

ΠΡΑΚΤΙΚΟ 6 - PYTHON

Επιλύστε τις παρακάτω ασκήσεις χρησιμοποιώντας Python

1. Μετατρέψτε τον παρακάτω κώδικα matlab σε κώδικα python.

Σημείωση: να χρησιμοποιηθεί η βιβλιοθήκη numpy και να χρησιμοποιήσετε για το π την τιμή numpy.pi και να ορίσετε το ημίτονο sin ως numpy.sin:

Αρχείο myFunction.m:

```
function [x] = myFunction(x, y)
rad = 0;
if ( x > 180 ) & ( x <= 360 ) then
    rad = -pi * (360 - x) / 180;
elseif (x >= 0) && (x <= 180)
    rad = pi * x / 180;
else
    rad = 0;
end
x = sin(rad);
end
```

Αρχείο myScript.m:

```
question = "provide a value from 0-360";
x = input(question);
while (x < 0 | x > 360)
    x = input(question)
end
sinValue = myFunction(x);
fprintf("The result is %f", sinValue);
```

Λύση:

```
import numpy

def myFunction(x):
    rad = 0
    if ( x > 180 ) & ( x <= 360 ):
        rad = -numpy.pi * (360 - x) / 180
    elif (x >= 0) & (x <= 180):
        rad = numpy.pi * x / 180
    else:
        rad = 0;
    return numpy.sin(rad)

question = "provide a value from 0-360";
x = int(input(question))
while (x < 0 | x > 360):
    x = int(input(question))

sinValue = myFunction(x);
print(f"The result is {sinValue}");
```

2. Γράψτε ένα πρόγραμμα που να διαβάζει από τον χρήστη την τρέχουσα και την επιθυμητή θερμοκρασία (temp, desiredTemp - σε βαθμούς κελσίου) και να επιστρέφει την κατάσταση του θερμοστάτη: 'Heat' αν η θερμοκρασία είναι χαμηλότερη από την επιθυμητή μείον 3 βαθμούς, 'Cool' αν η θερμοκρασία είναι υψηλότερη από την επιθυμητή συν 3 βαθμούς, και 'off' αλλιώς.

Λύση:

```
temp = input("Provide current temperature")
desiredTemp = input("Provide desired temperature")
temp = int(temp) # Κάνουμε typecasting γιατί μπορεί ο compiler να θεωρήσει ότι είναι
μεταβλητή τύπου string
desiredTemp = int(desiredTemp) # Κάνουμε typecasting γιατί μπορεί ο compiler να
θεωρήσει ότι είναι μεταβλητή τύπου string

state = "off"
if temp < desiredTemp - 3:
    state = "Heat"
elif temp > desiredTemp + 3:
    state = "Cool"
else:
    state = "off"

print("Thermostat state is", state)
```

3. Γράψτε ένα πρόγραμμα που θα υπολογίζει τον μεγαλύτερο αριθμό στη σειρά των n φυσικών αριθμών για τους οποίους το άθροισμα των τετραγώνων τους δεν ξεπερνά το 200. Να γράψετε τρεις εκδοχές:

α) με συνδυασμένη χρήση for και if

β) με χρήση επανάληψης υπό συνθήκη (while συνθήκη).

γ) με χρήση αέναης επανάληψης (while true).

Λύση:

A)

```
sum = 0
nmax = 0
for i in range(1, 201):
    sum = sum + i**2
    if (sum > 200):
        nmax = i-1
        break

print("max n = ", nmax)
```

B)

```
i = 0
sum = 0
while sum < 200:
    i = i + 1
    sum = sum + i**2

nmax = i - 1
print("max n = ", nmax)
```

c)

```
i = 0
sum = 0
while True:
    i = i + 1
    sum = sum + i**2
    if sum > 200:
        break
nmax = i - 1
print("max n = ", nmax)
```

4. Γράψτε ένα πρόγραμμα που θα τυπώνει τους πρώτους αριθμούς που εμπεριέχονται μέχρι τον φυσικό αριθμό n. Ένας πρώτος αριθμός είναι ο αριθμός που διαιρείται μόνον με το 1 και τον εαυτό του.

Σημείωση: χρησιμοποιείστε τον τελεστή modulo % για να υπολογίσετε το υπόλοιπο (modulus) της διαίρεσης. Ο τελεστής modulo % εφαρμόζεται μόνον σε ακεραίους και επιστρέφει το υπόλοιπο μιας ακέραιης διαίρεσης, δηλαδή τον ακέραιο αριθμό r ο οποίος αν αφαιρεθεί από τον αριθμό n που θέλουμε να διαιρέσουμε με τον αριθμό m θα αποτελέσει μια τέλεια ακέραιη διαίρεση.

η σύνταξη του modulo είναι: $x \% y$. Παράδειγμα το $5 \% 2$ υπολογίζεται ως 1.

