**Please, upload your answers in Canvas: assignment PC LAB – April 9**

Name: Vu The Doan

Student nr: 12918687

PC2b

**1**. Write a script beautyofmath that produces the following output (without strings, using numbers only!):

> source('beautyofmath')

1 x 8 + 1 = 9

12 x 8 + 2 = 98

123 x 8 + 3 = 987

1234 x 8 + 4 = 9876

12345 x 8 + 5 = 98765

123456 x 8 + 6 = 987654

1234567 x 8 + 7 = 9876543

12345678 x 8 + 8 = 98765432

123456789 x 8 + 9 = 987654321

- So, unlike exercise 1 of the previous PC lab, use numbers instead of strings!

- Note: 12=1\*10+2, 123=12\*10+3, etc.

- You can use either cat() or cat(sprintf(...)) for displaying.

**2.a** Toward the end of World 1-1 in Nintendo’s Super Mario Brothers, Mario must ascend right-aligned pyramid of blocks, a la the screenshot below.



Let’s recreate that pyramid in a function called pyramid using hashes (#) for bricks. The function takes as argument the height of the pyramid. For instance,

> pyramid(5)

#

##

###

####

#####

The function should first check if the height of the pyramid is between 1 and 9 (bounds included). Hint: use two nested for loops.

b. Next, we want to take it to the next level! Copy your function pyramid and modify it to display a two-sided pyramid.



Example output:

> pyramid2(6)

# #

## ##

### ###

#### ####

##### #####

###### ######

**3**. In the 20×20 grid below, four numbers along a diagonal line have been marked in red.

08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08  
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00  
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65  
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91  
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80  
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50  
32 98 81 28 64 23 67 10 **26** 38 40 67 59 54 70 66 18 38 64 70  
67 26 20 68 02 62 12 20 95 **63** 94 39 63 08 40 91 66 49 94 21  
24 55 58 05 66 73 99 26 97 17 **78** 78 96 83 14 88 34 89 63 72  
21 36 23 09 75 00 76 44 20 45 35 **14** 00 61 33 97 34 31 33 95  
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92  
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57  
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58  
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40  
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66  
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69  
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36  
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16  
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54  
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48

The product of these numbers is 26 × 63 × 78 × 14 = 1788696.

What is the greatest product of four adjacent numbers in the same direction (up, down, left, right, or diagonally) in the 20×20 grid?

The aim of the problem is coding a method of scanning elements in a two-dimensional matrix. Let x be the given matrix of size and @ be the cell , . We need to take the products of four adjacent numbers from @ along to the four directions as follows:

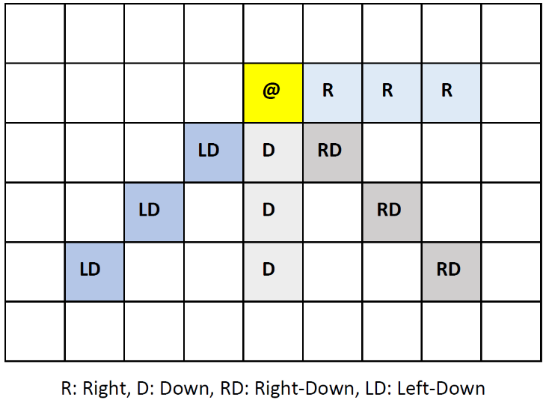
Right: ;

Down:;

Right-Down: ;

Left-Down:.

At each step we get the maximum product and assign it to a variable pmax.



Of course, we need to control the range of the indexes and . For example, if we scan from @ = then only two directions Down and Left-Down will be considered; Right and Right-Down will be ignored.

Hints:

- write four functions Right, Down, RightDown and LeftDown. The function Right looks like this:

Right <- function(x,i,j,pmax){

p=x[i,j]\*x[i,j+1]\*x[i,j+2]\*x[i,j+3]

if(p>pmax) pmax<-p

return(pmax)

}

- Initialize pmax to zero

- Use two for-loops: one for i and a second inner loop j

- Check if the coordinates i and j satisfy the constrains to call the function. For instance,

if (j<18) pmax<-Right(x,i,j,pmax)

- Show pmax at the end of the program

- Copy the lines below for the matrix!

