

Ακολουθία

- Πρόγραμμα εισαγωγής πλευρών α, β ορθογώνιου παραλληλογράμμου και υπολογισμού: περιμέτρου, εμβαδού και διαγωνίου (orthogwnio.psc)
- Πρόγραμμα εισαγωγής ακτίνας ρ και υπολογισμού: διαμέτρου, περιφέρειας, εμβαδού και όγκου σφαίρας (πr^3) (aktina.psc)
- Πρόγραμμα μετατροπής μιας θερμοκρασίας από C σε F: $F - 32/9 = C/5$ (thermokrasia.psc)
- Πρόγραμμα εισαγωγής 2 σημείων $(x_1, y_1), (x_2, y_2)$ και υπολογισμού της απόστασής τους (apostash.psc)
- Πρόγραμμα εισαγωγής ενός 3ψήφιου ακέραιου και υπολογισμού του αθροίσματος των ψηφίων του (athroisma_pshfiwn_trhpsifioy.psc)
- Πρόγραμμα εισαγωγής ενός 3ψήφιου ακέραιου και υπολογισμού του συμμετρικού του (symmetrikos_tripshfioy.psc)
- Πρόγραμμα εισαγωγής ενός χρονικού διαστήματος σε sec και αναγωγής του στις περισσότερες δυνατές ημέρες, ώρες, λεπτά και sec (xronos_anagwgh.psc)
- "Επιμερισμός ποσότητας" π.χ. να γραφεί πρόγραμμα που διαβάζει το συνολικό ποσό επιχορήγησης τριών δήμων και τους πληθυσμούς τους. Εάν το ποσό διανέμεται αναλογικά με βάση τον πληθυσμό του κάθε δήμου, να εμφανίζεται το ποσό που θα λάβει ο κάθε δήμος (epimerismos.psc)
- Εισαγωγή ενός 4ψήφιου και ενός 1ψήφιου ακέραιου. εμφάνιση του 5ψήφιου που προκύπτει εάν ο μονοψήφιος γίνει μεσαίο ψηφίο του τετραψήφιου π.χ. 1234, 5 \rightarrow 12534 (tetrapshfios_pentapshfios.psc)

113. Να γραφεί πρόγραμμα που διαβάζει δύο ώρες της ίδιας ημέρας (ώρα:λεπτά:δευτερόλεπτα) και εμφανίζει τη χρονική τους απόσταση σε ώρες, λεπτά και δευτερόλεπτα (chronikh_apostash.psc). π.χ.

```
Dwse 1h wra
8
15
5
Dwse 2h wra
10
0
20
Apechoyn kata 1 wres 45 lepta kai 15 sec
```

Επιλογή

- Πρόγραμμα εισαγωγής ενός ακέραιου και χαρακτηρισμού του ως άρτιος / περιττός (artios-perittos.psc)
- Πρόγραμμα εισαγωγής ενός βαθμού (0-20) και χαρακτηρισμού του ως κακός (≤ 9), μέτριος (≤ 14), καλός (≤ 18), άριστος (charakthrismos_bathmoy.psc)
- Πρόγραμμα που διαβάζει το βάρος και το ύψος(m) ενός ανθρώπου και υπολογίζει τον δείκτη μάζας σώματος $\Delta M\Sigma = B/(Y*Y)$. Βάση αυτού τον χαρακτηρίζει ως: Λιποβαρής εάν $\Delta M\Sigma < 18.5$ Φυσιολογικός εάν $18.5 \leq \Delta M\Sigma < 24.9$ Υπέρβαρος εάν $24.9 \leq \Delta M\Sigma < 29.9$ Παχύσαρκος εάν $\Delta M\Sigma \geq 29.9$ (DeikthsMazasSwmatos.psc)
- Πρόγραμμα εισαγωγής 3 αριθμών και υπολογισμός του μέγιστου και του ελάχιστου (min-max3.psc)
- Πρόγραμμα εισαγωγής των μοντέλων και των τιμών 3 αυτοκινήτων και υπολογισμός του φθηνότερου μοντέλου (μόνο ένα) και της τιμής του (MIN_MAX3_CARS.psc)
- Πρόγραμμα που διαβάδει έναν θετικό ακέραιο x και ελέγχει εάν αποτελεί το τετράγωνο θετικού ακέραιου y ($x = y^2$) (tetragwnoAkeraioy.psc)
- Πρόγραμμα επίλυσης πρωτοβάθμιας εξίσωσης $ax+\beta=0$ (prwtobathmia.psc)
- Πρόγραμμα επίλυσης δευτεροβάθμιας εξίσωσης $ax^2+\beta x+\gamma=0$ (deyterobathmia.psc)
- Πρόγραμμα που διαβάζει το ετήσιο εισόδημα ενός φορολογούμενου και υπολογίζει ΚΛΙΜΑΚΩΤΑ τον φόρο του: Από 0-3000 φόρος: 0% Από 3001-5000 φόρος: 10% Από 5001-10000 φόρος: 15% Από 10001 και άνω φόρος: 20% Στη συνέχεια να διαβάζει τον αριθμό των παιδιών του φορολογούμενου να του κάνει πιθανή έκπτωση φόρου ως εξής: 1 παιδί: -100 ευρώ 2 παιδιά: -200 ευρώ 3 παιδιά και άνω: -300 ευρώ Να εμφανίζει τον τελικό πληρωτέο φόρο (klimakwth.psc)

