

# Διάλεξη 3<sup>η</sup>

## ΔΟΜΕΣ ΕΛΕΓΧΟΥ

Δομές που Ελέγχουν την Ροή προγράμματος

# Διάλεξη 3<sup>η</sup>

## Δομές Ελέγχου - Ο λογικός τύπος Boolean

1 Συγκριτικοί Τελεστές

2 Λογικοί Τελεστές (and, or και not)

3 Προτεραιότητα Τελεστών

4 Δομές επιλογής (if)

- Απλή δομή επιλογής (if)
- Σύνθετη δομή επόγής (if)
- Εντολές if – else
- Πολλαπλή δομή επιλογής (if)
- Πολλαπλή δομή επιλογής if-elif-else
- Ένθετες εντολές if-else

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Εισαγωγή

Η εκτέλεση των προγραμμάτων γίνεται σειριακά, δηλαδή γραμμή-γραμμή από πάνω προς τα κάτω.

Σε οποιοδήποτε όμως σημείο **μπορούν να τεθούν περιορισμοί** για τον έλεγχο των εντολών κατά τη διάρκεια εκτέλεσής τους.

Το βασικότερο στον προγραμματισμό είναι ο έλεγχος της ροής εκτέλεσης ενός κώδικα.

Στις επόμενες διαφάνειες περιγράφονται τα εργαλεία που διαθέτει η Python για την υπό συνθήκη εκτέλεση εντολών

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Ο λογικός τύπος Boolean

Μία ειδική υποκατηγορία ακέραιων αριθμών είναι οι λογικές τιμές bool (Boolean) **True** και **False** (Αληθής-Ψευδής)

Τα True και False λέγεται ότι έχουν τύπο δεδομένων Boolean

```
.>>> a=True          #Δίνω ότι η μεταβλητή a=True
>>> type(a)         #πληκτρολογώντας type(a)
<class 'bool'>      #μου λέει ότι είναι τύπου bool
>>> b=False , >>> type(b), <class 'bool'>
```

```
c=2.3 >>> type(c) <class 'float'>
```

Η Python αντιστοιχίζει στις λογικές τιμές True και False τις τιμές 1 και 0

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Παραδείγματα True και false (στην κονσόλα)

```
>>> x=5
```

```
>>> print((3+x)<9)
```

```
True
```

```
>>> x=5
```

```
>>> print((3+x)<7)
```

```
False
```

```
>>> x=2
```

```
>>> y=3
```

```
>>> z=x<y
```

```
>>> print(z)
```

```
True
```

```
>>> a , b=5, 10      #a=5 και b=10
```

```
>>> a>b
```

```
False
```

```
>>> a!=b
```

```
#To a είναι διάφορο του b
```

```
True
```

```
>>> b==0
```

```
False
```

```
>>>
```

Όπως φαίνεται στα παραπάνω η σύγκριση μπορεί να γίνει ανάμεσα σε μεταβλητές ( $a>b$ ) και αριθμούς ( $b==0$ ), καθώς και με πολλούς άλλους τρόπους, όπως θα δούμε στη συνέχεια.

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## 1. Συγκριτικοί Τελεστές

Μία από τις βασικές λειτουργίες του υλικού ενός Η/Υ είναι η πραγματοποίηση συγκρίσεων

Για την πραγματοποίηση **κάθε είδους σύγκρισης** χρησιμοποιείται μία ειδική κατηγορία τελεστών, που ονομάζονται **συγκριτικοί ή σχεσιακοί τελεστές**

