

EJERCICIOS DE REGISTROS

Problema 1: Los mininos de la abuela Teresa

A la abuela Teresa le entusiasman los gatos, los tiene de todos los tamaños, colores, edades, etc. Cada uno tiene un nombre que ella le pone cuando los adopta, pero son tantos, que necesita que le ayudemos haciendo un programa para saber cuándo tiene que vacunar a cada minino. Por suerte, es sencillo, ya que el mismo día que los adopta el veterinario les pone la primera vacuna y lo único que hay que hacer es repetir la vacuna cuando ha transcurrido un año.



Se suministra un programa básico con los datos de 4 mininos. Hay que ampliar este programa incorporando las siguientes opciones:

- Dar de alta un gato. Hay que almacenar el nombre y la fecha de adopción del gato
- Mostrar gatos que hay que vacunar. Para ello, puedes emplear el fragmento de código que viene a continuación que permite averiguar el día y mes actual. Sólo hay que revisar las fechas de adopción de los gatos para mostrar los nombres de los gatos que toca vacunar.
- Listado de gatos. Mostrar nombre y fecha de adopción de todos los gatos que tiene la abuela
- Salir del programa

Para acceder a la fecha actual (día y mes) puedes usar este código que almacena en dos variables de tipo entero el día y el mes. Debes incluir la librería <time.h>

```
time_t t;
struct tm *tlocal;
int dia, mes;

t=time(NULL);
tlocal=localtime(&t);
dia=tlocal->tm_mday;
mes=tlocal->tm_mon+1;
```

Ejemplo de ejecución:

```
1. DAR DE ALTA UN MININO
2. COMPROBAR VACUNAS
3. LISTADO
4. SALIR
OPCION...2
-----
TOCA VACUNAR A:
Peluso
Bolita
-----
1. DAR DE ALTA UN MININO
2. COMPROBAR VACUNAS
3. LISTADO
4. SALIR
OPCION...3
-----
1. Manolo 1-1-2015
2. Kira 22-12-2018
3. Peluso 5-3-2017
4. Bolita 5-3-2017
-----
1. DAR DE ALTA UN MININO
2. COMPROBAR VACUNAS
3. LISTADO
4. SALIR
OPCION...1
-----
NOMBRE DEL MININO...Mushu
FECHA DE ADOPCION
      DIA...22
      MES...12
      ANYO...2018
-----
1. DAR DE ALTA UN MININO
2. COMPROBAR VACUNAS
3. LISTADO
4. SALIR
OPCION...4
```

Problema 2: La tienda de Andrea

La tía de Andrea se ha jubilado y le ha dejado la tienda que tiene en el pueblo a Andrea. Como es una tienda pequeña, su tía lo organizaba todo en papel, pero Andrea quiere modernizar un poco el sistema. Andrea te ha pedido que le hagas un programa para ayudarle con la organización. El programa debe tener las siguientes opciones:

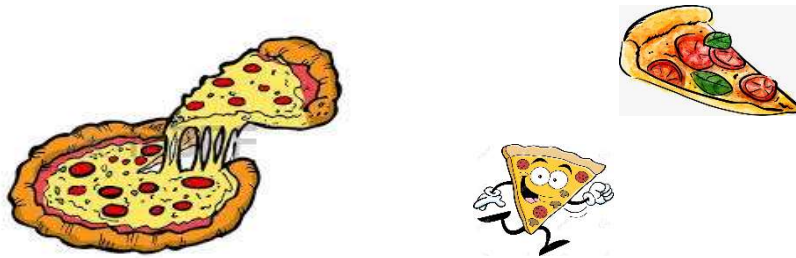
- **Altas:** Consiste en introducir nuevos objetos con sus datos (nombre y precio). Normalmente se efectuarán varias altas consecutivas, deteniendo el proceso cuando no se deseen introducir más productos.