- Πρόγραμμα το οποίο διαβάζει έναν αριθμό ατόμων και έναν αριθμό ομάδων (\leq άτομα) και χωρίζει τα άτομα σε "ισοδύναμες" ομάδες π.χ. 27 άτομα, 4 ομάδες \Rightarrow 1 ομάδα των 6 και 3 ομάδες των 7 π.χ. 20 άτομα, 5 ομάδες \Rightarrow 5 ομάδες των 4 π.χ. 21 άτομα, 6 ομάδες \Rightarrow 3 ομάδες των 3 και 3 ομάδες των 4 (XwrismosSeOmades.psc)

36. Να γραφεί πρόγραμμα που διαβάζει την τρέχουσα ώρα σε μορφή: ώρες(0-23)-λεπτά(0-59) και εμφάνιση της λεκτικής περιγραφής της ως εξής (time.psc)

00:10 \rightarrow 12 και 10,	01:30 \rightarrow 1 και μισή
02:40 \rightarrow 3 παρά 20,	03:00 \rightarrow 3 ακριβώς
13:05 \rightarrow 1 και 5,	04:15 \rightarrow 4 και τέταρτο
20:45 \rightarrow 9 παρά τέταρτο	00:45 \rightarrow 1 παρά τέταρτο

117. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό 3ψήφιο ακέραιο και εμφανίζει την **κρυπτογραφημένη τιμή του αυξάνοντας το κάθε ψηφίο του κατά 1 κυκλικά** ($0 \Rightarrow 1, 1 \Rightarrow 2, \dots 8 \Rightarrow 9, 9 \Rightarrow 0$) (kryptografhsh_tripshfiou.psc) π.χ. 327 \Rightarrow 438, 106 \Rightarrow 217, 394 \Rightarrow 405

129. **Αριθμοί Cunningham**: ορίζονται οι θετικοί ακέραιοι αριθμοί C οι οποίοι μπορούν να γραφτούν στη μορφή $C=b^2-1$, με b θετικό ακέραιο και $b>1$. Ονομάστηκαν έτσι προς τιμή του Allan Joseph Champneys Cunningham (1842–1928) Βρετανο-Ινδού μαθηματικού. π.χ. $15=4^2-1$, $24=5^2-1$, ενώ δεν υπάρχει θετικός ακέραιος b για να γραφτεί το 20 σ' αυτή τη μορφή. Να γράψετε πρόγραμμα το οποίο: να διαβάζει έναν θετικό ακέραιο αριθμό a . Να ελέγχει αν ο a αποτελεί αριθμό Cunningham και αν αυτό συμβαίνει να τον εμφανίζει στη μορφή $a=b^2-1$ (π.χ. για $a=8$ να εμφανίζει $8=3^2-1$), διαφορετικά να εμφανίζει σχετικό μήνυμα. (cunningham_numbers.psc)

Επανάληψη

- Πρόγραμμα εμφάνισης της προπαίδειας (propaideia.psc)
- Πρόγραμμα εμφάνισης όλων των συνδυασμών ρίψης 3 ζαριών (zaria.psc)
- Εμφάνιση όλων των ενδείξεων ενός ψηφιακού ρολογιού από την ώρα 0:0:0 έως και 23:59:59 (pshfiako_roloi.psc)
- Πρόγραμμα που συνεχώς εμφανίζει το μήνυμα 'ΚΑΛΑ ΧΡΙΣΤΟΥΓΕΝΝΑ' γράμμα - γράμμα (christmas.psc)
- Πρόγραμμα που εμφανίζει τους 3 θετικούς ακέραιους στο διάστημα 1-1000 που είναι πολλαπλάσια του 7, οι προηγούμενοί τους πολλαπλάσια του 5 και οι επόμενοί τους πολλαπλάσια του 9 (Pollaplasia_5_7_9.psc)
- Πρόγραμμα που εμφανίζει όλους τους θετικούς 3ψήφιους ακέραιους με όλα τα ψηφία τους να είναι διαφορετικά. Να εμφανίζει και το συνολικό πλήθος τους (TripshfioiMeDiaforetikaPshfia.psc)
- Εισαγωγή 10 βαθμών στην 20θμια. Εμφάνιση: μέσος όρος, μέγιστος, %ποσοστό άριστων (>18) (statistika1.psc)
- Πρόγραμμα εισαγωγής κεφαλαίου και επιτοκίου και υπολογισμού του κεφαλαίου μετά από δεδομένο αριθμό χρόνων (trapeza.psc)
- Πρόγραμμα που διαβάζει 2 θετικές ακέραιες τιμές και υπολογίζει το γινόμενο τους με τον αλγόριθμο του Πολλαπλασιασμού αλά ρωσικά (pollaplasiasmos_ala_rwsika.psc)

```

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
2 x 1 = 2

