Ram Luitel
ComS 417
Hw#1

Problem 2.9

Consider an application App that takes two inputs name and age where name is a nonempty string containing at most 20 alphabetic characters and age is an integer that must satisfy the constraint $0 \leq age \leq 120$. The App is required to display an error message if the input value provided for age is out of range. The application truncates any name that is more than 20 characters in length and generates an error message if an empty string is supplied for name.

a) For the application described in Exercise 2.9, define a uni-dimensional input partitioning.

b) Give an All Combinations test set for the partitioning.

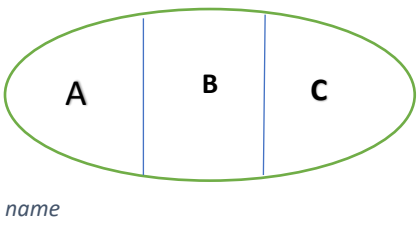
nameNAge(String name, int age)

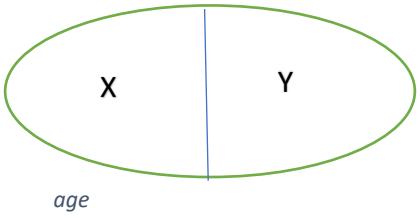
There are two parameters; name and age

- Equivalences classes for name:
  - A) Empty sting (no name or null), that is name.length() = 0
  - B) $20 \leq$ name.length() > 0
  - C) name.length() > 20
- Equivalences classes for age:
  - X) Age in range i.e ($0 \leq age \leq 20$)
  - Y) Age out of range i.e ($0 > age > 120$)

  Note: I am considering that user input is correct that is user is inputting string input as name and integer as age input. I am not testing special character in name meaning any space and special character falls under the length of 20 is still valid input.

  <u>All possible combination</u>





(A,X), (A,Y), (B,X), (B,Y), (C,X), (C,Y)

Problem 2.11

An application takes two inputs x and y where $x \leq y$ and $-5 \leq y \leq 4$. (a) Partition the input domain using uni-dimensional and multidimensional partitioning

a) For the application described in Exercise 2.11, define a uni-dimensional input partitioning.

b) Give an All Combinations test set for the partitioning.

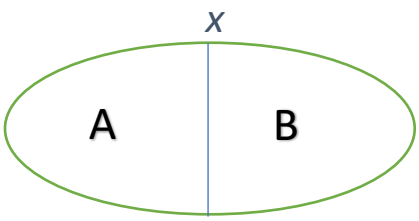
application(int x, int y)

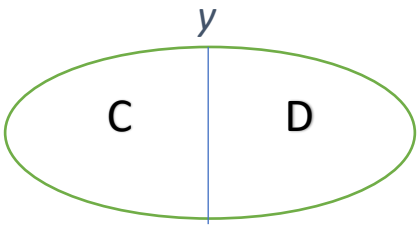
There are two parameters x and y

- Equivalence classes for x:
  A) $x \leq y$ i.e. x is smaller than equal to y
  B) $x > y$ i.e. x is greater than y

- Equivalence classes for y
  C) y in range ($-5 \leq y \leq 4$)
  D) y out of range ($-5 > y > 4$)

All possible combinations





(A,C), (A, D), (B,C), (B, D)