- **Bajas:** Cada vez que se produce una venta hay que dar de baja el objeto. Se introduce el nombre del producto a eliminar, se muestran sus datos y se pide al usuario confirmación de la baja. Para eliminar el producto se desplazan desde el producto inmediato siguiente una posición hacia delante los sucesivos productos, y se reduce en una unidad la variable que indica la posición ocupada.
Pista: se deben tratar de forma especial los casos en que no hay productos o sólo hay uno. Cuando se da de baja un producto su acumula su precio para que cuando se salga del programa Andrea pueda saber cuánto dinero ha ganado en total.
- **Modificaciones:** Andrea ha pensado, que si ve que a algún objeto le había puesto un precio no adecuado, quiere poder cambiarlo. Se introduce el nombre del producto, se muestra y se pide el nuevo precio.
- **Consulta:** mostrar el listado de todos los productos con su precio indicando al final del listado el valor total de los productos.

El programa debe mostrar un menú con las opciones de arriba más la opción de salir.

Problema 3: PizzaHack

La famosa cadena de pizzas **PizzaHack** va a desarrollar un nuevo sistema informático para gestionar los pedidos y tienes que implementar una versión inicial del software.



Se desea almacenar para cada pedido

- número de pedido
- nombre de la pizza ("hawaiana", "cuatro estaciones", ...)
- los ingredientes extras que puede llevar (3 como máximo). Ten en cuenta que no siempre habrá 3 ingredientes, por tanto, necesitas guardar también cuántos hay. Para simplificar la entrada de datos, los ingredientes se codifican con una sola letra (por ejemplo 'a' significa alcaparras, 'b' bacon, etc). No te preocupes del significado, éste ya lo saben los empleados de PizzaHack, solo almacena la letra.
- La hora en que se hizo el pedido (este dato está compuesto de hora y minutos).
- Si el pedido ha sido ya enviado o no

El programa tendrá un menú con las siguientes opciones:

1. **Introducir un nuevo pedido.** Como máximo el sistema podrá manejar 100 pedidos.

- Para simplificar la entrada de datos, ya que esta es una versión inicial del software, pon la hora y minutos del pedido siempre a 0, no la pidas al usuario.
 - Además, en esta versión inicial del programa el número de pedido se pone automáticamente (el primer pedido debería ser el número 1 y así sucesivamente).
 - Inicialmente un pedido está como “no enviado”, como es lógico
2. **Marcar un pedido como “enviado”**, sabiendo su número. Esta opción debe solicitar el número de pedido al usuario y si este existe, marcarlo como enviado.
 3. **Mostrar** el número de veces en total que se ha pedido un determinado ingrediente. Esto es necesario para poder preparar los ingredientes con antelación
 4. **Finalizar** la ejecución del programa

La ejecución del programa deberá consistir en ir seleccionando cualquiera de las 3 primeras opciones del menú, en cualquier orden y número de veces seleccionadas, hasta que elija la opción 4, en cuyo caso el programa finalizará.

Puedes utilizar la función de librería de C `cin.getline(cadena1, TAM_MAX)` para leer una cadena de tamaño TAM_MAX

Otras funciones que puedes utilizar:

```
// Función que nos permite limpiar el buffer de entrada.
// Lo utilizaremos después de leer un número entero,
// para evitar que quede el retorno de carro en el buffer.
void limpiarBuffer() {
    while(cin.get() != '\n');
}
```

EJEMPLO DE EJECUCIÓN

. . . se elige la opción 1 del menú (en negrita lo tecleado por el usuario)

Pedido número 1

Pizza: **Hawaiana**

Número de extras: **3**

Ingrediente 1: **a**

Ingrediente 2: **s**

Ingrediente 3: **b**

. . . se vuelve a elegir la opción 1 (en negrita lo tecleado por el usuario)

Pedido número 2

Pizza: **Barbacoa**

Número de extras: **1**

Ingrediente 1: **a**

. . . se elige la opción 2 (en negrita lo tecleado por el usuario)

Número de pedido:**2**

Pedido 2 marcado como enviado

. . . se elige la opción 3 (en negrita lo tecleado por el usuario)

Ingrediente a contar: **a**

Tienes que preparar 2 unidades del ingrediente a

. . . se elige la opción 4 del menú

Fin de la ejecución del programa.