Technologie Informacyjne

R – zadania

Opracował: Maciej Penar

Spis treści

[1.Zadania 3](#_Toc88843339)

# 1.Zadania

## Hello world

1. Uruchomić I przeanalizować poniższe programy:

Program 1:

|  |
| --- |
| print("Hello World!") |

Program 2:

|  |
| --- |
| for(index in 1:5){  print(index)  } |

Program 3:

|  |
| --- |
| for(index in 100:95){  print(index)  } |

**Na ich podstawie napisać skrypt który odlicza czas do wystartowania rakiety. Program powinien wypisać:**

|  |
| --- |
| 5  4  3  2  1  STARTUJEMY! |

1. Uruchomić I przeanalizować poniższe programy:

Program 1 – akcent na **paste**:

|  |
| --- |
| x = runif(1)  paste("Zmienna `x` ma wartość:", x) |

Program 2:

|  |
| --- |
| for(index in 1:10){  if(index %% 2 == 0){  print(paste("Wartość index` jest podzielna przez 2:", index))  }  } |

Program 3:

|  |
| --- |
| for(index in 1:10){  if(index > 5){  print(paste("Wartość index` jest większa niż 5:", index))  }  } |

**Na ich podstawie napisać skrypt który odlicza czas do wystartowania rakiety od 60 sekund – wyświetlając każde upływające 10 sekund w pełnych komunikatach oraz ostatnie 5 sekund w niepełnym komunikacie. Program powinien wypisać:**

|  |
| --- |
| 60 sekund do startu rakiety  50 sekund do startu rakiety  40 sekund do startu rakiety  30 sekund do startu rakiety  20 sekund do startu rakiety  10 sekund do startu rakiety  5..  4..  3..  2..  1..  STARTUJEMY! |

## Arytmetyka

1. Uruchomić oraz przeanalizować poniższe programy:

Program 1:

|  |
| --- |
| x = 10 # zapamiętanie jakiejś wartości pod nazwą `x`  x |

Program 2:

|  |
| --- |
| x = 3  y = 5  x + y  x \* y  x \*\* y  x / y  x - y |

Program 3:

|  |
| --- |
| x = 10  y = 15  z = 2 \* x + y  z |

Program 4:

|  |
| --- |
| pi # stała pi  exp(1) # exponent |

Na ich podstawie wyliczyć obwody wszystkich kół o promieniach 1,2,3…100 (centymetrów).

1. Uruchomić oraz przeanalizować poniższe programy:

Program 1 (wyliczenie pól kół o promieniach 1-15):

|  |
| --- |
| radius = 1:15  area = pi\*(radius\*\*2)  area  area[3] # wybranie 3-ciego pola |

Program 2:

|  |
| --- |
| radius = 1:15  area = pi\*(radius\*\*2)  c(area[3], area[10], area[5])  # Utworzenie wektora złożonego z 3-ciego, 10-tego oraz 5-tego elementu |

Utworzyć wektor składający się z: wybranej przez siebie wartości promienia, długości obwodu okręgu o podanym promieniu oraz pola koła o podanym promieniu.

1. Uruchomić oraz przeanalizować poniższe programy:

Program 1 – operatory logiczne

|  |
| --- |
| 1==1 # równoważność  1==0  1!=1  1!=0  1>0  1<0  1> 0 & 1==1  1> 0 & 1!=1  1>0 | 1==1   1. | 1!=1   !TRUE  !(1==1) |

Program 2 – utworzenie funkcji:

|  |
| --- |
| deMorgan = function(p,q){  !(p & q)==(!p|!q)  }  deMorgan(TRUE,TRUE)  deMorgan(TRUE,FALSE)  deMorgan(FALSE,TRUE)  deMorgan(FALSE,FALSE) |

Napisać funkcję która sprawdza czy wartościuje prawo deMorgana: 

1. Napisać program który sprawdza czy podane wyrażenia są tautologiami:

* (p ⇔ q) ⇒ (p ∧ q),
* (p ⇔ q) ⇒ [(∼ p ∨ q) ∧ (p∨ ∼ q)],
* (p ⇒ q) ⇒ [p ⇒ (q ∨ r)].

*Podpowiedź: Implikacja jako tako nie ma żadnego operatora, bo wyrażenie p ⇒ q można ją zastąpić wyrażeniem ∼p ∨ q, czyli de facto funkcją:*

|  |
| --- |
| implies = function(p,q){  !p | q  }  implies(TRUE,TRUE)  implies(TRUE,FALSE)  implies(FALSE,TRUE)  implies(FALSE,FALSE) |

1. Napisać funkcję zwracająca wartość bezwzględną podanej liczby
2. ! Napisać funkcję zwracającą wartość n-tego wyrazu funkcji fibonacciego:

## Rysowanie

1. Uruchomić oraz przeanalizować poniższy program:

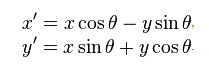
|  |
| --- |
| x = 0:10  y = 3 \* x + 1  # Linijka poniżej dla wybranych osi X/Y  # plot(x, y, type = "l", xlim = c(0,10), ylim=c(0,30))  plot(x, y, type = "l") |

Lub jeśli lubimy funkcje:

|  |
| --- |
| x = 0:10  f = function(x){ 3 \* x + 1 }  plot(x, f(x), type = "l") |

**Na jego podstawie narysować wykres funkcji: -5x + 2**

1. Narysować wykres funkcji: 1/x w przedziale -20:20
2. Narysować wykres funkcji kwadratowej + 5
3. Narysować wykres sinusa: funkcja **sin**
4. **! Narysować okrąg o promieniu r**. Żeby to wykonać należy wyznaczyć punkty należące do tego okręgu. Najlepiej zabrać się za to poprzez obrót punktu który wiemy że należy do okręgu np. [0,r] (lub [r,0] lub [0,-r] lub [-r,0]) i obrót tego punktu o pewien kąt wg. wzoru:

[](https://i.stack.imgur.com/oE6No.png)

|  |
| --- |
| pointsX = function(theta, r){  output = c()  point = c(0,r)  for(i in 1:100){  # Tu wyliczyć nowe położenie punktu x,y wg. punktu `point`  # x = ?  # y = ?  point = c(x, y)  output = c(output, x)  }  output  }  pointsY = function(theta, r){  output = c()  point = c(0,r)  for(i in 1:100){  # Tu wyliczyć nowe położenie punktu x,y wg. punktu `point`  # x = ?  # y = ?  point = c(x, y)  output = c(output, y)  }  output  }  px = pointsX(pi/50, 10)  py = pointsY(pi/50, 10)  plot(px,py) |