Alexandrescu Andra, gr 232

#Q: Daca vreau sa le sterg pe toate cum procedez?

rm(list = objects())

objects()

#Q: Explicati de ce nu merge urmatoarea atribuire multipla

d2+d4 -> d6 <-d1

#nu merge deoarece d6 poate sa fie asignat cu valori pe rand si nu simultan, nefiind cunoscuta prioritatea asignarii (prioritati necunoscute pentru d2+d4 si d1)

#Q: Explicati rolul parantezelor si al operatorului : in urmatoarea situatie

1:n-1

1:(n-1)

2\*1:5

2\*1+1:5

2+1:5

(2\*1+1):5

30:3

#e ilustrata prioritatea operatiilor de la mare la mica : > \* > +. parantezele obliga evaluarea expresiei care are o operatie cu o prioritate mai mica, spre exemlplu diferenta de rezultat intre liniile 50 si 48

#: descrie o secventa care incepe cu elem din st si se termina cu cel din dreapta (inclusiv), incrementarea/ decrementarea pentru fiecare valoare fiind 1 (in cazul in care nu apare operatorul \*); decrementarea are loc numai daca val din st > val din dr

#Q: Explicati comportamentul operatorului :

1.4:5

#in acest caz secventa porneste de la 1 si se termina pana la valoarea lui 5 (strict mai mic), fiind o valoare cu .4

#Q: Explicati urmatoarea secventa de cod

(b <- c(rep(rep(c(0,2),3),1:6),rep(rep(c(0:3,6),c(2,4:7)),24)))

rep(c(0,2),3) #repeta vectorul 0 2 de 3 ori: 0 2 0 2 0 2

rep(rep(c(0,2),3),1:6) #fiecare elem de mai sus e repetat de 1, 2, 3, 4, 5, respectiv 6 ori

# 0 2 2 0 0 0 2 2 2 2 0 0 0 0 0 2 2 2 2 2 2

c(0:3,6) #la elem din intervalul [0, 3] se adauga 6 la final: 0 1 2 3 6

c(2,4:7) #se adauga 2 in fata intervalului [4,7]: 2 4 5 6 7

rep(c(0:3,6),c(2,4:7)) #fiecare elem de la c(0:3,6) e repetat de 2, 4, 5, 6, respectiv 7 ori

# 0 0 1 1 1 1 2 2 2 2 2 3 3 3 3 3 3 6 6 6 6 6 6 6

rep(rep(c(0:3,6),c(2,4:7)),24) #secventa de sus e repetata de 24 de ori

#Q: Explicati urmatoarea secventa de cod

x <- sample(1:1000,100) # valorile intre 1 si 1000 randomised, in total fiind 100 de astfel de val

x[x%%8==0][1:3] # primele 3 val care dau 0 la impartirea cu 8 x[conditie][interval indici cu val cu ac conditie]

#Q: Explicati urmatoarea secventa de cod

x\_1 <- x[-(1:3)] # toate val lui x cu exceptia primelor trei valori

#Q: Explicati ce face urmatoarea comanda

c1 <- a1+b # se aduna fiecare elem cu acelasi indice din vectorii a1 si b, iar pt ca lungimea b e mai mica ca cea a lui a1, deci cicleaza indicii de la b, continuandu-se cu restul elementelor din a1 (pozitiile 4, 5 si 6 ca indici nu au val coresp pt b, dar 4 = 3\*1+1, 5 = 3\*1+2, 6 = 3\*2, asadar c1[indice>3] = b[3n+k]+a1[indice], daca indice de forma 3n+k, k=0,1,2)

# Q: De ce vector\_mixt este in continuare un vector si nu o lista?

#elem sunt transformate intr-un tip general, pt ca vectorul e format dintr-un singur tip de elem (date omogene)

#sunt creati de obicei prin evaluarea unor expresii logice

x <- sample(1:100,10)

y <- (x[1:50]<30)

class(y) #vector cu val de adevar pt fiecare val daca respecta sau nu conditia < 30

(y)

#Q: Ce este in neregula cu expresia de mai sus?

#y presupune ideea ca x are mai mult decat 50 de elemente, insa lungimea vectorului x este de 10, de aceea apar val de NA pentru elem cu indicii de la 11 la 50

#Q: Cum putem elimina in mod automat valorile NA dintr-un vector?

