Foundations of Computer Science (Semester 2) – 2015

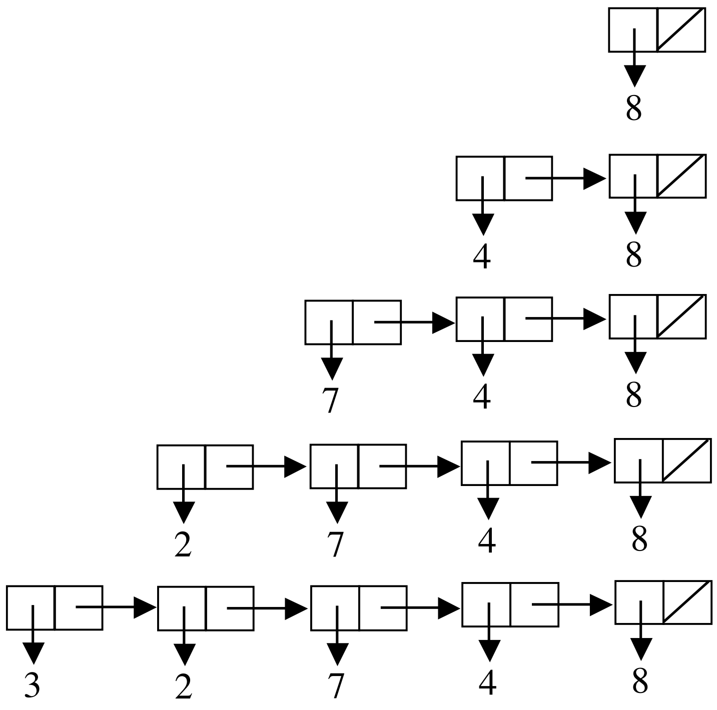
Assessed Exercise Sheet 1 – 10% of Continuous Assessment Mark

Deadline : 11pm Sunday 25th January, via Canvas

Question 1 (14 marks)

You need to insert the numbers 8, 4, 7, 2, 3 one at a time in that order into to an initially empty stack.

Show, in the standard two-cell notation used in the lectures, the state of the list at each stage of that process.



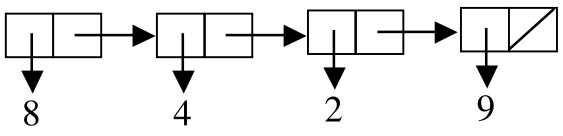
Represent that process using the standard constructors `push` and `EmptyStack`.

```
push(3, push (2, push (7, push (4, push (8, EmptyStack)))))
```

Question 2 (16 marks)

The numbers 9, 2, 4, 8 have been added in that order into an initially empty stack.

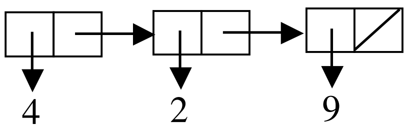
Show, in the standard two-cell notation, the resulting stack.



What is the result of the operation `top` on that stack?

8

What is the result of the operation `pop` on the original stack?



What is the result of the operation `pop` followed by `pop` followed by `top` on the original stack?

2

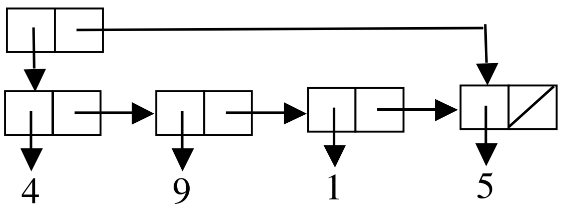
What is the result of the operation `pop` followed by `pop` followed by `pop` followed by `pop` on the original stack?

EmptyStack

Question 3 (16 marks)

You have inserted the numbers 4, 9, 1, 5, one at a time in that order into to an initially empty queue.

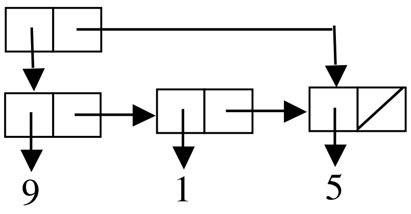
Show, in the standard two-cell notation, the resulting queue.



What is the result of the operation `top` on that queue?

4

What is the result of the operation `pop` on the original queue?



What is the result of the operation `pop` followed by `pop` followed by `top` on the original queue?

1

Question 4 (14 marks)

In the lecture notes (section 2.2) we looked at a procedure `last(L)` that returned the last item in the given list `L`. Modify that to give a recursive procedure `secondlast(L)` that returns the second to last item in a given list `L`.

```
secondlast(L) {
  if ( isEmpty(L) )
     error('Error: Empty list in procedure secondlast.')
  elseif ( isEmpty(rest(L)) )
     error('Error. Short list in procedure secondlast.')
  elseif ( isEmpty(rest(rest(L)) )
     return first(L)
  else return secondlast(rest(L))
}
```

What is the time complexity of your algorithm?

Linear in n, or O(n), where n is the length of the list.

Question 5 (16 marks)

It is often useful to know whether two given lists are the equal, i.e. contain the same items in the same order. Write a recursive procedure `equalList(L1,L2)` that returns `true` if the two lists `L1` and `L2` are the same, and `false` if they are not. The only other procedures it may call are the standard list operators `first`, `rest` and `isEmpty`.

```
equalList(L1,L2) {
  if ( isEmpty(L1) and isEmpty(L2) )
      return true
  elseif ( isEmpty(L1) or isEmpty(L2) )
     return false
  elseif ( first(L1) != first(L2) )
     return false
  else return equalList(rest(L1),rest(L2))
}
```

What is the time complexity of your algorithm?

Linear in n, or O(n), where n is the length of the shortest list.

Question 6 (24 marks)

A set can be represented as a list in which repeated items are not allowed and the order of the items does not matter.

Suppose you have sets `S1` and `S2` represented as linked-lists, and access to the standard list operators `first`, `rest` and `isEmpty`. Write a recursive procedure `member(x,S1)` that returns `true` if item `x` is in set `S1`, and `false` if it is not.

```
member(x,S1) {
  if ( isEmpty(S1) )
     return false
  elseif ( x == first(S1) )
     return true
  else return member(x,rest(S1))
}
```

Now write a recursive procedure `subset(S1,S2)` that returns `true` if set `S1` is a subset of set `S2`, and `false` if it is not. It is allowed to call any of the standard list operators `first`, `rest` and `isEmpty` and your `member` procedure.

```
subset(S1,S2) {
  if ( isEmpty(S1) )
     return true
  elseif ( !member(first(S1),S2) )
     return false
  else return subset(rest(S1),S2)
}
```

Finally, write a procedure `equalset(S1,S2)` that returns `true` if set `S1` is equal to set `S2`, and `false` if it is not. It is allowed call any of the standard list operators `first`, `rest` and `isEmpty` and your `member` and `subset` procedures.

```
equalset(S1,S2) {
   return ( subset(S1,S2) and subset(S2,S1) )
}
```