8-Off Solitaire

Results

- **pack :: Deck**

```
*EightOffSolitaire> pack
[(Ace,Spades),(Ace,Clubs),(Ace,Hearts),(Ace,Diamonds),(Two,Spades),(Two,Cl
ubs),(Two,Hearts),(Two,Diamonds),(Three,Spades),(Three,Clubs),(Three,Heart
s),(Three,Diamonds),(Four,Spades),(Four,Clubs),(Four,Hearts),(Four,Diamond
s),(Five,Spades),(Five,Clubs),(Five,Hearts),(Five,Diamonds),(Six,Spades),(
Six,Clubs),(Six,Hearts),(Six,Diamonds),(Seven,Spades),(Seven,Clubs),(Seven
,Hearts),(Seven,Diamonds),(Eight,Spades),(Eight,Clubs),(Eight,Hearts),(Eig
ht,Diamonds),(Nine,Spades),(Nine,Clubs),(Nine,Hearts),(Nine,Diamonds),(Ten
,Spades),(Ten,Clubs),(Ten,Hearts),(Ten,Diamonds),(Jack,Spades),(Jack,Clubs
),(Jack,Hearts),(Jack,Diamonds),(Queen,Spades),(Queen,Clubs),(Queen,Hearts
),(Queen,Diamonds),(King,Spades),(King,Clubs),(King,Hearts),(King,Diamonds
)]
```

When ‘pack’ is typed into terminal, a populated deck (list of cards) is returned with 52 unique playing cards.

- **isAce :: Card -> Bool**

```
*EightOffSolitaire> isAce (Ace,Hearts)
True
*EightOffSolitaire> isAce (Eight,Diamonds)
False
```

When ‘isAce’ is typed with a card as its argument, it will return ‘True’ or ‘False’ depending on whether the pip value of the card is Ace.

- **isKing :: Card -> Bool**

```
*EightOffSolitaire> isKing (King,Clubs)
True
*EightOffSolitaire> isKing (Two,Hearts)
False
```

When ‘isKing’ is typed with a card as its argument, it will return ‘True’ or ‘False’ depending on whether the pip value of the card is King.

- **sCard :: Card -> Card**

```
*EightOffSolitaire> pCard (Three,Clubs)
(Two,Clubs)
```

When ‘sCard’ is typed with a card as its argument, it will return the successor to that card (next card in suit).

```
*EightOffSolitaire> sCard (Ten,Diamonds)
(Jack,Diamonds)
```

When ‘pCard’ is typed with a card as its argument, it will return the predecessor to that card (previous card in suit).

- **shuffle :: Deck -> Deck**

```
*EightOffSolitaire> shuffle pack
[(Queen,Spades),(Four,Hearts),(Ace,Hearts),(Six,Diamonds),(Four,Clubs),(Four,Spades),(Five,Spades),(Eight,Hearts),(Five,Clubs),(Seven,Clubs),(Three,Hearts),(Jack,Diamonds),(Nine,Spades),(Ace,Clubs),(Jack,Clubs),(Four,Diamonds),(Ace,Diamonds),(Seven,Hearts),(Eight,Spades),(Three,Clubs),(Two,Spades),(Queen,Diamonds),(Nine,Clubs),(Three,Spades),(Ten,Hearts),(Ten,Clubs),(King,Spades),(Eight,Clubs),(Two,Hearts),(Jack,Hearts),(King,Diamonds),(Ten,Spades),(Seven,Spades),(King,Hearts),(Five,Hearts),(Six,Clubs),(Six,Hearts),(King,Clubs),(Seven,Diamonds),(Two,Diamonds),(Nine,Diamonds),(Queen,Clubs),(Queen,Hearts),(Six,Spades),(Nine,Hearts),(Two,Clubs),(Eight,Diamonds),(Ten,Diamonds),(Ace,Spades),(Three,Diamonds),(Jack,Spades),(Five,Diamonds)]
```

When ‘shuffle’ is typed with a [pack of 52 cards] as its argument, it will return the same deck but with its cards in random order.

- **eODeal :: Deck -> EOBoard**

```
*EightOffSolitaire> eODeal pack
([],[[(Queen,Spades),(Four,Hearts),(Ace,Hearts),(Six,Diamonds),(Four,Clubs),(Four,Spades)],[(Five,Spades),(Eight,Hearts),(Five,Clubs),(Seven,Clubs),(Three,Hearts),(Jack,Diamonds)],[(Nine,Spades),(Ace,Clubs),(Jack,Clubs),(Four,Diamonds),(Ace,Diamonds),(Seven,Hearts)],[(Eight,Spades),(Three,Clubs),(Two,Spades),(Queen,Diamonds),(Nine,Clubs),(Three,Spades)],[(Ten,Hearts),(Ten,Clubs),(King,Spades),(Eight,Clubs),(Two,Hearts),(Jack,Hearts)],[(King,Diamonds),(Ten,Spades),(Seven,Spades),(King,Hearts),(Five,Hearts),(Six,Clubs)],[(Six,Hearts),(King,Clubs),(Seven,Diamonds),(Two,Diamonds),(Nine,Diamonds),(Queen,Clubs)],[(Queen,Hearts),(Six,Spades),(Nine,Hearts),(Two,Clubs),(Eight,Diamonds),(Ten,Diamonds)]],[(Ace,Spades),(Three,Diamonds),(Jack,Spades),(Five,Diamonds)])
```

When ‘eODeal’ is typed with a [pack of 52 cards] as its argument, it returns a populated eOBoard with foundations, columns and reserves. Initially, 0 cards are dealt to the foundation pile, 6 cards are dealt to each column (8 in total, separated by ‘,’) and 4 cards are dealt to reserves.

