ADTseq – sequence ADT

- ADT ==> **A**bstract **D**ata **T**ype
- What is **ADT sequence**?
  - **Collection** of **elements** (values)
  - Collection with **properties** & **operations**
  - **Linear** collection - x x x x x x x …
  - Property – **value** - **4 2 7 9 8 3 1**
  - Property – **ordered** **(position)** - **1 2 3 4 5 6 7**
  - Property – **sorted (?)** - **1 2 3 4 7 8 9**

property == attribute

ADTseq - Operations

1. Display — `display`
2. Add value — `add v`
3. Find value — `find v`
4. Remove value — `rem v`
5. Add value at position — `addpos v p`
6. Remove value at position — `rempos p`
7. Size — `size`

ADTseq – Operation **add**

display	**empty**	**what did we do?**
add 3		**add to an empty sequence**
display	**3**	
add 9		**add at end of sequence**
display	**3 9**	
add 2		**add at beginning of sequence**
display	**2 3 9**	
add 5		**add in middle of sequence**
display	**2 3 5 9**	
add 7		**add in middle of sequence**
display	**2 3 5 7 9**	
size	**5**	

ADTseq – Operation **rem**

display	**2 3 5 7 9**	**what did we do?**
rem 2		**rem at beginning of sequence**
display	**3 5 7 9**	
rem 9		**rem at end of sequence**
display	**3 5 7**	
rem 5		**rem in middle of sequence**
display	**3 7**	
size	**2**	

ADTseq – Operation **addpos**

display	**empty**	**what did we do?**
addpos 9 1		**addpos to an empty sequence**
display	**9**	
addpos 3 2		**addpos at end of sequence**
display	**9 3**	
addpos 2 1		**addpos at beginning of seq.**
display	**2 9 3**	
addpos 5 3		**addpos in middle of sequence**
display	**2 9 5 3**	
addpos 1 4		**addpos in middle of sequence**
display	**2 9 5 1 3**	
size	**5**	