**Το αποτέλεσμα που προκύπτει είναι μια λογική τιμή True ή False.**

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Συγκριτικοί Τελεστές

ΠΙΝΑΚΑΣ 3-1 | ΣΥΓΚΡΙΤΙΚΟΙ ΤΕΛΕΣΤΕΣ

| Τελεστής           | Λειτουργία       | Παράδειγμα           | Τι σημαίνει                        | Αποτέλεσμα |
|--------------------|------------------|----------------------|------------------------------------|------------|
| <code>==</code>    | Ισότητα          | <code>5==12</code>   | Είναι το 5 ίσο με 12;              | False      |
| <code>!=</code>    | Ανισότητα        | <code>12!=3</code>   | Το 12 είναι διάφορο του 3;         | True       |
| <code>&gt;</code>  | Μεγαλύτερο       | <code>5&gt;2</code>  | Είναι το 5 μεγαλύτερο του 2;       | True       |
| <code>&lt;</code>  | Μικρότερο        | <code>-1&lt;1</code> | Είναι το -1 μικρότερο του 1;       | True       |
| <code>&gt;=</code> | Μεγαλύτερο ή ίσο | <code>3&gt;=3</code> | Είναι το 3 μεγαλύτερο ή ίσο του 3; | True       |
| <code>&lt;=</code> | Μικρότερο ή ίσο  | <code>4&lt;=2</code> | Είναι το 4 μικρότερο ή ίσο του 2;  | False      |

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Συγκριτικοί Τελεστές

Τελεστής ισότητας `==` (`a==b`, το `a` είναι ίσο με την τιμή `b`)

ενώ `a=b`, όπου `b` είναι μία παράσταση και το αποτέλεσμα της αποθηκεύεται στη μεταβλητή `a`. (`'='` τελεστής καταχώρησης)

Τελεστής διάφορο `!=`

Τελεστές `>`, `<`, `>=`, `<=` (Συγκρίνουν δύο αριθμούς ή δύο strings)

Ειδικότερα για τα strings θεωρείται ότι τα μικρά γράμματα είναι λεξικογραφικά μεγαλύτερα από τα κεφαλαία.

**Επιστρέφουν True/False**



# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Συγκριτικοί Τελεστές

Παραδείγματα:

```
>>> print(8+1==9)
```

True

```
>>> print(6*2!=7-3)
```

True

```
>>> print(4%2>=5/4)
```

False

```
>>> print(6-5<1%3)
```

False

```
>>> print('abc'<'def')
```

True

```
print('abc' < 'abcd')
```

True

```
>>> print('abc' < 'ABC')
```

False

Παράδειγμα 1. Να υπολογισθεί το επίδομα πολυετούς εργασίας υπαλλήλου, έως ποσοστό 3% του μισθού του, όταν τα έτη εργασίας είναι περισσότερα από 15.

```
misthos=float(input('Δώσε το μισθό του υπαλλήλου:'))
eti=float(input('Δώσε τα Έτη εργασίας :'))
epidoma=misthos*(eti>15)*0.03 #Γίνεται η σύγκριση και αν αληθεύει γίνεται True (=1)
print('to epidoma einai :',epidoma)
```

Δώσε το μισθό του υπαλλήλου :1245.89

Δώσε τα Έτη εργασίας :19.34

to epidoma einai : 37.3767

Παράδειγμα 2. Δίνεται το τελικό ποσό αγοράς προϊόντος. Για αγορά μεγαλύτερη από 200 ευρώ υπάρχει έκπτωση 20%. Να αναγραφεί το τελικό ποσό πληρωμής

```
poso=float(input('Δώσε το ποσό αγοράς:'))  
ekp=poso*(poso>200)*0.2  
poso=poso-ekp  
print('Το τελικό ποσό πληρωμής είναι :',poso)  
print('Η έκπτωση είναι :',ekp)
```

Δώσε το ποσό αγοράς :234.45

το τελικό ποσό πληρωμής είναι : 187.56

Η έκπτωση είναι : 46.89



# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## 2. Λογικοί Τελεστές

Αν και οι συγκριτικοί τελεστές επιτρέπουν τη σύγκριση δεδομένων σχεδόν οποιασδήποτε μορφής, πολύ συχνά οι συνθήκες που πρέπει να ελεγχθούν, είναι περισσότερο σύνθετες και απαιτείται ο συνδυασμός τους

Για να μπορέσει να γίνει ο έλεγχος πολλών συνθηκών γίνεται η χρήση των **λογικών τελεστών**.

