asm** que crea una lista lineal (como la de la imagen) en la que todas las inserciones se hacen por un extremo de la lista, la cima. Este tipo de lista recibe también el nombre de lista *LIFO* (Last In First Out).



En C, la estructura de datos para representar un **nodo** de la lista sería la siguiente:

```
typedef struct _node_t {  
    int val; /* valor del nodo; tamaño palabra */  
    struct _node_t *next; /* puntero al nodo anterior */  
} node_t;
```

En el ejemplo, el nodo siguiente ("next") al nodo 2 es NULL. Se pide implementar tres funciones:

(obligatoria) `node_t * push(node_t *top, int val)`: recibe como parámetros la dirección del nodo situado en la cima de la pila y el valor del nodo a añadir. Crea un nuevo nodo con el valor indicado y lo inserta en la cima de la pila. Devuelve la dirección del nuevo nodo creado. push debe distinguir el caso en el que la pila esté vacía, es decir, cuando la pila todavía no tiene ningún nodo.

(obligatoria) `node_t * remove(node_t *top, int val)`: recibe como parámetro la dirección del nodo en la cima de la pila. Recorre la pila buscando el nodo con valor "val". Si lo encuentra, lo elimina de la pila: el nodo anterior al buscado apuntará ahora al nodo siguiente (next) al nodo buscado. Además, devuelve la dirección del nodo buscado. Si no lo encuentra, devuelve un null.

(opcional) `void print(node_t *top)`: recorre la lista de forma **recursiva** imprimiendo los elementos en orden inverso al que fueron introducidos en la pila. Algoritmo:

```
if (top->next != null) {  
    print(top->next);  
}  
printf("%d\n", top->val);  
return;
```

En el `main` se debe realizar `push` N veces (hasta que se introduzca un 0). Después, se debe eliminar un nodo con valor X e imprimir el valor del nodo eliminado. Por último, se debe imprimir la pila resultante. El registro `$s0` se utilizará en el `main` para apuntar a la cima de la pila. `main` debe mantener actualizado el valor de `$s0`.