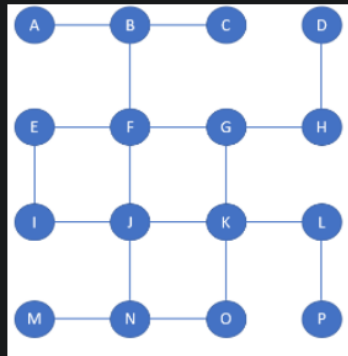


Final 1

- En el siguiente código [link](#)

Se tienen el grafo g , donde se han ingresado como nodos letras de la "a" a la "p" y aristas bidireccionales, de tal forma que se implemente el siguiente grafo.



Implemente el método `Path`, para que halle la ruta más corta entre cualquier par de nodos. El valor de los nodos en la ruta se deben almacenar en la lista `road`.

El método `PrintPath` imprime la lista `road`, y presenta la ruta en la pantalla.

Para el examen se está probando la ruta del nodo "a" al nodo "p". Se puede modificar lo que se considere conveniente dentro del método `Path`.

Tiempo: 1 hora

Final 2

Se tiene el código, [link](#), de una lista doblemente enlazada desordenada que fue llenada con letras, sin repeticiones. Esta lista tiene el método `Find_O_N` que realiza una búsqueda y devuelve un puntero al nodo encontrado en tiempo $O(N)$. Se pide realizar el código del método llamado `Find_O_1` que debe dar una respuesta a la búsqueda en orden 1. No se debe ordenar la lista.

Tiempo: 1 hora

Final 3

- En el siguiente código [link](#)

Se tiene una matriz 'm' de 70 números enteros entre 0 y 999. En esta matriz, se requiere calcular 10 subtotales, contenidos en el vector 'sum' de la siguiente manera:

- $sum[0]$ = suma entre 0 y 99
- $sum[1]$ = suma entre 100 y 199
- $sum[2]$ = suma entre 200 y 299
- $sum[3]$ = suma entre 300 y 399
- $sum[4]$ = suma entre 400 y 499
- $sum[5]$ = suma entre 500 y 599
- $sum[6]$ = suma entre 600 y 699
- $sum[7]$ = suma entre 700 y 799
- $sum[8]$ = suma entre 800 y 899
- $sum[9]$ = suma entre 900 y 999

Estos 10 subtotales deben ser calculados usando 4 threads. Cada thread debe procesar cargas de trabajo similares.

Tiempo: 1 hora