Αυτοί είναι οι δεσμευμένες λέξεις **and**, **or** και **not**

# Ο τελεστής and

Η γενική του μορφή είναι: Συνθήκη 1 and Συνθήκη 2

Χρησιμοποιείται για τη σύνδεση δύο λογικών προτάσεων, όπου απαιτείται να ελεγχθεί αν και οι δύο προτάσεις-συνθήκες αποδίδουν τη λογική τιμή True.

Παραδείγματα:

```
>>> a,b=10,15
```

```
>>> a<10 and b<20
```

```
True
```

```
>>> a+5==b and a!=0
```

```
True
```

```
>>>
```

```
>>> a<0 and a!=b
```

```
False
```

```
>>> a*b>0 and -a>0
```

```
False
```

# Ο τελεστής or

Η γενική μορφή του τελεστή or είναι: Συνθήκη 1 or Συνθήκη 2

Αν cond1 και cond2 είναι συνθήκες, cond1 or cond2 είναι true εάν μία από τις δύο συνθήκες είναι αληθής.

```
>>> a,b=10,15
```

```
>>> a<0 or a!=b
```

```
True
```

```
>>> a<0 or a==b
```

```
False
```

Επιστρέφει True αν έστω μία ή και οι δύο συνθήκες είναι True

Επιστρέφει False αν και οι δύο συνθήκες είναι False

# Ο τελεστής not

not cond1 είναι False εάν η συνθήκη είναι True

not cond1 είναι True εάν η συνθήκη είναι False

Ο τελεστής not χρησιμοποιείται για την αντιστροφή μίας λογικής τιμής-έκφρασης.

Σε αντίθεση με τους τελεστές and και or, ο τελεστής not δέχεται έναν τελεστή στα δεξιά του και επιστρέφει πάντα μία λογική τιμή True ή False

Παραδείγματα:

```
>>> a,b=5,10
```

```
>>> a<b
```

```
True
```

```
>>> a<b and b<15
```

```
True
```

```
>>> not(a<b and b<15)
```

```
False
```

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## 3. Προτεραιότητα τελεστών

Οι συγκριτικοί τελεστές έχουν την ίδια προτεραιότητα μεταξύ τους και χαμηλότερη σε σχέση με τους αριθμητικούς τελεστές

| Προτεραιότητα (ιεραρχία) τελεστών Γλώσσας |                     |               |
|---|---------------------|---------------|
|   | Τελεστής            | Προτεραιότητα |
| Αριθμητικοί                               | ^                   | 1             |
|   | *, /, MOD, DIV      | 2             |
|   | +,-                 | 3             |
| Συγκριτικοί                               | <, >, <=, >=, <>, = | 4             |
| Λογικοί                                   | ΟΧΙ                 | 5             |
|   | ΚΑΙ                 | 6             |
|   | Η                   | 7             |



# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Προτεραιότητα τελεστών

- **Οι σχεσιακοί τελεστές έχουν μικρότερη προτεραιότητα από τους αριθμητικούς τελεστές.**
  - Για το λόγο αυτό γίνονται πρώτα οι αριθμητικές πράξεις και έπειτα η εφαρμογή των σχεσιακών τελεστών
- **Οι λογικοί τελεστές έχουν μικρότερη προτεραιότητα από τους σχεσιακούς τελεστές.**
  - Για το λόγο αυτό οι λογικοί τελεστές εφαρμόζονται τελευταίοι

Παράδειγμα : δίνεται  $n=4$   $answ='Y'$  Καθορίστε True ή False (στον Idle)

(a)  $(2 < n) \text{ and } (n < 6)$

(b)  $(2 < n) \text{ or } (n == 6)$

(c)  $\text{not } (n < 6)$

(d)  $(answ == "Y") \text{ or } (answ == "y")$

(e)  $(answ == "Y") \text{ and } (answ == "y")$

(f)  $\text{not } (answ == "y")$

(g)  $((2 < n) \text{ and } (n == 5 + 1)) \text{ or } (answ == "No")$

(h)  $((n == 2) \text{ and } (n == 7)) \text{ or } (answ == "Y")$

(i)  $(n == 2) \text{ and } ((n == 7) \text{ or } (answ == "Y"))$

ΠΑΡΑΔΕΙΓΜΑ 3a: Να γραφτεί πρόγραμμα που ελέγχει αν κάποιος περνά το εργαστήριο

Για να περάσει πρέπει να έχει βαθμό $\geq 5$  και μέχρι 3 απουσίες

οι συνθήκες ελέγχου είναι:

$b \geq 5$  and not ( $ap > 3$ )

$b \geq 5$  and  $ap \leq 3$

not( $b < 5$  or  $ap > 3$ ) Μπορούμε να χρησιμοποιήσουμε όποια θέλουμε

```
b=float(input('dose bathmo :'))
```

```
ap=int(input('dose apoysies :'))
```

```
if b $\geq$ 5 and ap $\leq$ 3:
```

```
    print('to perases')
```

```
else:
```

```
    print('kopikes')
```

## ΠΑΡΑΔΕΙΓΜΑ 3b: (προσδιορίζει ακριβώς βαθμό και τρόπο που κόπηκε)

```
b=float(input('Δώσε το βαθμό :'))
```

```
ap=int(input('Δώσε τις απουσίες :'))
```

```
if b>=5 and ap<=3: #Μπορούσαμε να χρησιμοποιήσουμε όποια συνθήκη θέλαμε
```

```
    print('Το πέρασες')
```

```
    if 5<=b<7:
```

```
        print('ΚΑΛΩΣ')
```

```
    elif 7<=b<=8.5:
```

```
        print('ΠΟΛΥ ΚΑΛΑ')
```

```
    elif 8.5<=b<=10:
```

```
        print('ΑΡΙΣΤΑ')
```

```
elif b<5 and ap>3:
```

```
    print('Κόπηκε από βαθμό και απουσίες')
```

```
elif b<5:
```

```
    print('Κόπηκε λόγω βαθμού')
```

```
elif ap>3:
```

```
    print('Κόπηκε λόγω απουσιών')
```

# Παράδειγμα 3c (πιο εμπλουτισμένο)

```
gr=eval(input('Δώσε το βαθμό του Φοιτητή: '))
ab=eval(input('\nΔώσε τον αριθμό των απουσιών του: '))
if 0<=gr<=10 and ab>=0:
    if 5<=gr<=7 and ab<=3:
        print('\nΚΑΛΩΣ!')
    elif 7<gr<=8.5 and ab<=3:
        print('\nΠΟΛΥ ΚΑΛΑ!!')
    elif 8.5<gr<=10 and ab<=3:
        print('\nΑΡΙΣΤΑ!!!')
    elif 0<=gr<5 and ab<=3:
        print("\Nη επίδοση του Φοιτητή δεν είναι ικανοποιητική. Κόπηκε λόγω βαθμού")
    else:
        print("\nΟ Φοιτητής δεν περνά το μάθημα \nκόπηκε από χαμηλό βαθμό και απουσίες.")
else:
    print('\nΤ δεδομένα που δώσατε είναι έξω από τα όρια')
```

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## 4. Δομές επιλογής if

Η εκτέλεση των εντολών μέχρι τώρα εκτελούνταν σειριακά γιατί δεν περιείχαν σύνθετες περιπτώσεις και περισσότερες από μια επιλογές.

Η αντιμετώπιση σύνθετων προβλημάτων σε κάθε γλώσσα προγραμματισμού γίνεται με τις δομές επιλογής.

Η Python διαθέτει μια μόνο δομή επιλογής, την εντολή if, αλλά διαθέτει κάποιος πιο αποδοτικούς μηχανισμούς (π.χ. λεξικά)

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Απλή δομή επιλογής if

```
if συνθήκη:  
    εντολές A  
εντολές B
```

Η λογική της συνθήκης if είναι η εξής: **Αν το αποτέλεσμα της συνθήκης είναι True, τότε θα εκτελεστούν οι εντολές που βρίσκονται στο μπλόκ εντολών A.**

Διαφορετικά οι εντολές αυτές δεν εκτελούνται.