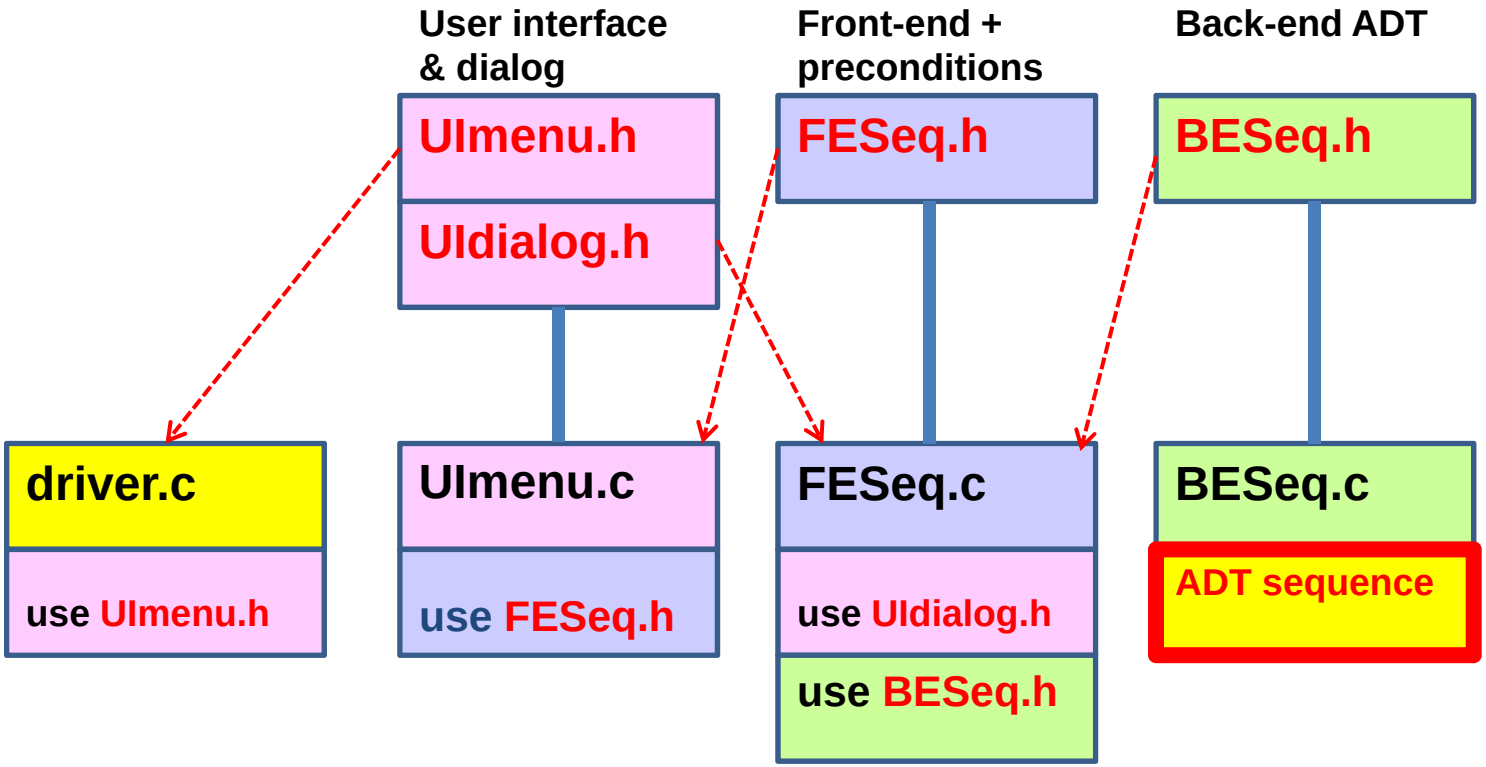
ADTseq – Operation **rempos**

display	**2 9 5 1 3**	**what did we do?**
rempos 1		**rempos at beginning of seq.**
display	**9 5 1 3**	
rempos 4		**rempos at end of sequence**
display	**9 5 1**	
rempos 2		**rempos in middle of sequence**
display	**9 1**	
size	**2**	
addpos 6 0	**error!**	**error – invalid position**
addpos 6 4	**error!**	**error – invalid position**
rempos 0	**error!**	**error – invalid position**
rempos 3	**error!**	**error – invalid position**

ADTseq – Operation **find**

display	**2 3 5 7 9**	**what did we do?**
size	**5**	
		value exists
find 2	**found**	**find at beginning of seq.**
find 5	**found**	**find in middle of sequence**
find 9	**found**	**find at end of sequence**
		value does not exist
find 1	**not found**	**find at beginning of seq.**
find 4	**not found**	**find in middle of sequence**
find 10	**not found**	**find at end of sequence**

ADTseq – implementation UI/FE/BE



xxx.h == interface
xxx.c == implementation

ADTseq – code (ADT - interface)

```
/*-------------------------------------------------------------------------------*/
 /* DSA Sequence program                                                        */
/*-------------------------------------------------------------------------------*/
 #ifndef BELIST_H
#define BELIST_H
/*-------------------------------------------------------------------------------*/
 /* back end function prototypes                                                */
 /*-------------------------------------------------------------------------------*/
void be_display();
void be_addval(int fval);
void be_addpos(int fval, int fpos);
void be_remval(int fval);
void be_rempos(int fpos);
int    be_is_member(int fval);
int    be_size();
#endif
```

ADTseq – preconditions

size	none
display	**if size == 0 ➔ empty** else display x x x x x
find	**if size == 0 ➔ empty** else search ➔ found/not found
add	none
rem	**if size == 0 ➔ empty** else search & remove if found
addpos	**if position not valid ➔ error**
	else add element at position
rempos	**if size == 0 ➔ empty**
	else if position not valid ➔ error
	else remove element at position

ADTseq – code (short functions!!!)

```
-----------------------------------------------------------------------------------------------------------------
void fe_display()
{  if (be_size()==0) ui_putSeqEmpty(); else { ui_putTitleSeq(be_size()); be_display(); } }
-----------------------------------------------------------------------------------------------------------------
int fe_size()         { return be_size(); }
-----------------------------------------------------------------------------------------------------------------
void fe_addsort()  { be_addval(ui_getValue()); }
void fe_remsort()  { if (be_size() == 0) ui_putSeqEmpty(); else be_remval(ui_getValue()); }
-----------------------------------------------------------------------------------------------------------------
void fe_push()      { be_addpos(ui_getValue(), 1); }
void fe_pop()       { if (be_size() == 0) ui_putStackEmpty(); else be_rempos(1); }
-----------------------------------------------------------------------------------------------------------------
void fe_enqueue() { be_addpos(ui_getValue(), be_size()+1); }
void fe_dequeue() { if (be_size() == 0) ui_putQueueEmpty(); else be_rempos(1); }
-----------------------------------------------------------------------------------------------------------------
```

ADTseq – code (short functions!!!)

```
-------------------------------------------------------------------------------------------------------------
void fe_addpos() {
  int pos = ui_getPosition(be_size()+1);                    /* position in 1 .. (size+1)          */
  if (pos<1 || pos > be_size()+1) ui_putPositionError();
  else be_addpos(ui_getValue(), pos); }
-------------------------------------------------------------------------------------------------------------
void fe_rempos() {
  int pos;
  if (be_size() == 0) ui_putSeqEmpty();
  else { pos = ui_getPosition(be_size());                   /* position in 1 .. (size)            */
        if (pos<1 || pos > be_size()) ui_putPositionError(); else be_rempos(pos); }  }
-------------------------------------------------------------------------------------------------------------
void fe_is_member(int v) {
 if (be_size() == 0) ui_putSeqEmpty();
 else if (be_is_member(ui_getValue())) ui_putValueFound(); else ui_putValueNotFound();
}
-------------------------------------------------------------------------------------------------------------
```