Ένας αριθμός είναι, πώς κάνεις έτσι;

Ξέρουμε κάποιους/ες που τους αγχώνει η ερώτηση «Πόσο χρονών είσαι;», γι αυτό θα την αντικαταστήσουμε με την ερώτηση «Πες μας έναν αριθμό από το 1 μέχρι το 9» και μετά από μερικές πράξεις θα φτάσουμε στην επίμαχη απάντηση.

Το πρόγραμμα μας:

1. Θα ζητάει από τον χρήστη έναν αριθμό από το 1 μέχρι το 9. Αυτόν τον αριθμό, θα το διπλασιάζει, θα του προσθέτει το 5 και τέλος θα τον πολλαπλασιάζει με το 50.

2. Στη συνέχεια θα του προσθέτει τον αριθμό που θα προκύπτει από την επόμενη διαδικασία: Θα ρωτάει τον χρήστη αν έχουν περάσει τα γενέθλια του. Αν ναι, το πρόγραμμα θα αφαιρεί από το τρέχον έτος τον αριθμό 250, αν όχι θα αφαιρεί από το τρέχον έτος τον αριθμό 251.

3. Από το αριθμό που έχει προκύψει, ο χρήστης θα πρέπει να αφαιρέσει το έτος της γέννησης του και να μας δώσει το υπόλοιπο.

4. Από το υπόλοιπο αφαιρούμε τον αρχικό αριθμό επί 100 και έχουμε την ηλικία του παίκτη.

5. Το πρόγραμμα θα δίνει τη δυνατότητα στον παίκτη να παίξει όσες φορές θέλει.

Έννοιες: είσοδος, έξοδος, μεταβλητές, δομή επιλογής, δομή επανάληψης, βιβλιοθήκες.

---

1. Μηνύματα

Το πρώτο πράγμα που μαθαίνουμε σε όλες τις γλώσσες προγραμματισμού είναι να κάνουμε τον υπολογιστή να πει «Hello World!!!» Ας μην χαλάσουμε την παράδοση …

```
# Καλωσόρισμα!
print (“Hello World!”)
```

Οτιδήποτε ακολουθεί το σύμβολο # είναι σχόλιο και αγνοείται κατά την εκτέλεση του προγράμματος. Απευθύνεται σε εκείνους που διαβάζουν τον κώδικα και χρησιμεύει στην καλύτερη κατανόηση του.

Για να εμφανίσουμε ένα μήνυμα στην οθόνη χρησιμοποιούμε την print(), δίνοντας ανάμεσα στις παρενθέσεις το μήνυμα που θα εμφανιστεί. Μπορούμε να εμφανίσουμε με την print() πολλές τιμές, αρκεί να τις χωρίσουμε με κόμμα.

```
# Εμφάνιση πολλών τιμών
print ( 5, 7 , 8 )
print (“green”, “red”, “blue”)
```

Στο παράδειγμα μας θέλουμε το πρόγραμμα να προτρέπει τον χρήστη να πληκτρολογήσει έναν αριθμό από το 1 έως το 9

```
print ("Δώσε έναν αριθμό από το 1 έως το 9")
```

2. Απαντήσεις – μεταβλητές

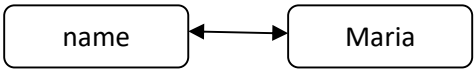
Καταφέραμε να εμφανίσουμε ένα μήνυμα που προτρέπει τον χρήστη να δώσει μια απάντηση. Όμως πως θα δοθεί αυτή;

Η input() επιστρέφει το κείμενο που πληκτρολόγησε ο χρήστης, επιστρέφει δηλαδή μια *αλφαριθμητική* τιμή.

```
print (“Πως σε λένε;”)
name = input()
print (“Γεια σου”, name)
```

Η name είναι μια μεταβλητή και έχει πλέον την τιμή που έδωσε ο χρήστης

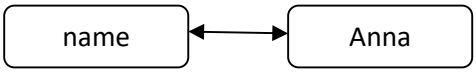
π.χ. εάν ο χρήστης πληκτρολόγησε *Maria* τότε name = Maria



Οι μεταβλητές χρησιμεύουν στην διατήρηση κάποιων τιμών που είναι χρήσιμες κατά την εκτέλεση των προγραμμάτων. Έτσι μπορούμε να αναφερθούμε στις μεταβλητές αυτές και να αντλήσουμε την τιμή που περιέχουν.

