Some things to add to your linked list module....

Copy `linkedList02.f03` to `linkedList03.f03`

1. Remove `head`, `ptr`, and `tail` from `linkedList03.f03`.
2. Add a new subroutine called `linkedListReal_incrementCurrent` that moves the object's `current` pointer one node forward. Modify `linkedList_getCurrent` to use `Function linkListReal_incrementCurrent` instead of the line

   ```
   if(ASSOCIATED(myLinkedList%current%next))
   myLinkedList%current=>myLinkedList%current%next
   ```

   `Function listLinkReal_incrementCurrent` should take only 1 argument, the `linkedListReal` object.
3. Edit `Function linkListReal_incrementCurrent` so that it has a second dummy argument called `incrementBy`. This dummy argument (`incrementBy`) should be an integer and be optional. It will indicate the number of moves to incrment the linkedListReal object's `current` pointer by. It should accept both positive and negative values to move forward and reverse in the list. If it is not sent by the calling program unit, it should default to 1.

Copy `linkedList03.f03` to `linkedList04.f03` -and- Copy `linkedListMod.f03` to `linkedListMod1.f03`

4. Edit `linkedList04.f03` so that the `program` and `end program` lines use the updated program name.
5. Edit `linkedList04` to use `linkedListMod1` instead of `linkedListMod`.
6. Edit `Function linkedListReal_getCurrent` so that it has a new optional dummy argument called `incrementBy`. This argument should default to 0 within function `linedlistReal_getCurrent`.