Σε κάθε περίπτωση το πρόγραμμα θα συνεχίσει την εκτέλεσή του, με τις εντολές B μετά την if.

Δύο σημεία που πρέπει να προσέξουμε

- Το μπλόκ εντολών μέσα στην if **πρέπει να έχει εσοχή** και
- Το σύμβολο **':'** Αυτό σηματοδοτεί την έναρξη ενός μπλόκ εντολών

## Παράδειγμα 4: Να γραφεί πρόγραμμα, που θα δέχεται έναν αριθμό και θα εμφανίζει την απόλυτη τιμή του

```
x=eval(input('Δώσε έναν αριθμό :'))  
if x<0:  
    x=-x  
print('Η απόλυτη τιμή είναι :',x)
```

Παράδειγμα 4a: Στο τμήμα παραγωγής ενός εργοστασίου εγκαταστάθηκε ένα σύστημα προειδοποίησης για τη θερμοκρασία του φούρνου. Να γράψετε πρόγραμμα που θα εμφανίζει μήνυμα αν η θερμοκρασία ξεπεράσει τους 300° C

```
t=float(input('Θερμοκρασία: '))  
if t>300:  
    print('Προσοχή, μεγάλη θερμοκρασία')  
print('Η θερμοκρασία είναι ',t,'βαθμούς Κελσίου')
```



# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## 5. Σύνθετη δομή επιλογής if

Συχνά κάποια εντολή χρειάζεται να εκτελεστεί υπό συνθήκη

Αν ισχύει η συνθήκη, εκτελείται η εντολή,  
διαφορετικά όχι

If συνθήκη:

μπλόκ εντολών

else:

μπλόκ εντολών

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Σύνθετη δομή επιλογής if

```
if συνθήκη:
    _ _ _ _ εντολές } A
else:
    _ _ _ _ εντολές } B
εντολές
```

Να γράψετε πρόγραμμα που θα διαβάζει έναν αριθμό και να εμφανίζει την τιμή  $1/x$  (υπολογισμός αντιστρόφου)

```
x=float(input('Δώσε μου την τιμή του x:'))
if x!=0:
    print(1/x)
else:
    print(' Αδύνατο ')
```

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## 6. Εντολές *if-else*

*To else* σε μια εντολή *if-else* μπορεί να παραληφθεί.

- Όταν η συνθήκη είναι *false*
- η εκτέλεση συνεχίζεται με την γραμμή που ακολουθεί το μπλοκ της πρότασης *if*.

Παράδειγμα 5: Φτιάξτε ένα πρόγραμμα το οποίο θα βρίσκει το μεγαλύτερο από δύο αριθμούς που εισάγονται από το πληκτρολόγιο

```
num1=eval(input(Δώσε τον 1ο αριθμό :'))  
num2=eval(input(δοσε τον 2ο αριθμό:'))  
if num1>num2:  
    megalyteri_timi=num1  
else: #το if με το else ίδια στοίχιση  
    megalyteri_timi=num2  
print(Η μεγαλύτερη τιμή είναι:',megalyteri_timi)
```

Παράδειγμα 6: Ένας πωλητής παίρνει προμήθεια 15% αν οι πωλήσεις του ξεπεράσουν τα 5000 €. Για πωλήσεις λιγότερες από 5000 €, παίρνει μισθό 700 €. Να γραφεί πρόγραμμα που θα δέχεται τις πωλήσεις και θα εμφανίζει το ποσό που θα πληρωθεί.

```
sales=float(input(Δώσε το ποσό των πωλήσεων:'))
if sales >=5000:
    pro=sales*15/100
else:
    pro=700
print('Θα πληρωθεί :',pro,'€')
```

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## 7. Πολλαπλή δομή επιλογής if

```
if συνθήκη 1:
    _ _ _ _ εντολές } A
elif συνθήκη 2:
    _ _ _ _ εντολές } B
elif συνθήκη 3:
    _ _ _ _ εντολές } Γ
...
else:
    _ _ _ _ εντολές } ...
εντολές
```