```

...

```

9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81

```

```

---
984
985
986
987
Πλήθος αριθμών: 648

```

```

Κ
ΚΑ
ΚΑΑ
ΚΑΑΑ
ΚΑΑΑ Χ
ΚΑΑΑ ΧΡ
ΚΑΑΑ ΧΡΙ
ΚΑΑΑ ΧΡΙΣ
ΚΑΑΑ ΧΡΙΣΤ
ΚΑΑΑ ΧΡΙΣΤΟ
ΚΑΑΑ ΧΡΙΣΤΟΥ
ΚΑΑΑ ΧΡΙΣΤΟΥΓ
ΚΑΑΑ ΧΡΙΣΤΟΥΓΕ
ΚΑΑΑ ΧΡΙΣΤΟΥΓΕΝ
ΚΑΑΑ ΧΡΙΣΤΟΥΓΕΝΝ
ΚΑΑΑ ΧΡΙΣΤΟΥΓΕΝΝΑ

```


- Πρόγραμμα εισαγωγής των ψηφίων ενός αριθμού ένα προς ένα. Τερματίζει με την εισαγωγή της τιμής -1. Εμφανίζει τον αριθμό ολοκληρωμένο π.χ. για είσοδο: 2, 5, 8, 0, 4, -1 έξοδος: 25804 (sxhmatismosArithmoy.psc)
- Πρόγραμμα που διαβάζει 2 θετικούς ακέραιους και τους συνενώνει σε έναν. π.χ. $x=12$, $y=765$, αποτέλεσμα: 12765 (SynenwshArithmwon.psc)
- Πρόγραμμα το οποίο διαβάζει έναν θετικό ακέραιο και υπολογίζει και εμφανίζει τον ακέραιο $[1,24]$ που θα προκύψει μετά από συνεχόμενη πρόσθεση των ψηφίων του. π.χ. με είσοδο τον 6789 $\rightarrow 6+7+8+9=30 \rightarrow 3+0=3$ (fternisma.psc)
- Πρόγραμμα που διαβάζει έναν 2δικό αριθμό και εμφανίζει τον ισοδύναμο 10δικό (π.χ. 110101 \rightarrow 53) (BIN2DEC.pdf)
- Πρόγραμμα που διαβάζει έναν 10δικό αριθμό και εμφανίζει τον ισοδύναμο 2δικό (π.χ. 53 \rightarrow 110101) (DEC2BIN.pdf)

1. Εμφάνιση των 4ψήφιων ακέραιων με τα ψηφία τους ανά δύο ίσα. π.χ. 5757 (τροποποίηση για 6ψήφιους) (ana2isa.psc)

```
82 . 9191
83 . 9292
84 . 9393
85 . 9494
86 . 9595
87 . 9696
88 . 9797
89 . 9898
90 . 9999
```

4. Εμφάνιση των 3ψήφιων ακέραιων xyz: $x < y < z$ και πόσοι είναι συνολικά (tripshfioiMeSxeshMikroteroy.psc)

```
76 . 568
77 . 569
78 . 578
79 . 579
80 . 589
81 . 678
82 . 679
83 . 689
84 . 789
```

5. Εμφάνιση των 3ψήφιων ακέραιων xyz που είναι «ισορροπημένοι»: $\max\{x,y,z\} = (x+y+z)/2$ π.χ. 123: $3 = (1+2+3)/2$ και πόσοι είναι συνολικά (126) (isorrophenomenoi.psc)

```
118 . 918
119 . 927
120 . 936
121 . 945
122 . 954
123 . 963
124 . 972
125 . 981
126 . 990
```

7. Εμφάνιση των 100 πρώτων τιμών της ακολουθίας:
1,2,4,7,11,16,22,...4951 (akolythia100oroi.psc)

```
4187
4279
4372
4466
4561
4657
4754
4852
4951
```

9. Εισαγωγή ακέραιου και εμφάνιση του ακέραιου με τα ψηφία του αρχικού αυξημένα κατά 1 (το 9 να γίνεται 0). Π.χ. 12389 → 23490 (akeraios_pshfia+1.psc)

```
Enter number:
12389
Result: 23490
```

