

# Homework 1

## Problem #1

- One good example:  
Prepare the mixture  
Prepare wet mixture
  - A. Cream butter in a large bowl
  - B. Add 1¼ cups of sugar
  - C. Repeat: mix mixture Until: light and fluff
  - D. Repeat: Beat in one egg Until: No more eggPrepare dry mixture
  - A. In a separate bowl add flour
  - B. Add baking soda
  - C. Add saltMix wet and dry mixture
  - A. Add milk as you are mixing
  - B. Mash berries into mixture
  - C. Stir

## Problem #1

- Common mistakes:
  - Repeat
    - Add flour mixture
    - Add milk
    - Add berry mixture
  - Until No more mixtures remain
  
- Repeat:
  - Fill muffin cups with dough
- Until: no more dough

## Problem #1

- Mix flour, baking powder and salt. Add flour mixture, alternating with milk. Mash 1/2 cup berries and stir into batter. Fold in remaining berries.
  - Mix flour, baking powder and salt.
  - ...
  - Add flour mixture, alternating with milk.

## Problem #1

- Loop
  - Loop
    - Fill one muffin cup with mixture
  - Until muffin cup full
  - Go on to the next one
  - Until no muffin cups left

## Problem #2

- Algorithm: Mysterious(A)
 

```
placeholder i, j, min_pos
loop i from 1 to A.length - 1, begin
  min_pos ← i
  loop j from i + 1 to A.length, begin
    if A[j] < A[min_pos], do min_pos ← j
  end loop j
  swap A[i] and A[min_pos]
end loop i
```
- Input A=[5,2,4,6,1,3];

## Problem #2

- Tracing:
  - What if  $j$  starts from  $i$ , instead of  $i+1$ ?
- Purpose:
  - Sort the sequence of numbers from left to right in ascending order
- English description of the method:
  - The method provided by Mike
  - Starting from the 1<sup>st</sup> position, find the minimum number of the (unsorted) list on the right including the 1<sup>st</sup> number, and swap the minimum with the first number. Repeat the process with every position going from left to right.

## Problem #3

- In plain English, describe your method to sum the numbers stored in an input sequence. Express your method in pseudocode for an input sequence  $A$ .
- Description:
  - Create a placeholder named "sum" to record the total of the sequence. Going from left to right, add the values in the sequence one by one to sum.

## Problem #3

- Algorithm: Sum(A)  
**placeholder:** sum, i  
sum  $\leftarrow$  0  
**loop** i **from** 1 **to** A.length, begin  
    sum  $\leftarrow$  sum+A[i]  
end loop i  
return sum

## Problem #4

- In plain English, describe your method for finding the median of an input sequence. Express your method in pseudocode for an input sequence  $A$ . A median is the number in the **middle position** of a **sorted** sequence. You may assume the sequence of odd length.

## Problem #4

- In plain English: Sort the sequence first using bubble-sort (or any sorting methods of your choice), find the position of the middle cell of the list and return the content stored at the middle cell.

## Problem #4

```
// Bubble sorting
placeholders i,j, med_pos
loop i from A.length to 2, begin
  loop j from 2 to i, begin
    if A[j-1]>A[j], do swap A[j-1] and A[j]
  end loop j
end loop i
// find the median
med_pos ← A.length/2
return A[mid_pos]
```