#HINT: is.na()

#!is.na(vector) inlocuieste toate val cu TRUE, iar is.na() cu FALSE, deci trebuie sa fie o conditie pentru care se aleg elementele vectorului x

y <- (x[!is.na(x)]<30)

(y)

#Q: Creati un vector logic t ce compara daca elementul de pe pozitia i(impara)

# este mai mic decat elementul de pe pozitia para imediat urmatoare

2\*0:(length(x))+1

2\*1:length(x)

x[2\*0:(length(x)/2-1)+1] # impartire lungime vector x la doi pentru a scapa de valorile de NA

#var 2: x[!is.na(x[2\*0:length(x)+1])]

x[2\*1:(length(x)/2)]

vec <- (x[2\*0:(length(x)/2-1)+1]<x[2\*1:(length(x)/2)])

(vec)

#Q: Determinati pozitiile pentru care conditia de la prima intrebare este adevarata

#HINT: which()

vec\_pos <- which(vec)

(vec\_pos)

#Q: Determinati cate numere cuprinse intre 411 si 7870 sunt divizibile cu 9

#dar nu cu 5

num <- 411:7870

nr <- length(num[num%%9==0 & num%%5!=0])

(nr)

#To do

# Construiti doi vectori x si y cu 1000 de elemente fiecare, extrase in mod

#aleator din multimea cu numere intregi -24500:76000.

x <- sample(-24500:76000, 1000)

y <- sample(-24500:76000, 1000)

#a)Stabiliti care dintre cei doi vectori are mai multe elemente,

#luate in valoare absoluta, mai mari decat valoarea absoluta a elementului

#corespondent din celalalt vector

x\_y <- c(x, y)[abs(x)>abs(y)]

y\_x <- c(x, y)[abs(x)<abs(y)]

nr\_x\_y <- sum(abs(x)>abs(y))

nr\_y\_x <- sum(abs(x)<abs(y))

if(nr\_x\_y>nr\_y\_x)

{

print(cat("x cu ", nr\_x\_y, " elem mai mari castiga"))

} else if(nr\_x\_y<nr\_y\_x)

{

print(cat("y cu ", nr\_y\_x, " elem mai mari castiga"))

}

#b)Stabiliti care dintre cei doi vectori are minimul pe o pozitie mai mare

x\_min\_pos <- which.min(x)

y\_min\_pos <- which.min(y)

if(x\_min\_pos<y\_min\_pos)

{

print(cat("Pozitia minima e in vect x: cu val ", min(x), ", poz ", x\_min\_pos))

} else if(x\_min\_pos>y\_min\_pos)

{

print(cat("Pozitia minima e in vect y: cu val ", min(y), ", poz ", y\_min\_pos))

}

#c)Stabiliti care dintre cei doi vectori are cele mai multe valori care se repeta

# vector de frecvente folosind table(); perechi val: aparitii\_val

table(x)

table(y)

max\_freq\_x = max(table(x))

max\_freq\_y = max(table(y))

if(max\_freq\_x<max\_freq\_y)

{

print(cat("Maximul de repetii ale unei val apartine vect y, ea fiind ", max\_freq\_y))

} else if(max\_freq\_x>max\_freq\_y)

{

print(cat("Maximul de repetii ale unei val apartine vect x, ea fiind ", max\_freq\_x))

}

#d)Stabiliti care dintre cei doi vectori are o secventa de cel putin 2 valori consecutive

if(any(diff(sort(x))==1))

{

print("Vectorul x are cel putin 2 val consecutive")

}

if(any(diff(sort(y))==1))

{

print("Vectorul y are cel putin 2 val consecutive")

}

#e)Stabiliti care dintre cei doi vectori are mai multe valori divizibile cu corespondentele

#lor din celalalt vector.

k\_x\_div <- sum(abs(x)%%abs(y)==0 & y!=0)

k\_y\_div <- sum(abs(y)%%abs(x)==0 & x!=0)

if(k\_x\_div>k\_y\_div)

{

print("x are mai multe val divizibile")

} else if(k\_x\_div<k\_y\_div)

{

print("y are mai multe val divizibile")

}