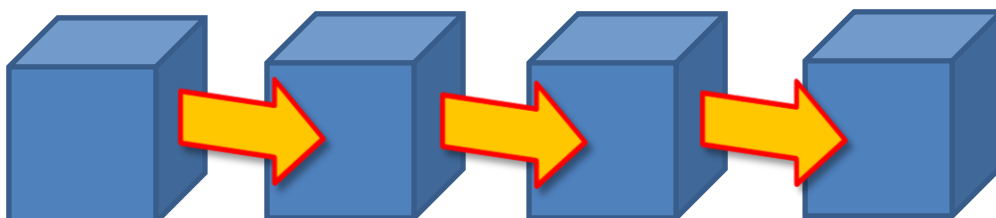




UCAM
UNIVERSIDAD
CATÓLICA DE MURCIA

2020

GRUPO  TUDIUM
FORMACIÓN



Ejercicios Tema 5

LISTAS

STUDIUM

www.grupostudium.com
informacion@grupostudium.com
954 539 952

Enunciados

Para los siguientes ejercicios vamos a usar las siguientes **estructuras**, la primera para las Simples y las Circulares, la segunda para las doblemente enlazadas.

```
// Definición del nodo
typedef struct estructura
{
    int codigo;
    char nombre[10];
    struct estructura *siguiente;
}nodo;
```

```
// Definición del nodo
typedef struct estructura
{
    int codigo;
    char nombre[10];
    struct estructura *anterior;
    struct estructura *siguiente;
}nodo;
```

Para la asignación de memoria de los nodos nuevos, debemos usar la función vista en el temario. Para las doblemente enlazadas, haced la modificación oportuna de dicha función.


```
nodo* nuevo()
{
    nodo *p;
    p = (nodo*) malloc(sizeof(nodo));
    p->siguiente = NULL;
    return(p);
}
```

Listas Simples

1. Insertar al principio - Realizar un programa en C que vaya pidiendo un entero y una cadena y vaya metiendo los datos en una lista simple por el **principio**. Tras insertar, que pregunte si meter otro elemento o acabar. Si se elige esta opción, se mostrará el contenido de la lista al completo y se liberará el espacio ocupado.

2. Insertar al final - Realizar un programa en C que vaya pidiendo un entero y una cadena y vaya metiendo los datos en una lista simple por el **final**. Tras insertar, que pregunte si meter otro elemento o acabar. Si se elige esta opción, se mostrará el contenido de la lista al completo y se liberará el espacio ocupado.

3. Insertar en posición determinada - Realizar un programa en C que vaya pidiendo un entero, una cadena y una posición (otro entero) y vaya metiendo los datos en una lista simple tras la posición indicada. Tras insertar, que pregunte si



meter otro elemento o acabar. Si se elige esta opción, se mostrará el contenido de la lista al completo y se liberará el espacio ocupado

4. Eliminar primer elemento - Realizar un programa en C que genere de forma aleatoria una lista simple de tres elementos, muestre la lista cómo ha quedado, borre el primer elemento, libere el espacio ocupado por dicho elemento, vuelva a mostrar la lista resultante y por último libere el espacio ocupado por el resto de la lista.

5. Eliminar último elemento - Realizar un programa en C que genere de forma aleatoria una lista simple de tres elementos, muestre la lista cómo ha quedado, borre el último elemento, libere el espacio ocupado por dicho elemento, vuelva a mostrar la lista resultante y por último libere el espacio ocupado por el resto de la lista.

6. Eliminar un elemento - Realizar un programa en C que genere de forma aleatoria una lista simple de cinco elementos, muestre la lista cómo ha quedado, pregunte por alguno de los elementos mostrados para borrarlo, borre dicho elemento, libere el espacio ocupado por ese elemento, vuelva a mostrar la lista resultante y por último libere el espacio ocupado por el resto de la lista. Si el elemento a eliminar no existiese, se mostrará error y se volverá a preguntar por otro elemento a borrar.

Listas Circulares

1. Insertar - Realizar un programa en C que vaya pidiendo un entero y una cadena y vaya metiendo los datos en una lista circular por el elemento apuntado por el puntero de la lista. Tras insertar, que pregunte si meter otro elemento o acabar. Si se elige esta opción, se mostrará el contenido de la lista al completo y se liberará el espacio ocupado. Como "primer" elemento considerar el apuntado por el puntero que identifica la lista.

2. Insertar tras elemento - Realizar un programa en C que genere de forma aleatoria una lista circular de tres elementos, muestre la lista cómo ha quedado, pida por teclado un elemento nuevo, indicar el elemento de la lista circular tras el que insertar el elemento nuevo, insertarlo, mostrar la lista tal como haya quedado. Como "primer" elemento considerar el apuntado por el puntero que identifica la lista.