Λύση:

```
n = input("provide search range")
first_numbers = [1]
for i in range(1, int(n)):
    count = 0;
    for j in range(1, i+1):
        if i % j == 0:
            count = count + 1
    if count == 2:
        first_numbers.append(i)
print(first_numbers)
```

5. Έχουμε 4 παίκτες σε ένα τυχερό παιχνίδι το οποίο παίζεται σε κάθε κύκλο του ως εξής:

- Όλοι οι παίκτες ξεκινούν έχοντας 100 πόντους.
- Κάθε παίκτης ποντάρει έναν πόντο.
- Με τυχαιοποίηση επιλέγεται ένας αριθμός 1-4 που αντιστοιχεί στον αριθμό του παίκτη.
- Ο παίκτης που έχει τον αριθμό κερδίζει όλους τους πόντους.

A. Να φτιαχτεί κώδικας που να κάνει εξομοίωση του παιχνιδιού και να μας εκτυπώνει σε κάθε κύκλο πόσους πόντους έχουν οι παίκτες και στο τέλος να μας ενημερώνει την βαθμολογία. Θεωρείστε δεδομένο ότι το παιχνίδι σταματάει όταν ένας παίκτης χάσει όλους τους πόντους του.

B. Να λύσετε το ερώτημα A με την αλλαγή ότι το παιχνίδι σταματάει μόνον όταν όλοι οι παίκτες χάσουν τους πόντους τους (εκτός από τον νικητή).

Για να υπολογίσετε έναν τυχαίο αριθμό χρησιμοποιείτε την βιβλιοθήκη random (import random), και χρησιμοποιείτε την κλήση: random.randrange(0, 4) -> μας επιστρέφει έναν ακέραιο που είναι μεγαλύτερος από το μηδέν και μικρότερος ή ίσος με το 4.

Σχεδίαση του κώδικα:

Θα χρειαστούμε δυο μεταβλητές για να κρατάμε τα **ονόματα των παικτών** και τους **πόντους**. Αν ονόμασουμε τους παίκτες A, B, C & D, τότε η μια μεταβλητή θα είναι μια λίστα και η δεύτερη ένα dictionary. Ας τα ονομάσουμε players και game αντίστοιχα:

```
players = ["A", "B", "C", "D"]
```

```
game = {"A": 100, "B": 100, "C": 100, "D": 100}
```

Για να τερματιστεί το παιχνίδι θα χρησιμοποιήσουμε μια Boolean μεταβλητή **finish**, η οποία όσο είναι False το παιχνίδι θα προχωρά μέσα σε μια while loop.

Σε κάθε κύκλο του παιχνιδιού πρέπει να κάνουμε τις εξής ενέργειες:

- Αφαιρούμε από 1 πόντο από κάθε παίκτη.
- Ποιός είναι ο αριθμός του παίκτη που κέρδισε **winner = random.randrange(0, 4)**, άρα θα είναι ο παίκτης players[winner].
- Στη συνέχεια μεταφέρουμε τους πόντους στον κερδισμένο.
- Εξετάζουμε αν κάποιος παίκτης έχει σύνολο μηδέν, αν ναι σταματάμε το παιχνίδι.

Για το ερώτημα B θα πρέπει να κάνουμε την εξής μετατροπή, εξετάζουμε πριν ξεκινήσει ο κύκλος αν κάποιος παίκτης έχει υπόλοιπο μηδέν. Αν ναι, τότε φεύγει από τη λίστα των players, και αντίστοιχα ο νικητής θα υπολογίζεται από το **winner = random.randrange(0, len(players))**. Αν len(players) == 1 τότε το παιχνίδι τερματίζεται γιατί υπάρχει ένας μοναδικός νικητής.

Ερώτημα A:

```
import random

game = {"A": 100, "B": 100, "C": 100, "D": 100}
players = ["A", "B", "C", "D"]

i = 0
finish = False
while not finish:
    for player in players:
        game[player] = game[player] - 1
    winner = random.randrange(0, 4)
    player = players[winner]
    game[player] = game[player] + 4
    for player in players:
        if game[player] == 0:
            finish = True
    i = i + 1
    print("game ", i, " result: ", game, " finish: ", finish)

print("game result: ", game)
```

Ερώτημα Β:

```
import random

game = {"A": 100, "B": 100, "C": 100, "D": 100}
players = ["A", "B", "C", "D"]

i = 0
finish = False
while not finish:

    player_id = 0
    while player_id < len(players):
        player = players[player_id]
        if game[player] == 0:
            players.remove(player)
        else:
            player_id = player_id + 1

    for player in players:
        game[player] = game[player] - 1
    winner = random.randrange(0, len(players))
    player = players[winner]
    game[player] = game[player] + len(players)

    if len(players) == 1:
        finish = True
    i = i + 1
    # print("game ", i, " result: ", game, " finish: ", finish)

print("game terminated after ", i, "cycles, result: ", game)
```