Μια μεταβλητή αντιστοιχεί σε μια τιμή (ενώ μια τιμή π.χ. 5 μπορεί να αντιστοιχεί σε πολλές μεταβλητές π.χ. x, y, z). Η αντιστοίχιση αυτή δεν ισχύει για πάντα αλλά μπορεί να αλλάξει (αντικατασταθεί) κατά την εκτέλεση του προγράμματος.

Εάν εκτελέσουμε ξανά τον προηγούμενο κώδικα και αυτή τη φορά πληκτρολογήσουμε την τιμή Anna τότε name = Anna



Εδώ χρησιμοποιούμε την input() για να διαβάσουμε την απάντηση του χρήστη, η οποία αποθηκεύεται στη μεταβλητή X. Η τιμή που επιστρέφει η input() είναι αλφαριθμητική, έχει δηλαδή την μορφή κειμένου. Αυτό δεν είναι πρόβλημα όταν ζητάμε το όνομα του χρήστη γιατί κι αυτό έχει την μορφή κειμένου. Όταν όμως θέλουμε να ζητήσουμε από το χρήστη έναν ακέραιο ή έναν πραγματικό αριθμό τότε θα πρέπει να μετατρέψουμε την αλφαριθμητική τιμή που επιστρέφει η input() σε αριθμητική, με τον τρόπο που φαίνεται παρακάτω:

```
# το κείμενο που πληκτρολογεί ο χρήστης μετατρέπεται σε ακέραιο αριθμό και
# αποθηκεύεται στην μεταβλητή x
Print ("δώσε έναν αριθμό από το 1 έως το 9")
x = int(input())
```

3. Πράξεις

Το σύμβολο * αντιστοιχεί στην πράξη του πολλαπλασιασμού. Μπορείτε επίσης να χρησιμοποιήσετε τα κλασικά + και -, καθώς επίσης και το / για τη διαίρεση, το // για το πηλίκο της ακέραιας διαίρεσης και το % για το υπόλοιπο της ακέραιας διαίρεσης.

```
y = x * 2
y = y + 5
y = y * 50
```

4. Βιβλιοθήκες

Οι βιβλιοθήκες περιέχουν έτοιμο κώδικα και τις συναντάμε στις περισσότερες γλώσσες προγραμματισμού: είναι συλλογές από μικρά προγράμματα που μπορούμε να χρησιμοποιήσουμε στα προγράμματά μας. Για να χρησιμοποιήσουμε μια βιβλιοθήκη θα πρέπει πρώτα να την εισάγουμε (**import**). Εδώ θα εισάγουμε τη βιβλιοθήκη **time** (που περιέχει έτοιμα υποπρογράμματα για τη διαχείριση του χρόνου) και θα χρησιμοποιήσουμε την localtime() για να πάρουμε το τρέχον έτος. Η localtime() της βιβλιοθήκης time επιστρέφει πληροφορίες για την ημερομηνία και την ώρα του συστήματος.

```
import time
year = time.localtime().tm_year
```

Επίσης, η βιβλιοθήκη time περιέχει την sleep η οποία προσθέτει μια καθυστέρηση (σε δευτερόλεπτα) στην ροή εκτέλεσης του προγράμματος.

```
import time
year = time.localtime().tm_year
print (“calculating…”)
time.sleep(5)
print (year)
```

Το μήνυμα calculating… παραμένει στην οθόνη του υπολογιστή δίνοντας την αίσθηση πως διενεργείται κάποιος υπολογισμός. Στην πραγματικότητα ο υπολογισμός του έτους έχει ήδη πραγματοποιηθεί και το πρόγραμμα βρίσκεται σε παύση.