3. Eliminar un elemento - Realizar un programa en C que genere de forma aleatoria una lista circular de cinco elementos, muestre la lista cómo ha quedado, pregunte por alguno de los elementos mostrados para borrarlo, borre dicho elemento, libere el espacio ocupado por ese elemento, vuelva a mostrar la lista resultante y por último libere el espacio ocupado por el resto de la lista. Como "primer" elemento

considerar el apuntado por el puntero que identifica la lista. Si el elemento a eliminar no existiese, se mostrará error y se volverá a preguntar por otro elemento a borrar.

Listas Doblemente Enlazadas

1. Insertar principio - Realizar un programa en C que vaya pidiendo un entero y una cadena y vaya metiendo los datos en una lista doblemente enlazada por el **principio**. Tras insertar, que pregunte si meter otro elemento o acabar. Si se elige esta opción, se mostrará el contenido de la lista al completo, en ambas direcciones, y se liberará el espacio ocupado.

2. Insertar al final - Realizar un programa en C que vaya pidiendo un entero y una cadena y vaya metiendo los datos en una lista doblemente enlazada por el **final**. Tras insertar, que pregunte si meter otro elemento o acabar. Si se elige esta opción, se mostrará el contenido de la lista al completo, en ambas direcciones y se liberará el espacio ocupado.

3. Insertar en posición determinada - Realizar un programa en C que vaya pidiendo un entero, una cadena y una posición (otro entero) y vaya metiendo los datos en una lista doblemente enlazada tras la posición indicada. Tras insertar, que pregunte si meter otro elemento o acabar. Si se elige esta opción, se mostrará el contenido de la lista al completo, en ambas direcciones y se liberará el espacio ocupado

4. Eliminar primer elemento - Realizar un programa en C que genere de forma aleatoria una lista doblemente enlazada de tres elementos, muestre la lista cómo ha quedado, borre el primer elemento, libere el espacio ocupado por dicho elemento, vuelva a mostrar la lista resultante, en ambas direcciones y por último libere el espacio ocupado por el resto de la lista.

5. Eliminar último elemento - Realizar un programa en C que genere de forma aleatoria una lista doblemente enlazada de tres elementos, muestre la lista cómo ha quedado, borre el último elemento, libere el espacio ocupado por dicho elemento, vuelva a mostrar la lista resultante, en ambas direcciones y por último libere el espacio ocupado por el resto de la lista.

6. Eliminar un elemento - Realizar un programa en C que genere de forma aleatoria una lista doblemente enlazada de cinco elementos, muestre la lista cómo ha quedado, pregunte por alguno de los elementos mostrados para borrarlo, borre dicho elemento, libere el espacio ocupado por ese elemento, vuelva a mostrar la lista resultante, en ambas direcciones y por último libere el espacio ocupado por el

resto de la lista. Si el elemento a eliminar no existiese, se mostrará error y se volverá a preguntar por otro elemento a borrar.

Listas Circulares Doblemente Enlazadas

1. Insertar - Realizar un programa en C que vaya pidiendo un entero y una cadena y vaya metiendo los datos en una lista circular doblemente enlazada por el elemento apuntado por el puntero de la lista. Tras insertar, que pregunte si meter otro elemento o acabar. Si se elige esta opción, se mostrará el contenido de la lista al completo, en ambas direcciones y se liberará el espacio ocupado. Como “primer” elemento considerar el apuntado por el puntero que identifica la lista.

2. Insertar tras elemento - Realizar un programa en C que genere de forma aleatoria una lista circular doblemente enlazada de tres elementos, muestre la lista cómo ha quedado, pida por teclado un elemento nuevo, indicar el elemento de la lista circular tras el que insertar el elemento nuevo, insertarlo, mostrar la lista tal como haya quedado, en ambas direcciones. Como “primer” elemento considerar el apuntado por el puntero que identifica la lista.

3. Eliminar un elemento - Realizar un programa en C que genere de forma aleatoria una lista circular doblemente enlazada de cinco elementos, muestre la lista cómo ha quedado, pregunte por alguno de los elementos mostrados para borrarlo, borre dicho elemento, libere el espacio ocupado por ese elemento, vuelva a mostrar la lista resultante, en ambas direcciones y por último libere el espacio ocupado por el resto de la lista. Como “primer” elemento considerar el apuntado por el puntero que identifica la lista. Si el elemento a eliminar no existiese, se mostrará error y se volverá a preguntar por otro elemento a borrar.



17/08/2020