10. Εισαγωγή 10 βαθμών στην 20θμια (0-20) με έλεγχο εγκυρότητας και εύρεση: του μέσου όρου τους και του πλήθους των άκυρων βαθμών που δόθηκαν (plithos_akyrown_bathmwn.psc)

```
12
22
14
16
33
12
20
44
16
19
18
11
66
15
μέσος όρος: 15.30 πλήθος άκυρων: 4
```

11. Εμφάνιση όλων των θετικών 5ψήφιων ακέραιων που είναι συμμετρικοί (π.χ. 12321) (symmetrikoι5pshfioi.psc)

```
892 . 99199
893 . 99299
894 . 99399
895 . 99499
896 . 99599
897 . 99699
898 . 99799
899 . 99899
900 . 99999
```

16. Πάρτε οποιοδήποτε θετικό ακέραιο n . Αν ο n είναι άρτιος, διαιρέστε τον δια το 2, για να πάρετε το $n/2$. Εάν ο n είναι περιττός, πολλαπλασιάστε τον επί 3 και προσθέστε 1 για να πάρετε το $3n+1$. Κατά την εικασία του Κόλατζ, από όποιο αριθμό κι αν ξεκινήσετε, θα καταλήξετε πάντα στο ένα. Μετρήστε τον αριθμό των βημάτων που απαιτήθηκαν. (colatz.psc)

```
Enter number
15
1 . 46
2 . 23
3 . 70
4 . 35
5 . 106
6 . 53
7 . 160
8 . 80
9 . 40
10 . 20
11 . 10
12 . 5
13 . 16
14 . 8
15 . 4
16 . 2
17 . 1
steps = 17
```

23. Πρόγραμμα που εμφανίζει το παρακάτω:
(pollaplasiasmos_111.psc)

```
1 x 1 = 1
11 x 11 = 121
111 x 111 = 12321
1111 x 1111 = 1234321
11111 x 11111 = 123454321
111111 x 111111 = 12345654321
1111111 x 1111111 = 1234567654321
11111111 x 11111111 = 123456787654321
111111111 x 111111111 = 12345678987654321
```

26. Εισαγωγή χαρακτήρα προς χαρακτήρα των συμβόλων μιας αριθμητικής έκφρασης μέχρι να δοθεί ο χαρακτήρας '#' ή όταν δοθεί δεξιά παρένθεση ')' που δεν έχει αντίστοιχη αριστερή. Να ελεγχθεί εάν η έκφραση έχει ισορροπημένες παρενθέσεις καθώς και το πλήθος των αριστερών και δεξιών παρενθέσεων που περιέχει (parentheseis.psc).

```
(
1
+
2
)
*
3
#
Ισορροπημένες
αριστερές παρενθέσεις: 1 δεξιές παρενθέσεις: 1
```

```
(
1
+
2
)
)
Μη ισορροπημένες
αριστερές παρενθέσεις: 1 δεξιές παρενθέσεις: 2
```

```
(
1
+
2
#
Μη ισορροπημένες
αριστερές παρενθέσεις: 1 δεξιές παρενθέσεις: 0
```


35. Να γραφεί πρόγραμμα που διαβάζει μία ακολουθία χαρακτήρων (έναν προς έναν) και εμφανίζει το είδος του χαρακτήρα:

- πεζός Ελληνικός (α-ω),
- κεφαλαίος Ελληνικός (Α-Ω),
- πεζός Αγγλικός (a-z),
- κεφαλαίος Αγγλικός (A-Z),
- αριθμός (0-9),
- άλλος.

Να σταματάει όταν δοθεί ο χαρακτήρας: '#' (eidh_xarakthrwon.psc). π.χ.

```
Φ
Ελληνικός πεζός
Ε
Ελληνικός κεφαλαίος
Γ
English capital
ω
English lowercase
8
Αριθμός
&
Ήθος
#
```

38. Να γραφεί πρόγραμμα που εμφανίζει όλες (13.983.816) τις πιθανές τυχερές εξάδες κλήρωσης του ΛΟΤΤΟ. Το ΛΟΤΤΟ κληρώνει 6 αριθμούς στο διάστημα 1-49. (lotto_syndyasmoi.psc)

```
1 2 3 4 5 6
1 2 3 4 5 7
1 2 3 4 5 8
1 2 3 4 5 9
1 2 3 4 5 10
1 2 3 4 5 11
1 2 3 4 5 12
1 2 3 4 5 13
1 2 3 4 5 14
1 2 3 4 5 15
1 2 3 4 5 16
1 2 3 4 5 17
1 2 3 4 5 18
1 2 3 4 5 19
1 2 3 4 5 20
1 2 3 4 5 21
```