5. Έλεγχος και Επιλογές

```
# Το λάθος παράδειγμα!
print (“Δώσε το μέσο όρο του μαθητή”)
mo = int(input())
if mo > 10:
        print (“Ο μαθητής πέρασε”)
else:
        print(“Ο μαθητής θα επανεξεταστεί”)
```

Η if συνοδεύεται από μια συνθήκη, η οποία ελέγχεται κατά την εκτέλεση του προγράμματος και μπορεί να είναι αληθής (True) ή ψευδής (False). Οι εντολές που ακολουθούν την if είναι στοιχισμένες δεξιότερα από αυτή. Η στοίχιση επιτυγχάνεται εισάγοντας κενά πριν από τις εντολές, τα οποία υποδηλώνουν ότι αυτές οι εντολές θα εκτελεστούν μόνο αν η συνθήκη είναι αληθής. Παρομοίως, οι εντολές που ακολουθούν την else είναι στοιχισμένες δεξιότερα και θα εκτελεστούν μόνο εφόσον η συνθήκη είναι ψευδής.

Θα πρέπει να εκτελούμε τις δομές επιλογής για όλες τις δυνατές περιπτώσεις ώστε να επιβεβαιώσουμε πως λειτουργούν ορθά. Στο παράδειγμα μας έχουμε μια δομή επιλογής με δύο πιθανές περιπτώσεις. Εισάγοντας τιμές όπως 15, 18, 9 βλέπουμε πως το πρόγραμμα επιστρέφει το επιθυμητό αποτέλεσμα. Τι συμβαίνει όταν δώσουμε ως μέσο όρο την τιμή 10; Πρέπει να διορθώσουμε την συνθήκη της δομής επιλογής.

```
# η σωστή συνθήκη για το προηγούμενο παράδειγμα
if mo > = 10
# ή εναλλακτικά
if mo < 10
```

Το σύμβολο < χρησιμοποιείται για την σύγκριση τιμών, όπως επίσης και τα < =, > και > =. Για να ελέγξουμε αν δύο τιμές είναι ίσες χρησιμοποιούμε το == και για να ελέγξουμε αν είναι διαφορετικές το != .

```
# έλεγχος για αν έχουν περάσει τα γενέθλια του παίχτη
print ('Έχουν περάσει τα γενέθλια σου; (N/O)')
        ans1= input()
        if ans1=="N":
                z=year-250
        else:
                z=year-251
```

Μην παραλείπετε το σύμβολο **:** μετά την συνθήκη της if και την else.

6. Βρόχος και όχι Βρόγχος

Έχετε όλα τα απαραίτητα για να γράψετε ένα πρόγραμμα που θα λύνει το παιχνίδι της αποκάλυψης της ηλικίας. Ένας χρήστης θα παίξει μια φορά και το πρόγραμμα θα τερματίσει. Κι αν θέλουμε να παίξουμε περισσότερες φορές ;

Στην Python, η βασική επαναληπτική δομή (βρόχος) είναι η while, η οποία μπορεί να χρησιμοποιηθεί σε κάθε περίπτωση, ενώ για συγκεκριμένου είδους επαναλήψεις υπάρχει και η δομή for. Η επαναληπτική δομή while συνοδεύεται από μια συνθήκη συνέχειας. Η συνθήκη ελέγχεται στην αρχή κάθε νέου κύκλου της επανάληψης και μπορεί να είναι αληθής (True) ή ψευδής (False). Όσο η συνθήκη είναι αληθής, η επανάληψη συνεχίζεται για άλλον έναν κύκλο. Οι εντολές που ακολουθούν τη while είναι στοιχισμένες δεξιότερα. Η στοίχιση αυτή επιτυγχάνεται εισάγοντας κενά πριν από τις εντολές, τα οποία υποδηλώνουν ότι αυτές οι εντολές “ανήκουν” στη while και θα επαναλαμβάνονται. Η πρώτη εντολή μετά τη while που δεν θα είναι στοιχισμένη δεξιότερα δεν θα επαναλαμβάνεται, αλλά θα εκτελεστεί μόνο μια φορά, όταν η επανάληψη τερματιστεί.

```
again=True
while again == True:
        ……..
        ……
        ……
       print ("Θέλεις να ξαναπαίξεις με άλλον αριθμό; (N/O)")
        ans2= input()
        if ans2=="N":
                again = True
        else:
                again = False
```

