Derek Feng (derekf2)

Shuqing Ye (shuqiny2)

Shuvam Ghosh (shuvamg)

Jinghao Su (jinghaos)

* Read and **understand** programming **problem** statements. Ask questions.
* **Identify edge cases** for the problem
* **Define** effective **test case**(s) and expected result(s) for program
* **Design** one **algorithmic solution** on paper (or whiteboard)
* **Analyze** the **time** and space **complexity** of the solution
* **Write** nearly correct **code on paper** (or whiteboard) to solve problem
* **Explain** your algorithm/**program** to others
* **Simulate test case** and verify your program produces correct results
* Maybe, **implement, test**, and demonstrate correct function of the solution

**Identify edge cases** for the problem:

1. The number of students divided by the number of students in each group has remainder
2. The file input is not found or incorrect
3. Group size is 0 or negative number
4. All the students are absent
5. Group size is larger than the number of students
6. A very large file
7. Roster.txt: file doesn’t exist, invaild input(none…...)
8. Group size: 1, # of enrolled students, some invaild input(none……)
9. Absent students: 0 students, all of the enrolled students, students not enrolled in this class…..

.

**Define** effective **test case**(s) and expected result(s) for program:

1. **Case:**The number of students is not perfectly divisible by group size.

**First argument: roster.txt (Has no. of students = 14)**

**Group size: 3**

**Absent students: Raymond Klefstad**

**Expected Result: (Two groups will have 3 students, two groups will have 4 students)**

**groups:**

1. Name1

Name2

Name3

1. Name4

Name5

Name6

1. Name 7

Name 8

Name 9

Name 10

4. Name11

Name12

Name13

Name14

**2. Case: The groups size is 1 and the number of groups is the number of students**

**First argument: roster.txt (has 3 names)**

**Group size: 1**

**Absent students: Raymond Klefstad**

**Expected Result:**

1. **Name1**
2. **Name2**
3. **Name3**

**3. Case: Invalid Input**

**File: Roster.txt**

**Group size: 0**

**Absent student: Raymond Klefstad**

**Expected Output:**

**Invalid Group size**

To make it more organized and intuitive, let’s set test cases from 3 perspectives/parameters:

1. roster file
2. The size of groups
3. Absent students

@Test

1. test different group sizes

Just for example….

Case 1:

Input: roster.txt, 17, none

Expected output: 17 groups

Case 2:

Input: roster.txt, 16, none

Expect output: 16 groups, 15 \* 1 + 1 \* 2

Case 3:

Input: roster.txt, 9, none

Expect output: 9 groups, 8 \* 2 + 1 \* 1

…...

Case:

Input: roster.txt, 1, none

Expect output: 1 groups

Case :

Input: roster.txt, 0, none

Expect output: invaild error

1. Test different absent students

Case 1:

input: roster.txt, 2, [studentA]

Expect output: 2 groups of 8

Case 2:

input: roster.txt, 2, [studentA, studentB]

Expect output: 1 group of 8, 1 group of 7

…….

Case 3:

input: roster.txt, 2, [all the students]

Expect output: 1 group of 0

Case 4:

input: roster.txt, 2, [some weird random name]

Expect output: no such student error

**Design** one **algorithmic solution** on paper (or whiteboard, or a text file.) put the screenshot or text here:

Attending students = students in roster - absent students

Number of Groups = size(Attending array) / group size

While attending students not empty:

Randomly select a student from attending array

Place student into a group and move onto the next group

Delete student from the attending array

Print out the groups in specific format

**Analyze** the **time** and space **complexity** of the solution:

**The time complexity is O(N) because our while loop will go through all the students in the list. Other operations are O(1)**

**The space complexity is O(N) also because we allocated N slots in the groups array for the randomly selected students.**

**Write** nearly correct **code on paper** (or whiteboard, or text file) to solve problem:

Attending students = students in roster - absent students

Number of Groups = size(Attending array) / group size

While attending students not empty:

Randomly select a student from attending array

Place student into a group and move onto the next group

Delete student from the attending array

Print out the groups in specific format

**Explain** your algorithm/programinsimple words:

* 1. Reading the file, load all the students into an array
* 2. “Attending array” will have all the students except the absentees.
* 3. Determine the number of groups by dividing the size of the “Attending array” by group size
* 4. Randomly select a student from the array and place the student into a group. Delete the student from the “Attending array”. Repeat the process until the array is empty
* 5. Print out the groups in certain format

**Simulate test case** and verify your program produces correct results:

javac roster.java

java roster roster.txt 3 “Raymond Klefstad”

Expected result:

**Groups:**

1. Name1

Name2

Name3

1. Name4

Name5

Name6

1. Name 7

Name 8

Name 9

Name 10

1. Name11

Name12

Name13

Name14

Output:

**Groups:**

1. Name1

Name2

Name3

1. Name4

Name5

Name6

1. Name 7

Name 8

Name 9

Name 10

1. Name11

Name12

Name13

Name14