**Homework Assignment 3** (Due Day: June 1, 2017)

**Name:** Princess Kim **PSU\_ID:** 967760113 **Email:** prikim@pdx.edu

**Submission Date:** 6/1/17 **Late Days Used:** 0

**Collaboration Acknowledgment:** Eunji Lee, Chris M

# Solutions of 6.11

A: N is a positive integer

At B and C

B: M = 1

C: M = 1 and q = 2

An expression needs to be provided so that at point D it knows if the loops has been executed 0, 1, or many times.

D: assume that it holds at point D for m equal to some value. Because of the hypothesis, it follows that. Which implies that

H: it is the output specification.

However, if the control passes from point D to E then

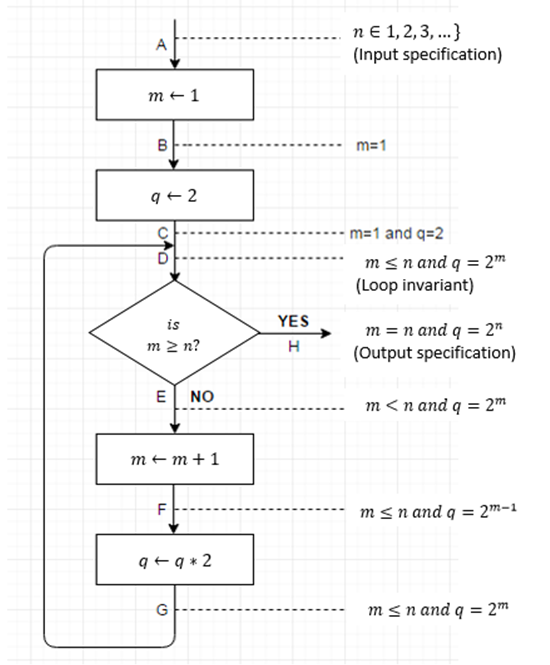
E: the statement is executed, so at point F the following happens

F: the next statement is to be executed. Multiplying by 2 is the following assertion at point G

G: the assertion that holds at G is identical to those at point G, but if point D has m=, then it again will hold at point D with m=. By induction, it follows that loop at point D for all values m.

All that remains is to prove that the loop terminates. Initially by assertion at point C, the value of m is equal to 1. Each iteration of the loop increases the value of m by 1 when the statement is executed. Eventually, m must reach the value n at which point the loop is executed and the value of q is given by assertion at point H, thereby satisfying the ouput specification.

Flow chart below for reference



# Solutions of 6.14

static public Void Naur(String input){

int j = 0;

int lim = input.length()/maxPos;

String[] strings = new String [lim+3];

int temp = maxPos;

String temp1 = "";

for (int i = 0; i <= lim+1; ++i)

{

if ( j+temp+1 < input.length())

temp1 = input.substring(j+temp, (j+temp +1));

while ((temp1.compareTo("\n") != 0) && (temp1.compareTo(" ") != 0))

{

temp = temp - 1;

temp1 = input.substring(j+temp-1, (j+temp ));

}

//strings[i] =

if (j+temp < input.length()) {

System.out.println(input.substring(j, j + temp));

}else{

System.out.println(input.substring(j, input.length()));

}

j += temp;

temp = maxPos;

}

Faults:

1. Leaves a space at the beginning or end of the line
2. It can have double breaks with new line

# Solutions of 15.15

Input expected outcome

1. blank newline
2. newline newline
3. single char char newline
4. string (a word) if length <= maxpos, then string and new line
5. char blank string if length + length <= maxpos, str blank str newline
6. char blank string if length <= maxpos & length + length > maxpos,

char newline str newline

# Solutions of 15.16

By running the program with test cases of long string swith new lines and spaces will test the run through each of the branches. The program will be tested with strings being no longer than the maxPos, whether it will break at space or newline, and that each line is filled as long as possible.

# Solutions of 15.17

if ( j+temp+1 < input.length())

temp1 = input.substring(j+temp, (j+temp +1));

Test case j = 0, temp = 4, input = banana temp1 = banana

  while ((temp1.compareTo("\n") != 0) && (temp1.compareTo(" ") != 0))

      {

          temp = temp - 1;

          temp1 = input.substring(j+temp-1, (j+temp ));

      }

Temp1 = banana, temp = 4, j = 0   => temp = 3 temp1 = na

Temp1 = na, temp = 3, j = 0 => temp = 2 temp1 = an

Temp1 = an, temp = 2, j = 0 => temp = 1 temp1 = ba

Temp1 = ba, temp = 1, j =0 => temp = 0 temp1 = b

Temp 1 = b, temp = 0, j = 0 => temp = -1 temp1 = b

Temp 1 = b, temp = -1, j = 0 => temp = -2 temp1 = “ “

if (j+temp < input.length())

System.out.println(input.substring(j, j + temp));

j = 0, temp = -1, input = banana prints: b

# Solutions of 15.18

1. Input = “ “, j = 0, maxPos = 5 Error (out of bounds)
2. Input = “a”, j = 0, maxPos = 1 Acceptable
3. Input = “ba”, j = 0, maxPos = 2 Acceptable
4. Input = “ba”, j = 1, maxPos = 2 Error (out of bounds)
5. Input = “\n”, j = 0, maxPos = 0 Error (can’t split string)

# Solutions of 15.19

A. If j is less than input’s length

1. Input = “ “ Exit program
2. Input.length > 0 Enter loop
3. If false Exit Program
4. If true Enter loop

B. While input(j + temp) is an empty spot (“ “ or “\n”)

1. If true --temp

If false print, change j and temp, go back to A

# C:\Users\Princess\Desktop\Naur flow chart.pngSolutions of 15.20