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Πολλαπλή δομή επιλογής if

Παράδειγμα 6α: Ο χαρακτηρισμός του βαθμού γίνεται σύμφωνα με τις αντιστοιχίες του παρακάτω πίνακα. Ζητείται πρόγραμμα όπου θα εισάγεται ένας βαθμός και θα εμφανίζεται ο κατάλληλος χαρακτηρισμός

| ΒΑΘΜΟΣ   | ΧΑΡΑΚΤΗΡΙΣΜΟΣ |
|----------|---------------|
| [0-5)    | Ανεπαρκής     |
| [5-7)    | Καλός         |
| [7-8.5)  | Πολύ καλός    |
| [8.5-10] | Άριστος       |

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Πολλαπλή δομή επιλογής if-elif-else

Παράδειγμα 6a: (Για σύγκριση πάνω από δύο συνθηκών)

```
b=float(input('Δώσε το βαθμό : '))
if 0<=b<5:           #Μπορούσε και :           #b>=0 and b<5:
    print('Ανεπαρκής')
elif 5<=b<7:         #b>=5 and b<7:
    print('Καλός')
elif 7<=b<8.5:       #b>=7 and b<8.5:
    print('Πολύ καλός')
else: #elif 8.5<=b<=10: #b>=8.5 and b<=10:
    print('Άριστος')
```



# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## 8. Ένθετες εντολές *if-else*

Τα μπλοκ με εσοχή των προτάσεων *if-else* και *if* μπορούν να περιέχουν άλλες προτάσεις *if-else* και *if*.

Τότε οι προτάσεις λέγεται ότι είναι ένθετες.

## Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

### Ένθετες εντολές if-else

Παράδειγμα 7: Μία εταιρεία courier τιμολογεί τους πελάτες της σύμφωνα με τις χρεώσεις του παρακάτω πίνακα, ανάλογα με το βάρος του δέματος σε gr και τον τρόπο αποστολής του

| ΒΑΡΟΣ (gr) | ΧΡΕΩΣΗ ΑΠΟΣΤΟΛΗΣ (ΕΥΡΩ) |        |
|------------|-------------------------|--------|
|            | ΚΑΝΟΝΙΚΟ                | ΕΞΠΡΕΣ |
| 0-1000     | 3                       | 5      |
| 1001-2000  | 4                       | 7      |
| 2001-4000  | 5                       | 9      |
| 4001-....  | 8                       | 12     |

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Παράδειγμα 7 : Ένθετες εντολές if-else

```
we=int(input('Δώσε το βάρος : '))
tr=int(input('Τρόπος αποστολής (0:κανονικό, 1:εξπρές) : '))
if tr==1:
    if we <= 1000:
        x=5
    elif we <= 2000:
        x=7
    elif we <= 4000:
        x=9
    else:
        x=12
else:
    if we <= 1000:
        x=3
    elif we <= 2000:
        x=4
    elif we <= 4000:
        x=5
    else:
        x=8
print('Έξοδα αποστολής : ',x)
```

# Κεφάλαιο 3 - ΔΟΜΕΣ ΕΛΕΓΧΟΥ

## Ένθετες εντολές if-else

Παράδειγμα 8: Να γράψετε ένα πρόγραμμα που θα διαβάζει έναν αριθμό και θα ελέγχει αν είναι άρτιος ή περιττός. Στην περίπτωση που είναι άρτιος θα εμφανίζει τον αντίστροφό του, ενώ αν είναι περιττός θα εμφανίζει αν είναι θετικός ή αρνητικός

```
x=eval(input('Δώσε έναν αριθμό : '))
```

```
if x%2==0: #ελέγχει αν το υπόλοιπο είναι μηδέν, δηλαδή αν είναι άρτιος
```

```
    if x!=0:
```

```
        print(1/x)
```

```
    else:
```

```
        print('Αδύνατος υπολογισμός')
```

```
else:
```

```
    if x<0: #έλεγχος για αρνητικό περιττό
```

```
        print('Αρνητικός')
```

```
    else:
```

```
        print('Θετικός')
```

```
print('Τέλος προγράμματος')
```