

## Άσκηση 1

Φτιάξτε μία συνάρτηση σε R η οποία θα παίρνει 2 παράμετρους:

- BMI : ένας ακέραιος με το BMI (Body Mass Index)
- gender: Ένα αλφαριθμητικό το οποίο θα είναι είτε "male" είτε "female"

Αν το gender είναι male, τότε αν το BMI  $\leq 25$  θα πρέπει να επιστρέφει "normal" αλλιώς θα επιστρέφει "overweight"

Αν το gender είναι female, τότε αν το BMI  $\leq 24$  τότε θα πρέπει να επιστρέφει "normal" αλλιώς θα επιστρέφει "overweight"

```
In [5]: ask1 <- function(bmi, gender) {  
  if (gender == "male") {  
    if (bmi <= 25) {  
      return ("normal")  
    }  
    return ("overweight")  
  } else if (gender == "female") {  
    if (bmi <= 24) {  
      return ("normal")  
    }  
    return ("overweight")  
  }  
  else {  
    return ("error")  
  }  
}
```

```
In [6]: ask1(26, 'male')
```

'overweight'

```
In [9]: ask1(25, 'male')
```

'normal'

```
In [10]: ask1(25, 'female')
```

'overweight'

```
In [11]: ask1(24, 'female')
```

'normal'

## Άσκηση 2

Σύμφωνα με αυτό το [site \(https://alen.space/basic-guide-nanosatellites/\)](https://alen.space/basic-guide-nanosatellites/) οι μικροί δορυφόροι μπορούν να κατηγοριοποιηθούν ανάλογα με το βάρος τους σε αυτές τις κατηγορίες:

- Minisatellite: 100-500 kg
- Microsatellite: 10-100 kg
- Nanosatellite: 1-10 kg
- Picosatellite: Less than 1 kg

Φτιάξτε μία συνάρτηση σε R η οποία θα παίρνει ως παράμετρο το βάρος ενός δορυφόρου και θα επιστρέφει το όνομα της κατηγορίας που ανήκει. Θεωρήστε ότι:

- Πάνω από 500 κιλά επιτρέπει "error"
- 500 κιλά --> Minisatellite
- 100 κιλά --> Minisatellite
- 10 κιλά --> Microsatellite
- 1 κιλό: --> Nanosatellite

```
In [12]: ask2 <- function(kg) {  
  if (kg>500) {  
    return ("error")  
  }  
  if (kg >= 100) {  
    return ("Minisatellite")  
  }  
  if (kg >= 10) {  
    return ("Microsatellite")  
  }  
  return ("Nanosatellite")  
}
```

```
In [14]: ask2(500.0001)  
  
'error'
```

```
In [15]: ask2(500)  
  
'Minisatellite'
```

```
In [16]: ask2(100)  
  
'Minisatellite'
```

```
In [17]: ask2(10)  
  
'Microsatellite'
```

```
In [18]: ask2(9.999)  
  
'Nanosatellite'
```

### Άσκηση 3

Δίνεται αυτή η συνάρτηση η οποία επιστρέφει TRUE/FALSE ανάλογα με το αν ένας αριθμός είναι πρώτος ή όχι:

```
In [19]: prime <- function(n) {  
  for (i in 2:(n-1)) {  
    if (n%%i == 0) {  
      return (FALSE)  
    }  
  }  
  
  return (TRUE)  
}
```

Ποιος είναι ο μεγαλύτερος πρώτος αριθμός που είναι μικρότερος του 1000;

1ος τρόπος:

```
In [20]: n <- 999  
while (!prime(n)) {  
  n <- n - 1  
}  
print (n)  
  
[1] 997
```

2ος τρόπος (αργός!)

```
In [21]: for (n in 1:999) {  
  if (prime(n)) {  
    d <- n  
  }  
}  
print (d)  
  
[1] 997
```