37. Να γραφεί πρόγραμμα που διαβάζει για 2 χώρες: τα ονόματα, τους σημερινούς πληθυσμούς και τους % ετήσιους ρυθμούς αύξησής τους. Να εμφανίζει σε πόσα χρόνια η μικρότερη σε πληθυσμό χώρα θα ξεπεράσει τη μεγαλύτερη ή το μήνυμα "δεν θα συμβεί ποτέ" (xwres_plithismoι2.psc).

```
Όσσε όνομα, πληθυσμό και % ρυθμό της 1ης χώρας
x1
1000
2
Όσσε όνομα, πληθυσμό και % ρυθμό της 2ης χώρας
x2
2000
1
Σε 71 χρόνια πληθυσμός χώρας x1 : 4079 και πληθυσμός χώρας x2 : 4053
```

```
Όσσε όνομα, πληθυσμό και % ρυθμό της 1ης χώρας
x1
1000
1
Όσσε όνομα, πληθυσμό και % ρυθμό της 2ης χώρας
x2
2000
-1
Σε 35 χρόνια πληθυσμός χώρας x1 : 1416 και πληθυσμός χώρας x2 : 1406
```

```
Όσσε όνομα, πληθυσμό και % ρυθμό της 1ης χώρας
x1
1000
-1
Όσσε όνομα, πληθυσμό και % ρυθμό της 2ης χώρας
x2
2000
-2
Σε 69 χρόνια πληθυσμός χώρας x1 : 499 και πληθυσμός χώρας x2 : 496
```

```
Όσσε όνομα, πληθυσμό και % ρυθμό της 1ης χώρας
x1
1000
0
Όσσε όνομα, πληθυσμό και % ρυθμό της 2ης χώρας
x2
2000
-1
Σε 69 χρόνια πληθυσμός χώρας x1 : 1000 και πληθυσμός χώρας x2 : 999
```

```
Όσσε όνομα, πληθυσμό και % ρυθμό της 1ης χώρας
x1
1000
1
Όσσε όνομα, πληθυσμό και % ρυθμό της 2ης χώρας
x2
2000
2
Δεν θα συμβεί ποτέ
```

```
Όσσε όνομα, πληθυσμό και % ρυθμό της 1ης χώρας
x1
1000
-2
Όσσε όνομα, πληθυσμό και % ρυθμό της 2ης χώρας
x2
2000
-1
Δεν θα συμβεί ποτέ
```


39. Να γραφεί πρόγραμμα που διαβάζει έναν πραγματικό αριθμό και υπολογίζει και εμφανίζει το ακέραιο μέρος του χωρίς την χρήση της εντολής A_M(). (akeraio_meros.psc)

```
Enter number  
234.33465  
Result 234
```

40. Ένας Ινδός μαθηματικός υπολόγισε την τιμή του π σύμφωνα με τη σχέση:

$$\pi = \sqrt{12} \left(1 - \frac{1}{3 \cdot 3} + \frac{1}{5 \cdot 3^2} - \frac{1}{7 \cdot 3^3} + \frac{1}{9 \cdot 3^4} - \frac{1}{11 \cdot 3^5} + \dots \right)$$

Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο n και υπολογίζει την τιμή του π έως και τον όρο

$$\frac{1}{(2 \cdot n + 1) \cdot 3^n}$$

Η τιμή του π με 20 δεκαδικά ψηφία είναι: 3.14159265358979323846. (ypologismos_pi.psc) Παραδείγματα:

$n = 10 \rightarrow \pi = 3.14159330450308, n = 20 \rightarrow \pi = 3.14159265359563, n = 30 \rightarrow \pi = 3.14159265358979$

41. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο αριθμό και εμφανίζει αν είναι άρτιος ή περιττός χωρίς να κάνει χρήση των αριθμητικών τελεστών της διαίρεσης (div, mod, * και /). (artios_perittos.psc, artios_perittos2.psc)

44. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο αριθμό (διασφάλιση εισόδου) και ελέγχει αν είναι καρκινικός αριθμός ή όχι εμφανίζοντας σχετικό μήνυμα. Καρκινικοί ονομάζονται οι αριθμοί που μπορούν να διαβαστούν και από τα δεξιά και από τα αριστερά π.χ. 232, 4554, 12321 κλπ. (karkinikos_arithmos.psc)

49. Να γραφεί πρόγραμμα που παρακολουθεί τα στατιστικά ενός παίκτη του μπάσκετ κατά τη διάρκεια ενός παιχνιδιού. Το πρόγραμμα θα παίρνει τα εξής στοιχεία:

- '1' κάθε φορά που ο παίκτης επιτυγχάνει καλάθι με ελεύθερη βολή,
- '2' κάθε φορά που ο παίκτης επιτυγχάνει δίποντο,
- '3' κάθε φορά που ο παίκτης επιτυγχάνει τρίποντο και
- '4' κάθε φορά που ο παίκτης κάνει φάουλ.