vec=c(08,02,22,97,38,15,00,40,00,75,04,05,07,78,52,12,50,77,91,08,

49,49,99,40,17,81,18,57,60,87,17,40,98,43,69,48,04,56,62,00,

81,49,31,73,55,79,14,29,93,71,40,67,53,88,30,03,49,13,36,65,

52,70,95,23,04,60,11,42,69,24,68,56,01,32,56,71,37,02,36,91,

22,31,16,71,51,67,63,89,41,92,36,54,22,40,40,28,66,33,13,80,

24,47,32,60,99,03,45,02,44,75,33,53,78,36,84,20,35,17,12,50,

32,98,81,28,64,23,67,10,26,38,40,67,59,54,70,66,18,38,64,70,

67,26,20,68,02,62,12,20,95,63,94,39,63,08,40,91,66,49,94,21,

24,55,58,05,66,73,99,26,97,17,78,78,96,83,14,88,34,89,63,72,

21,36,23,09,75,00,76,44,20,45,35,14,00,61,33,97,34,31,33,95,

78,17,53,28,22,75,31,67,15,94,03,80,04,62,16,14,09,53,56,92,

16,39,05,42,96,35,31,47,55,58,88,24,00,17,54,24,36,29,85,57,

86,56,00,48,35,71,89,07,05,44,44,37,44,60,21,58,51,54,17,58,

19,80,81,68,05,94,47,69,28,73,92,13,86,52,17,77,04,89,55,40,

04,52,08,83,97,35,99,16,07,97,57,32,16,26,26,79,33,27,98,66,

88,36,68,87,57,62,20,72,03,46,33,67,46,55,12,32,63,93,53,69,

04,42,16,73,38,25,39,11,24,94,72,18,08,46,29,32,40,62,76,36,

20,69,36,41,72,30,23,88,34,62,99,69,82,67,59,85,74,04,36,16,

20,73,35,29,78,31,90,01,74,31,49,71,48,86,81,16,23,57,05,54,

01,70,54,71,83,51,54,69,16,92,33,48,61,43,52,01,89,19,67,48)

x=matrix(vec,20,20)

What is the greatest product of four adjacent numbers in the same direction (up, down, left, right, or diagonally) in the 20×20 grid?

**Answer**:

R-code:

R-code:

getwd()

setwd("D:/UvA/Year 1/Block 5/Programming and Numerical Analysis/Week 2")

rm(list = ls())

#1

xstr <- 0

i <- 1

ystr <- 0

for(i in c(1:9)) {

xstr <- xstr\*10 + i

ystr <- ystr\*10 + (10-i)

cat(sprintf("%s x 8 + %d = %s\n", xstr, i, ystr))

}

#2

pyramid <- function(x) {

if(x <= 9){

A <- matrix(,nrow = x,ncol = x)

for(i in 1:x) {

A[i, c((x+1-i):x)] <- "#"

for(j in 1:x) {

A[i, c((x+1-i):x)] <- "#"

return(A)

}

}

}

}

pyramid(5)

#3

vec=c(08,02,22,97,38,15,00,40,00,75,04,05,07,78,52,12,50,77,91,08,

49,49,99,40,17,81,18,57,60,87,17,40,98,43,69,48,04,56,62,00,

81,49,31,73,55,79,14,29,93,71,40,67,53,88,30,03,49,13,36,65,

52,70,95,23,04,60,11,42,69,24,68,56,01,32,56,71,37,02,36,91,

22,31,16,71,51,67,63,89,41,92,36,54,22,40,40,28,66,33,13,80,

24,47,32,60,99,03,45,02,44,75,33,53,78,36,84,20,35,17,12,50,

32,98,81,28,64,23,67,10,26,38,40,67,59,54,70,66,18,38,64,70,

67,26,20,68,02,62,12,20,95,63,94,39,63,08,40,91,66,49,94,21,

24,55,58,05,66,73,99,26,97,17,78,78,96,83,14,88,34,89,63,72,

21,36,23,09,75,00,76,44,20,45,35,14,00,61,33,97,34,31,33,95,

78,17,53,28,22,75,31,67,15,94,03,80,04,62,16,14,09,53,56,92,

16,39,05,42,96,35,31,47,55,58,88,24,00,17,54,24,36,29,85,57,

86,56,00,48,35,71,89,07,05,44,44,37,44,60,21,58,51,54,17,58,

19,80,81,68,05,94,47,69,28,73,92,13,86,52,17,77,04,89,55,40,

04,52,08,83,97,35,99,16,07,97,57,32,16,26,26,79,33,27,98,66,

88,36,68,87,57,62,20,72,03,46,33,67,46,55,12,32,63,93,53,69,

04,42,16,73,38,25,39,11,24,94,72,18,08,46,29,32,40,62,76,36,

20,69,36,41,72,30,23,88,34,62,99,69,82,67,59,85,74,04,36,16,

20,73,35,29,78,31,90,01,74,31,49,71,48,86,81,16,23,57,05,54,

01,70,54,71,83,51,54,69,16,92,33,48,61,43,52,01,89,19,67,48)

x=matrix(vec,20,20)

Right <- function(x,i,j,pmax){

p=x[i,j]\*x[i,j+1]\*x[i,j+2]\*x[i,j+3]

if(p>pmax) pmax<-p

return(pmax)

}

Down <- function(x,i,j,pmax){

p=x[i,j]\*x[i+1,j]\*x[i+2,j]\*x[i+3,j]

if(p>pmax) pmax<-p

return(pmax)

}

RightDown <- function(x,i,j,pmax){

p=x[i,j]\*x[i+1,j+1]\*x[i+2,j+2]\*x[i+3,j+3]

if(p>pmax) pmax<-p

return(pmax)

}

LeftDown <- function(x,i,j,pmax){

p=x[i,j]\*x[i+1,j-1]\*x[i+2,j-2]\*x[i+3,j-3]

if(p>pmax) pmax<-p

return(pmax)

}

pmax <- 0

for(i in 1:20) {

for(j in 1:20) {

}

}