1. Considere el siguiente fragmento de código:
   Declare
   A={NewCell 0}
   B={NewCell 0}
   T1=@A
   T2=@B
   {Show A==B} % a).
   {Show T1==T2} % b).
   {Show T1=T2} % c).
   A:=@B
   {Show A==B} % d).
   Indique que valor será imprimido en cada una de las marcas a), b), c) y d) true, false, A, B o 0.
   Explique claramente.

Respuesta:

a) {Show A == B}
- ¿A y B son la misma celda?
- No, se crearon con NewCell por separado.
- Resultado: false

b) {Show T1 == T2}
- T1 y T2 son ambos el número 0.
- == para enteros compara si son idénticos.
- Resultado: true

c) {Show T1 = T2}
- = compara estructura. Ambos son 0.
- Resultado: true

d) A := @B seguido de {Show A == B}
- @B es 0, así que A := 0 asigna 0 al contenido de la celda A.
- Pero no cambia la referencia de la celda A
- Resultado: false

2. La función fun {Q A B} ... end calcula iterativamente la suma de los elementos entre A y B (A y B incluidos). Implemente Q de tal manera que use celdas para calcular la suma. Su solución debe guardarla en un archivo de nombre sum.oz.

Respuesta: Se usa for y una celda para guardar la suma y hacerlo iterativo.

3. Implemente la función que calcula el factorial de un número dado mediante celdas. Su solución debe guardarla en un archivo de nombre fact.oz.

Respuesta: Se usa for y una celda para guardar el resultado y hacerlo de manera iterativa.

4. La función de Fibonacci está definida de la siguiente manera:
   $fib(n) = 1$ Si $n < 2$
   $fib(n) = fib(n-1) + fib(n-2)$ Si $n \geq 2$
   Implemente esta función utilizando celdas.
   Su solución debe guardarla en un archivo de nombre fibo.oz.

Respuesta: Se utiliza un for y 2 celdas, una para n-2 y otra para n-1 para los números mayores a 1.

5. Implemente el algoritmo de ordenamiento BubbleSort utilizando celdas.
   Su solución debe guardarla en un archivo de nombre bubble.oz.

Respuesta: Se utiliza la conversión a celdas para no tener que copiar listas completas en cada paso y mutar directamente el contenido.

6. Amigos y dinero. Nuestra triste historia comienza con un apretado grupo de amigos. Juntos realizaron un viaje al pintoresco país de Molvania. Durante su estancia, ocurrieron diversos acontecimientos que son demasiado horribles para mencionar. El resultado final fue que la ´última noche el viaje terminó con un cambio trascendental de ”Yo nunca quiero verte de nuevo! ”s. Un cálculo rápido nos dice que esta frase fue dicha casi 50 millones de veces! De regreso en Escandinavia, nuestro grupo de ex-amigos se dan cuenta de que no se han dividido los gastos realizados durante el viaje de manera uniforme. Algunas personas pueden deber varios miles de coronas. La liquidación de las deudas tiende a ser un poco más problemática de lo que debería ser, ya que muchos en el grupo ya no desean hablarse, y menos a´un a dar dinero entre sí.
   Para dar solución a este problema, se le pide a cada persona que diga cuanto dinero debe o le es debido y de que personas sigue siendo amigo. Teniendo en cuenta esta información, es pobile determinar si todo el mundo puede pagar su deuda o recibir el dinero que le es debido pasando dinero solamente entre personas que aún siguen siendo amigos. Implemente la función {PagoDeuda Amistades Deudas}. Donde Amistades corresponde a una lista de parejas I#J que representan que la persona I mantiene una relaci´on de amistad con la persona J y Deudas corresponde a una tupla de valores enteros (positivos y negativos) donde la posición I en esta tupla corresponde al dinero que debe o le es debido a la persona I (positivo si debe dinero y negativo si le es debido). Esta función debe determinar si es posible que todas las deudas sean pagadas entre las personas si solo se pasa dinero entre personas que aún son amigas. Puede asumir que la sumatoria de los valores en la tupla Deudas es igual a 0 y que el tamaño de esta tupla corresponde al número total de personas. Utilice las funciones auxiliares que considere necesarias.
   Ej. Los siguientes llamados a la función PagoDeuda:
   {Browse {PagoDeuda [1#2 2#3 4#5] [100 ˜75 ˜25 ˜42 42]}}
   {Browse {PagoDeuda [1#3 2#4] [15 20 ˜10 ˜25]}}
   Deberían desplegar true y false respectivamente.
   Su solución debe guardarla en un archivo de nombre amigos.oz.

   Respuesta: Se realiza la generación de los adyacentes y se arma un grafo luego se identifican los diferentes caminos y por último se valida si el costo del camino es 0