Πληκτρολογώντας '0' δείχνουμε στο πρόγραμμα ότι τελείωσε το παιχνίδι. Όταν ο παίκτης συμπληρώσει πέντε φάουλ, θα πρέπει να βγαίνει το μήνυμα: "ΒΓΑΙΝΕΙ ΕΞΩ ΜΕ 5 ΦΑΟΥΛ". Όταν τελειώσει η συμμετοχή του παίκτη –είτε λόγω αποβολής είτε λόγω τέλους του παιχνιδιού– θα γράφεται στην οθόνη η στατιστική του. (basket_statistika.psc).

Προέκταση: οι επιλογές 1-3 να δίνονται για κάθε βολή/δίποντο/τρίποντο και να εισάγεται επιπλέον (1/0) κατά πόσο ήταν εύστοχη/άστοχη προσπάθεια, ώστε τελικά να βγαίνουν και τα αντίστοιχα % ποσοστά ευστοχίας.

51. Να γραφεί πρόγραμμα που εμφανίζει τους 256 (2^8) διαφορετικούς δυαδικούς αριθμούς με 8 bits (dyadiko_systhma_8bits.psc):

```
1 - 0 0 0 0 0 0 0 0
2 - 0 0 0 0 0 0 0 1
3 - 0 0 0 0 0 0 1 0
4 - 0 0 0 0 0 0 1 1
5 - 0 0 0 0 0 1 0 0
6 - 0 0 0 0 0 1 0 1
7 - 0 0 0 0 0 1 1 0
8 - 0 0 0 0 0 1 1 1
248 - 1 1 1 1 0 1 1 1
249 - 1 1 1 1 1 0 0 0
250 - 1 1 1 1 1 0 0 1
251 - 1 1 1 1 1 0 1 0
252 - 1 1 1 1 1 0 1 1
253 - 1 1 1 1 1 1 0 0
254 - 1 1 1 1 1 1 0 1
255 - 1 1 1 1 1 1 1 0
256 - 1 1 1 1 1 1 1 1
```

53. Να δημιουργήσετε πρόγραμμα το οποίο να εμφανίζει, να μετράει και να αθροίζει όλους τους τριψήφιους θετικούς ακέραιους αριθμούς των οποίων τουλάχιστον δύο από τα ψηφία τους είναι ο αριθμός 8 (AtLeast2Eights.psc)

```
cnt = 29 sum = 22667
```

54. Να γραφεί πρόγραμμα το οποίο να εμφανίζει τους 18 θετικούς ακέραιους (1-1000) των οποίων το γινόμενο των ψηφίων τους είναι το διπλάσιο του αθροίσματός τους (ginomeno_pshfiwn_2plasio_athroismatos.psc):

36, 44, 63, 138, 145, 154, 183, 224, 242, 318, 381, 415, 422, 451, 514, 541, 813, 831

55. Πολλαπλασιαστική επιμονή ενός ακέραιου είναι ο αριθμός των βημάτων που απαιτούνται ώστε το γινόμενο των ψηφίων του να είναι μονοψήφιος αριθμός. Π.χ. η πολλαπλασιαστική επιμονή του αριθμού 679 είναι 5, διότι: $679 \rightarrow 6*7*9=378 \rightarrow 3*7*8=168 \rightarrow 1*6*8=48 \rightarrow 4*8=32 \rightarrow 3*2=6$. Να γραφεί πρόγραμμα (pollaplasiastikh_epimoneh.psc) το οποίο να διαβάζει έναν θετικό ακέραιο και να εμφανίζει την πολλαπλασιαστική επιμονή του καθώς και τα βήματα υπολογισμού της:

```
679
-----
1 . 378
2 . 168
3 . 48
4 . 32
5 . 6
-----
pollaplasiastikh_epimoneh = 5
```

Προέκταση: βρείτε τον θετικό ακέραιο (≤ 100) με τη μεγαλύτερη πολλαπλασιαστική επιμονή. (pollaplasiastikh_epimoneh2.psc)

```
max pollaplasiastikh_epimoneh = 4 twv arithmwv:
77
```

56. Να γραφεί πρόγραμμα το οποίο να εμφανίζει τους 56 θετικούς 3ψήφιους ακέραιους με όλα τα ψηφία τους να αποτελούν διαιρέτες του αριθμού (*ola_ta_pshfia_diairoyn_ton_arithmo.psc*)

1 . 111	45 . 672
2 . 112	46 . 728
3 . 115	47 . 735
4 . 122	48 . 777
5 . 124	49 . 784
6 . 126	50 . 816
7 . 128	51 . 824
8 . 132	52 . 848
9 . 135	53 . 864
10 . 144	54 . 888
11 . 155	55 . 936
12 . 162	56 . 999