- **eODealA :: Deck -> [Deck]**

```
*EightOffSolitaire> eODealA pack
[[(Ace,Spades),(Ace,Clubs),(Ace,Hearts),(Ace,Diamonds),(Two,Spades),(Two,Clubs)],[(Two,Hearts),(Two,Diamonds),(Three,Spades),(Three,Clubs),(Three,Hearts),(Three,Diamonds)],[(Four,Spades),(Four,Clubs),(Four,Hearts),(Four,Diamonds),(Five,Spades),(Five,Clubs)],[(Five,Hearts),(Five,Diamonds),(Six,Spades),(Six,Clubs),(Six,Hearts),(Six,Diamonds)],[(Seven,Spades),(Seven,Clubs),(Seven,Hearts),(Seven,Diamonds),(Eight,Spades),(Eight,Clubs)] … [ … ]]
```

Auxilary function to eODeal that takes a deck and returns a list of 6 card decks, used for creating each column.

- **toFoundations :: EOBoard -> EOBoard**

For this test I created a dummy deck that thoroughly tests the auxilary functions utilised by **toFoundations** ~

```
*EightOffSolitaire> myEOBoard
([],[[(Ace,Hearts),(Nine,Hearts),(Jack,Clubs),(Nine,Spades),(Two,Hearts)],
[(Ace,Spades),(Two,Diamonds),(Three,Clubs),(Seven,Hearts),(Ten,Clubs),(Thr
ee,Hearts),(Six,Spades)],[(Seven,Spades),(Four,Hearts),(Queen,Clubs),(Four
,Diamonds),(Six,Diamonds),(King,Spades)],[(Seven,Clubs),(Four,Clubs),(Ten,
Diamonds),(King,Hearts),(Five,Clubs)],[(Two,Spades),(Three,Spades),(King,C
lubs),(Jack,Hearts),(Five,Spades),(Queen,Hearts)],[(Two,Clubs),(Six,Hearts
),(Ten,Hearts),(Jack,Spades),(Eight,Spades),(Queen,Diamonds)],[(Nine,Clubs
),(Three,Diamonds),(Eight,Clubs),(Eight,Hearts),(Six,Clubs)],[(Eight,Diamo
nds),(Five,Hearts),(Ten,Spades)]],[(Ace,Diamonds),(Ace,Clubs),(Nine,Diamon
ds),(Four,Spades)])
```

Supplying this board as an argument for **toFoundations** produces:

```
*EightOffSolitaire> toFoundations myEOBoard
([(Ace,Hearts),(Four,Spades),(Two,Diamonds),(Three,Clubs)],[[(Nine,Hearts)
,(Jack,Clubs),(Nine,Spades),(Two,Hearts)],[(Seven,Hearts),(Ten,Clubs),(Thr
ee,Hearts),(Six,Spades)],[(Seven,Spades),(Four,Hearts),(Queen,Clubs),(Four
,Diamonds),(Six,Diamonds),(King,Spades)],[(Seven,Clubs),(Four,Clubs),(Ten,
Diamonds),(King,Hearts),(Five,Clubs)],[(King,Clubs),(Jack,Hearts),(Five,Sp
ades),(Queen,Hearts)],[(Six,Hearts),(Ten,Hearts),(Jack,Spades),(Eight,Spad
es),(Queen,Diamonds)],[(Nine,Clubs),(Three,Diamonds),(Eight,Clubs),(Eight,
Hearts),(Six,Clubs)],[(Eight,Diamonds),(Five,Hearts),(Ten,Spades)]],[(Nine
,Diamonds)])
```

The cards highlighted represent the foundation pile they will be appended too after execution

We can see through the following test result that:

toFoundationsA works as the *(Ace,Spades)* at the head of column 1 has been placed in the second foundation pile, and that column 1 has been updated (head now *(Nine,Hearts)*

toFoundationsAA works as the *(Ace,Diamonds)* in the reserves has been placed in the third foundation pile, and the reserves has been updated without the card

toFoundationsAAA works as the *(Two,Spades)* at the head of column 4 has been appended to the first foundation pile, after (Ace,Spades). The column is updated and the head is now *(Three,Spades)*

toFoundationsAAAA works as the *(Four,Spades)* in the reserves is appended to the second foundation pile, after *(Three,Spades).* The reserves have been updated without the card