Τελικό Πρόγραμμα

```
# ένας αλγόριθμος που βρίσκει την ηλικία σου από έναν τυχαίο αριθμό από το 1 μέχρι το 9
# μπορούμε να διαβάσουμε τον αριθμό από τον χρήστη ή να το παράγουμε με την random.
# εισάγω τη βιβλιοθήκη time για να πάρω το τρέχον έτος
import time
again=True
while again == True:
        print ("δώσε έναν αριθμό από το 1 έως το 9")
        x=int(input())
        y = x * 2
        y = y + 5
        y = y * 50
# καταχωρώ στη μεταβλητή year το τρέχον έτος
        year = time.localtime().tm_year
# αν έχουν περάσει τα γενέθλια του παίκτη αφαιρώ από το τρέχον έτος το 250 αλλιώς
#αφαιρώ το 251.
        print ("έχουν περάσει τα γενέθλια σου; (N/O)")
        ans1=input()
        if ans1=="N":
                z=year-250
        else:
                z=year-251
        y=y+z
        print ("από τον" ,y," αφαίρεσε το έτος της γέννησης σου")
        print ("δώσε τον αριθμό που βρήκες")
        k=int(input())
        age = k-x*100
        print ("είσαι", age," χρονών")
        print ("Θέλεις να ξαναπαίξεις με άλλον αριθμό; (N/O)")
        ans2=input()
        if ans2=="N":
                again = True
        else:
              again = False
```

7. Βελτιστοποίηση –Τροποποίηση προγράμματος

Τα προγράμματα μετά την αρχική τους υλοποίηση και δοκιμή μπορούν να τροποποιηθούν ώστε να βελτιστοποιηθεί η λειτουργία τους ή/και να προστεθούν νέοι παράμετροι – λειτουργίες. Στο προηγούμενο πρόγραμμα μετά από κάποιες δοκιμές θα παρατηρήσουμε πως το αποτέλεσμα δεν είναι αυτό το οποίο περιμέναμε. Συγκεκριμένα:

Τι θα συμβεί εάν ο χρήστης πληκτρολογήσει

α)έναν αριθμό εκτός του εύρους 1 – 9;

……

……

……

β) κάποιον αλφαριθμητικό χαρακτήρα;

……

……

……

Προσθέστε κατάλληλες εντολές στο πρόγραμμα ώστε να λυθούν τυχόν προβλήματα από τις παραπάνω περιπτώσεις.

Ένα ακόμη σημείο του προγράμματος όπου μπορεί να αντιμετωπίσουμε αντίστοιχα προβλήματα με την προηγούμενη περίπτωση είναι όταν ο χρήστης πρέπει να απαντήσει ΝΑΙ ή ΟΧΙ ώστε να συνεχιστεί κατάλληλα το πρόγραμμα. Πως θα το πει το ΝΑΙ ο χρήστης; Με “Ν”, “n”, “Ν”(ελληνικό) ή με κάτι άλλο;

Στην 1η περίπτωση όπου δηλώνουμε εάν έχουν περάσει τα γενέθλια ή όχι,

```
print ("έχουν περάσει τα γενέθλια σου Ν/Ο;")
        ans1=input()
        if ans1=="N":
                z=year-250
        else:
                z=year-251
```

η λανθασμένη πληκτρολόγηση χαρακτήρα θα μας οδηγήσει σε λάθος αποτέλεσμα (π.χ. ενώ έχουν περάσει τα γενέθλια να εκτελέσει την πράξη για όταν δεν έχουν περάσει)

Μπορείτε να προτείνετε και να υλοποιήσετε μια λύση που θα αντιμετωπίζει το πρόβλημα που περιγράψαμε;

……

……

……

……

Στο τέλος του προγράμματος , εάν ο χρήστης δεν μείνει ικανοποιημένος από την ηλικία που έδωσε ο υπολογιστής ή πιστεύει πως κάνει λάθος μπορεί να δοκιμάσει ξανά αποκρινόμενος με N στην τελευταία ερώτηση του προγράμματος

```
print ("θέλεις να ξαναπαίξεις με άλλον αριθμό Ν/Ο")
        ans2=input()
        if ans2=="N":
                again = True
        else:
                again = False
```

Μπορείτε και εδώ να προτείνετε μια λύση που να περιλαμβάνει όλες τις θετικές απαντήσεις του χρήστη; (π.χ. N, ν, N, n …)

……

……

……

……