57. Στα μαθηματικά η σειρά Fibonacci είναι η εξής: 0, 1, 1, 2, 3, 5, 8, 13, 21... Εξ ορισμού, οι δύο πρώτοι αριθμοί της σειράς είναι το 0 και το 1, ενώ καθένας από τους επόμενους αριθμούς είναι το άθροισμα των δύο προηγούμενων. Να δημιουργήσετε πρόγραμμα (*fibonacci.psc*) το οποίο να διαβάζει έναν ακέραιο αριθμό N και να εμφανίζει τους πρώτους N αριθμούς της σειράς Fibonacci στην οθόνη. Παράδειγμα εισόδου: 10, Παράδειγμα εξόδου: 0 1 1 2 3 5 8 13 21 34

58. Αριθμός *ariadne* ορίζεται ο αριθμός που το άθροισμα του πιο σημαντικού (ψηφίο στα αριστερά) και του λιγότερου σημαντικού ψηφίου (ψηφίο στα δεξιά) ισούται με το άθροισμα των υπόλοιπων ψηφίων (π.χ. 121, 1335). Να γραφεί πρόγραμμα (*ariadne.psc*) το οποίο να εμφανίζει

- τους 45 θετικούς 3ψήφιους ακέραιους που είναι αριθμοί *ariadne*:
- τους 63 θετικούς 4ψήφιους ακέραιους (από 1000 - 2000) που είναι αριθμοί *ariadne*:

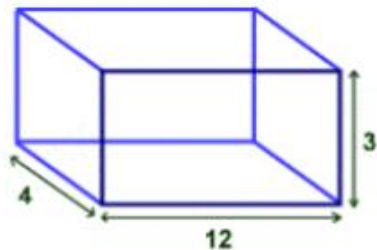
1 . 110	34 . 583
2 . 121	35 . 594
3 . 132	36 . 660
4 . 143	37 . 671
5 . 154	38 . 682
6 . 165	39 . 693
7 . 176	40 . 770
8 . 187	41 . 781
9 . 198	42 . 792
10 . 220	43 . 880
11 . 231	44 . 891
12 . 242	45 . 990

1 . 1010	52 . 1627
2 . 1021	53 . 1638
3 . 1032	54 . 1649
4 . 1043	55 . 1706
5 . 1054	56 . 1717
6 . 1065	57 . 1728
7 . 1076	58 . 1739
8 . 1087	59 . 1807
9 . 1098	60 . 1818
10 . 1100	61 . 1829
11 . 1111	62 . 1908
12 . 1122	63 . 1919

61. Να γραφεί πρόγραμμα το οποίο εμφανίζει τους 10 πρώτους όρους των παρακάτω **ακολουθιών** (*series.psc*):

- 1, 3, 6, 10, 15, 21, 28, 36, 45, 55
- 1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800
- 5, 10, 13, 26, 29, 58, 61, 122, 125, 250
- 1, 2, 4, 8, 16, 32, 64, 128, 256, 512
- 8, 24, 12, 36, 18, 54, 27, 81, 40, 120

69. Ο όγκος ενός ορθογώνιου παραλληλεπιπέδου ισούται με το γινόμενο των τριών διαστάσεών του (μήκος x πλάτος x ύψος). Να γραφεί πρόγραμμα το οποίο διαβάζει τον όγκο ενός ορθογώνιου παραλληλεπιπέδου και εμφανίζει όλες τις πιθανές ακέραιες διαστάσεις του. (*ogkos_orthogwniou_parallhleripedoy.psc*)

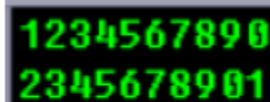
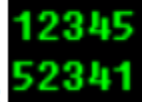


Λίγη γεωμετρία:

Όγκος κουτιού: μήκος x πλάτος x ύψος = $12 \times 4 \times 3 = 144 \text{cm}^3$

```
10
1 . 1 1 10
2 . 1 2 5
3 . 1 5 2
4 . 1 10 1
5 . 2 1 5
6 . 2 5 1
7 . 5 1 2
8 . 5 2 1
9 . 10 1 1
```

```
12
1 . 1 1 12
2 . 1 2 6
3 . 1 3 4
4 . 1 4 3
5 . 1 6 2
6 . 1 12 1
7 . 2 1 6
8 . 2 2 3
9 . 2 3 2
10 . 2 6 1
11 . 3 1 4
12 . 3 2 2
13 . 3 4 1
14 . 4 1 3
15 . 4 3 1
16 . 6 1 2
17 . 6 2 1
18 . 12 1 1
```

72. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και υπολογίζει τον ακέραιο που προκύπτει εάν το 1ο ψηφίο γίνει τελευταίο (*prwtoPshfioTeleytaio.psc*). . Προέκταση: να κάνει αντιμετάθεση του 1^{ου} με το τελευταίο ψηφίο:  (*swapPrwtoPshfioMeTeleytaio.psc*)

75. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και υπολογίζει τον ακέραιο που προκύπτει εάν κάθε ψηφίο υψωθεί στο τετράγωνο. Π.χ. 9119 → 811181, 2483 → 416649, 3212 → 9414 (*squareDigits.psc*)