### Άσκηση 4

Ρίχνω μία μπάλα από ύψος 1m. Κάθε φορά που αναπηδάει στο έδαφος χάνει το 10% του ύψους της. Μετά από πόσες αναπηδήσεις το ύψος που θα φτάσει θα είναι κάτω από 0.5m;

```
In [22]: h <- 1
c <- 0
while (h>=0.5) {
  h <- h - (0.1*h)
  c <- c + 1
}
print (c)

[1] 7
```

## Άσκηση 5

Φτιάξτε τον παρακάτω πίνακα:

```
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
```

```
In [23]: matrix(1:5, nrow=5, ncol=5)
```

A matrix: 5 × 5 of  
type int

```
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
```

## Άσκηση 6

Φτιάξτε τον παρακάτω πίνακα:

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

```
In [25]: matrix(1:5, nrow=5, ncol=5, byrow=TRUE)
```

A matrix: 5 × 5 of  
type int

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

## Άσκηση 7

Φτιάξτε τον παρακάτω πίνακα:

0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0

1ος τρόπος:

```
In [39]: m <- matrix(0, ncol=10, nrow=10)

for (i in 1:10) {
  m[i,11-i] <- 1
}
print (m)
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	0	0	0	0	0	0	0	0	0	1
[2,]	0	0	0	0	0	0	0	0	1	0
[3,]	0	0	0	0	0	0	0	1	0	0
[4,]	0	0	0	0	0	0	1	0	0	0
[5,]	0	0	0	0	0	1	0	0	0	0
[6,]	0	0	0	0	1	0	0	0	0	0
[7,]	0	0	0	1	0	0	0	0	0	0
[8,]	0	0	1	0	0	0	0	0	0	0
[9,]	0	1	0	0	0	0	0	0	0	0
[10,]	1	0	0	0	0	0	0	0	0	0

2ος τρόπος:

```
In [41]: m <- matrix(0, ncol=10, nrow=10)
s <- 10
for (i in 1:10) {
  m[s] <- 1
  s <- s + 9
}
print (m)
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	0	0	0	0	0	0	0	0	0	1
[2,]	0	0	0	0	0	0	0	0	1	0
[3,]	0	0	0	0	0	0	0	1	0	0
[4,]	0	0	0	0	0	0	1	0	0	0
[5,]	0	0	0	0	0	1	0	0	0	0
[6,]	0	0	0	0	1	0	0	0	0	0
[7,]	0	0	0	1	0	0	0	0	0	0
[8,]	0	0	1	0	0	0	0	0	0	0
[9,]	0	1	0	0	0	0	0	0	0	0
[10,]	1	0	0	0	0	0	0	0	0	0

## Άσκηση 8

Φτιάξτε μία λίστα που να περιέχει το όνομα (αλφαριθμητικό), το επίθετο (αλφαριθμητικό) ενός μαθητή, τους βαθμούς του (ένα διάνυσμα από 5 τυχαίες αριθμητικές τιμές).

```
In [43]: l <- list(
  name="mitsos",
  surname="katsouranis",
  grades=c(5,7,4,3,2)
)
print (l)
```

```
$name
[1] "mitsos"

$surname
[1] "katsouranis"

$grades
[1] 5 7 4 3 2
```

Αντικαταστήστε το πεδίο "grades" της προηγούμενης λίστας με ένα πίνακα 3 γραμμές X 5 στήλες με τυχαίες τιμές από το 1 μέχρι τι 20.

```
In [45]: l$grades <- matrix(runif(15, 1, 20), nrow=3, ncol=5)
```

```
In [46]: print(l)
```

```
$name
```

```
[1] "mitsos"
```

```
$surname
```

```
[1] "katsouranis"
```

```
$grades
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	11.096912	16.880288	19.84811	11.703675	14.46216
[2,]	14.136438	6.839001	16.66767	6.278271	16.63007
[3,]	9.824791	6.306368	12.17413	9.439518	19.38922