**Fundamentos de Programación (20 de junio de 2017)**

**(Grados en Ingeniería Mecánica, Eléctrica, Electrónica Industrial y Química Industrial)**

***Ejercicio:*** construir un programa en ***C*** lo más modular posible (atendiendo a los criterios de modularidad) que se ajuste a la especificación dada y documentar el diseño preliminar con la definición de las nuevas tipologías de datos, el diagrama de módulos (estructura del programa) y las interfaces de los módulos (prototipos de funciones), y el diseño detallado con las definiciones de los respectivos sub-programas.

**Ejercicio:** Construir un programa para gestionar un almacén de piezas de recambio de automóviles. Las diferentes piezas debidamente empaquetadas son ubicadas en los diferentes compartimentos de un conjunto de ***8*** estanterías metálicas (ver *figura 1*), apiladas de ***2*** en ***2***, distribuidas paralelamente en el almacén y numeradas de ***0*** a ***7***, respectivamente. Cada estantería tiene ***5*** niveles de altura (***0*** a ***4***), cada uno de los cuales se encuentra dividido longitudinalmente en ***10*** compartimentos de iguales dimensiones (numerados de ***0*** a ***9***).

Para el acceso a cada compartimento de almacenamiento, se dispone de un robot especial (*figura 2*) que se puede desplazar solamente en pasos discretos horizontalmente y/o verticalmente por un conjunto de calles definidas por un raíl según se muestra en la *figura 3* (vista en planta del almacén). El robot dispone además de una plataforma elevadora para acceder a los distintos niveles de las estanterías, así como de medios articulados para la carga y descarga de los paquetes en cuatro direcciones perpendiculares (N, S, E y O, con respecto al sistema de coordenadas de la figura). Se dispone además en el almacén de dos posiciones especiales denominadas muelle de carga (para la entrada de artículos a almacenar) y muelle de descarga (para la retirada de artículos del almacén), ambos accesibles desde el origen del sistema de coordenadas de referencia, en direcciones O y S, respectivamente. En la siguiente tabla se indican las principales órdenes para el funcionamiento del robot. Nótese que los desplazamientos del robot son relativos a su posición actual.

|  |  |
| --- | --- |
| **x ± n** | Desplazamiento horizontal del robot |
| **y ± n** | Desplazamiento vertical del robot |
| **z ± n** | Desplazamiento plataforma elevadora en altura |
| **s dir** | Almacena contenido plataforma en compartimento situado en orientación ***dir*** |
| **l dir** | Carga contenido del compartimento situado en orientación ***dir*** en plataforma |

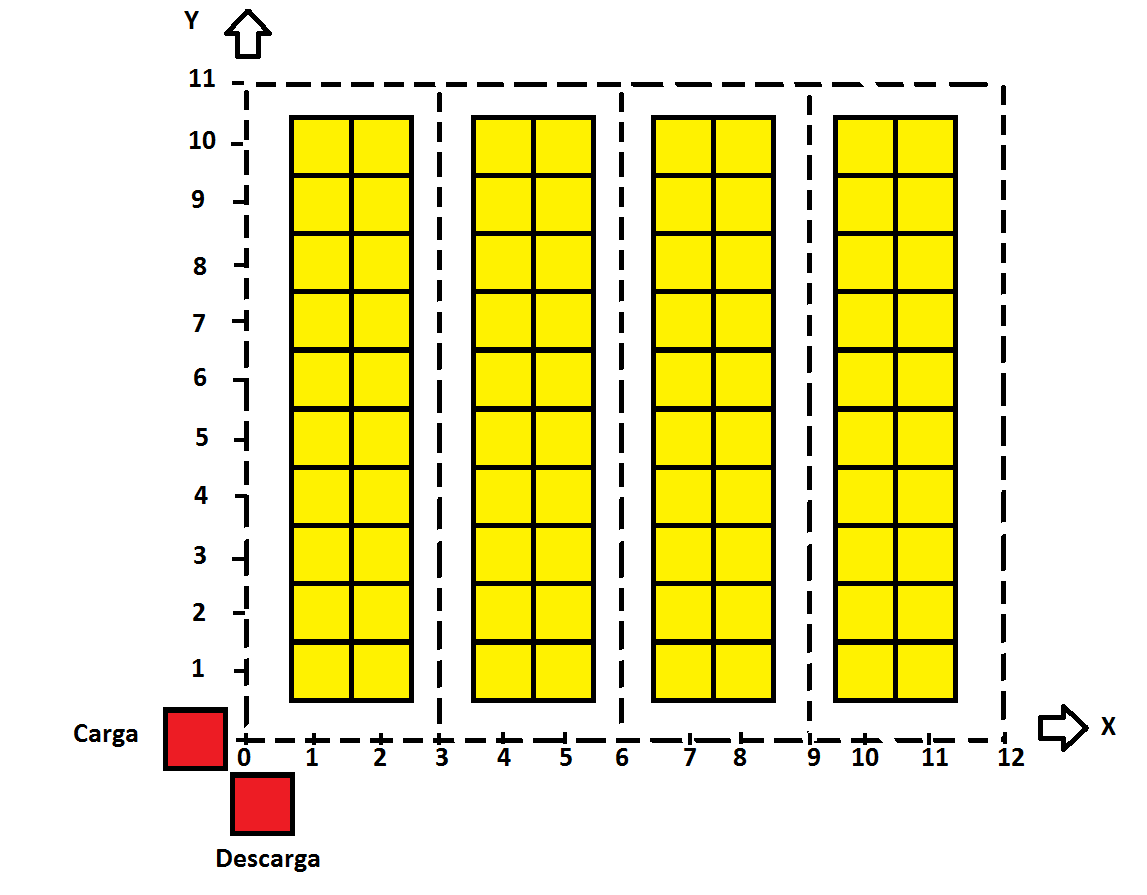
Las órdenes enviadas al robot serán simuladas mediante su escritura en pantalla.



