* 1. For moving from testing level 2 to testing level 4, there are several factors:
* People in testing are good and having technical kwowledge.
* Good connection between developers and testers.
* Good management leadership.

1.3 Level 3 thinking is that pupose of testing is to reduce risk means:

* Whenever we use software, we incur some risk. The risk may be small impacct or consequence catastrophic. It makes entire development team want to reduce risk. However, we have to accept that risks are always there, we can’t reduct risk to 0.

1.4

a)

*# Python3 code to demonstrate list  
# concatenation using naive method  
  
# Initializing lists*test\_list1 = []  
test\_list2 = []  
*#Input list*for i in range (0,5):  
 temp1 = input(**"Input list 1: "**)  
 test\_list1.append(temp1)  
for i in range (0,5):  
 temp2 = input(**"Input list 2: "**)  
 test\_list2.append(temp2)  
*# using naive method to concat*for i in test\_list2:  
 test\_list1.append(i)  
  
*# Printing concatenated list*print(**"Concatenated list: "** + str(test\_list1))

b, c) Some faults exist:

- User want to input each list x elements, however, it’s fixed at 5 elements now.

d) I can update to be:

*# Python3 code to demonstrate list  
# concatenation using naive method  
  
# Initializing lists*test\_list1 = []  
test\_list2 = []  
*#No of Elements*no1 = int(input(**'How many elements in list 1: '**))  
no2 = int(input(**'How many elements in list 2: '**))  
*#Input list*for i in range (0,no1):  
 temp1 = input(**"Input list 1: "**)  
 test\_list1.append(temp1)  
for i in range (0,no2):  
 temp2 = input(**"Input list 2: "**)  
 test\_list2.append(temp2)  
*# using naive method to concat*for i in test\_list2:  
 test\_list1.append(i)  
  
*# Printing concatenated list*print(**"Concatenated list: "** + str(test\_list1))

1.5

-findLast solution

a) The loop must start with 0 index:

for (int i=x.length-1; i>=0; i--)

b) if x = null, y = 7, before the loop, it will result in NullPointerException

c) x = [5,6,7]; y=6, return 1, because y = 6 is in index 1, there is no error

d) x = [5,6,7]; y=8, return -1, because 8 is not in x

e) x [5,6,5]; y=5; expected 2, return 0, because i has valeu 0, not at the end of array

- lastZero solution:

a) the loop shoue be:

for (int i=x.length-1; i >=0; i--)

b) All input will result in fault

c) If x=[1], return -1, input has length 0 or 1, there is no error

d) If x=[2,0,4], return 1, there is an error when array has more than 1 element, sice the values if index increase

e) If x=[0,1,0], expected 2, return 0, index i has value 0, however it should be at the end of the array

-countPositive solution

a)Test condition should be:

if (x[i] > 0)

b) If x = [], return 0, because we give empty array array here

c) If x = [2,3,4], return 3, because it not contain 0

d) Every array contain 0 will lead to error, and lead to failure

e) if x = [-5, 0, 1, 2], expected 2, return 3, because it contain 0

-oddOrPos solution

a) Condition should be:

if (x[i]%2 == -1 || x[i] > 0)

b) if x[], return 0 , because array is empty

c) if x=[1,2,3], return 3, because array doesn’t contain negative value

d) If array contain negative value, it will result in error and failure, regardless the rest are positive

e) if x=[-1, 0, 1, 2], expected 3, return 2, because it not count -1 value

2.2

I haven’t work with company, I work with my friend at university. We have some project as final examination of subject. We are not good in testing. We wrote the source code, I ran to test with input. It can’t cover all failure, we can’t expected wether having failure in the future. We also don’t have plan from the beginning, the do and update the program, it’s waste a lot of time, and i feel tired of not having strategy.