Examen Lenguajes

P2:

1) Se va a requerir una pila de tamaño n, ya que por cada elemento de la lista, se llamará a (* (expt 2 x) (pow2-list xs)), teniendo que guardar en la pila el valor de cada (expt 2 x) que vaya apareciendo. Sólo luego de llegar al último elemento, es que se empezarán a multiplicar los (expt 2 x) con los (pow2-list xs), y por lo tanto, se comenzará a reducir la pila.

2) De ser posible, tendría que ser una implementación que sea recursiva por la cola, es decir, que al retornar pow2-list se entregue un valor independiente, logrando terminar la ejecución del pow2-list que se llamó inicialmente (eliminando el frame de la pila) y continuando con un siguiente pow2-list o bien el valor final esperado.
Una implementación posible para lograr esto, es mediante un acumulador:

```
(define (pow2-list acum l)
    (match l
        ['() acum]
        [(cons x xs)  (pow2-list  (* (expt 2 x) accum)  xs) ] ))
```

De esta manera, en cada llamada recursiva se estará entregando el valor acumulado de las llamadas anteriores y sin tener que almacenar nada de información de la llamada anterior, por lo tanto la cola permanece constante. Hay que tener en cuenta que para esta implementación en particular se ha cambiado la firma de la función, teniendo que llamar a (pow2-list 0 l) en la primera llamada. Aún así, hay métodos un poco más elaborados que lograrían mantener la firma de la función y que también aprovecharían la recursión por la cola.