**Figura 1**



**Figura 2**



**Figura 3**

Ej: Almacenar pieza situada en muelle de carga en estantería=1, compartimento=4, nivel=3. Se accede al compartimento a través de la posición del Robot (3,5,3) por su izquierda. Considerar Robot situado en (0,0) con plataforma en z=0. Órdenes al robot:

l O // Carga pieza de muelle de carga

x +3 // Desplazamiento a calle de la estantería

y +5 // Desplazamiento a posición compartimento

z +3 // Elevación plataforma al nivel corresp.

s O // Almacena pieza en compartimento

**Nota**: al desplazar el robot de una posición inicial **(x1,y1)** a una posición final **(x2,y2)**, de las dos rutas alternativas posibles, habrá que escoger la que involucre la menor distancia de desplazamiento del robot.

Al principio de la ejecución del programa, éste deberá cargar de un archivo de texto de nombre ***"almacen2017.txt"*** (en el caso de existir) el nº total de piezas almacenadas, el último nº de referencia asignado, así como la ubicación física de las piezas almacenadas junto con su información asociada. A continuación, presentará al usuario un menú en pantalla con las siguientes opciones, seleccionándose una de ellas por teclado, tras cuya ejecución se volverá a presentar nuevamente el menú de opciones, terminándose la ejecución del programa cuando se seleccione explícitamente la opción de finalización:

1. Almacenar una nueva pieza en el almacén, la cual se encuentra situada inicialmente en el muelle de carga. El programa comprobará que hay sitio para almacenarla, le asignará automáticamente un nº de referencia entero (incrementando en una unidad el último nº asignado, siendo ***1*** el primer nº asignado) y solicitará por teclado la siguiente información de la pieza:
   * Código de la pieza: valor entero comprendido entre ***100 000*** y ***900 000***.
   * Peso de la pieza en kg.
   * Precio de Venta al Público (PVP): importe de la pieza en euros (IVA incluido).

A continuación, localizará la ubicación de un compartimento libre cualquiera. Finalmente, enviará las órdenes pertinentes al robot para que éste realice las operaciones necesarias para el almacenaje de la pieza.

1. Imprimir el estado actual de ocupación del almacén. Se presentarán en pantalla para cada calle por la que circula el robot las estanterías situadas a su izquierda y derecha simulando una proyección en perspectiva con el formato indicado en el siguiente ejemplo, y con la posibilidad de desplazarnos una calle a izquierda ó derecha ó terminar la visualización introduciendo la correspondiente opción por teclado:

**ALMACEN: calle 1 (Izq: Estantería 1, Der: Estantería 2)**

**0 0**

**0 0 0 0**

**0 0 0 0 0 0**

**0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0**

**X X X X 0 0 0 0**

**X X X 0 X X**

**X X X X**

**X X**

**Opciones: I: izq D: der S: salir**

**ALMACEN: calle 0 (Izq: Pared, Der: Estantería 0)**

**0**

**0 0**

**0 0 0**

**0 0 0 0**

**0 0 0 0 0**

**0 0 0 0 0**

**0 0 0 0 0**

**0 0 0 0 0**

**X X X X X**

**X X X X X**

**X X X X**

**X X X**

**X X**

**X**

**Opciones: I: izq D: der S: salir**

1. Retirar una pieza del almacén. El programa solicitará por teclado el código de la pieza a retirar, y localizará la posición de la pieza solicitada. En el caso de disponer de varias piezas con el mismo código, se seleccionará la de menor precio; si no se encontrara la pieza, el programa escribirá en pantalla el correspondiente mensaje de error. Finalmente, y si es pertinente, enviará las órdenes al robot para que éste realice las operaciones necesarias para la retirada de la pieza del almacén y su depósito en el muelle de descarga.
2. Re-estructurar el almacén: el programa cambiará la ubicación de las piezas del almacén para tenerlas situadas en los compartimentos más próximos a los muelles de carga/descarga, esto es, en los compartimentos que involucren el menor nº de movimientos del robot (**x+y+z**) a partir de la posición ***(0,0,0)***. Finalmente, se enviarán las órdenes pertinentes al robot para que éste realice las operaciones necesarias para la re-estructuración. Nota: puede utilizar cualquiera de los dos muelles como almacenamiento temporal para estas operaciones.
3. Finalizar la ejecución del programa. El nº total de piezas almacenadas, el último nº de referencia asignado, así como estado actual de las piezas almacenadas serán registrados en un archivo de texto de nombre ***"almacen2017.txt"***, y en el caso de existir un archivo con el mismo nombre, se le cambiará previamente el nombre por ***"almacen2017.bak"*** borrando si fuera necesario dicho archivo. Finalmente, se enviarán las órdenes pertinentes al robot para posicionarlo en el origen de coordenadas.