79. **Ετερομήκιος** (*heteromecic*) είναι ένας ακέραιος που ισούται με το γινόμενο 2 συνεχόμενων ακεραίων. Είναι δηλ. της μορφής $n*(n+1)$. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και ελέγχει εάν είναι ετερομήκιος εμφανίζοντας τους 2 αντίστοιχους συνεχόμενους ακέραιους (*heteromecicNumber.psc* πιο αποδοτικά: *heteromecicNumber2.psc*). π.χ. 9702 ==> Yes: 9702 = 98 * 99

86. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο N και εμφανίζει όλους του "δίδυμους" πρώτους αριθμούς έως και το N. Δηλ. τους αριθμούς που είναι πρώτοι (διαιρούνται μόνο με το 1 και τον εαυτό τους) και διαφέρουν κατά 2. π.χ. οι "δίδυμοι" πρώτοι μέχρι το 200 είναι οι: 5 7, 11 13, 17 19, 29 31, 41 43, 59 61, 71 73, 101 103, 107 109, 137 139, 149 151, 179 181, 191 193, 197 199 (*prwtoi_didymoi_arithmoi.psc*)

90. Υπάρχουν 6 θετικοί τριψήφιοι ακέραιοι με άθροισμα ψηφίων ίσο με το γινόμενό τους. Να γραφεί πρόγραμμα που τους εμφανίζει. (*athroisma_ginomeno_pshfiwn_isa.psc*)

94. Πρόγραμμα που εμφανίζει το παρακάτω (trigwno_arithmwon(1-5).psc):

α) 1
12
123
1234
12345
123456
1234567
12345678
123456789
12345678
1234567
123456
12345
1234
123
12
1

β) 9
98
987
9876
98765
987654
9876543
987654321
987654321
98765432
9876543
987654
98765
9876
987
98
9

γ) 1
21
321
4321
54321
654321
7654321
87654321
987654321
87654321
7654321
654321
54321
4321
321
21
1

δ) 1
11
111
1111
11111
111111
1111111
11111111
11111111
11111111
1111111
111111
11111
1111
111
11
1

ε) 8

8
88
888
8888
88888
888888
8888888
88888888
88888888
88888888
88888888
8888888
888888
88888
8888
888
88
8

95. Ο αριθμός e (**αριθμός Όιλερ**) είναι σημαντική μαθηματική σταθερά, η οποία αποτελεί τη βάση του φυσικού λογαρίθμου (wikipedia). Μπορεί επίσης να υπολογιστεί ως το άθροισμα της άπειρης σειράς

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1} + \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} + \dots$$

Η αριθμητική αξία του e μέχρι τα 50 δεκαδικά ψηφία είναι 2,71828182845904523536028747135266249775724709369995. Να γραφεί πρόγραμμα που υπολογίζει την τιμή του e για τους 20 πρώτους όρους της παραπάνω σειράς

(e_arithmos_Oiler.psc). **2.718281828459040**

96. Ένας 13ψήφιος ραβδωτός κώδικας (**barcode**) έχει 12 ψηφία και ένα 13ο το ψηφίο ελέγχου που υπολογίζεται βάση των 12 πρώτων, ως εξής: προσθέτουμε τα 12 ψηφία με τα ψηφία στις μονές θέσεις ως έχουν και τα ψηφία στις ζυγές θέσεις πολλαπλασιασμένα επί 3. Από το αποτέλεσμα κρατάμε το τελευταίο ψηφίο. Αν αυτό είναι μηδέν τότε το ψηφίο ελέγχου είναι το μηδέν. Διαφορετικά, το ψηφίο ελέγχου είναι το αποτέλεσμα που προκύψει εάν αφαιρέσουμε από το 10 το τελευταίο ψηφίο. Να γραφεί πρόγραμμα που διαβάσει τα 12 πρώτα ψηφία ενός ραβδωτού κώδικα και υπολογίζει το 13ο ψηφίο ελέγχου. (barcode.psc)



99. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και θα ελέγχει αν είναι ή δεν είναι της μορφής aa . Π.χ. οι 123123, 56215621 είναι της μορφής aa , ενώ οι 123321, 12312 δεν είναι (arithmos_aa.psc)

103. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και εμφανίζει το πλήθος των καταληκτικών μηδενικών του (trailing_zeros.psc). π.χ. 120300(2), 1203(0)

104. Στα μαθηματικά, ένας αυτομορφικός αριθμός είναι ένας θετικός ακέραιος του οποίου το τετράγωνο "τελειώνει" με τα ίδια ψηφία με τον ίδιο τον αριθμό. π.χ. ο 376: $376^2 = 141376$. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και ελέγχει αν είναι αυτομορφικός (aytomorfikos_arithmos.psc). Π.χ. 0, 1, 5, 6, 25, 76, 376, 625, 9376, 90625, 109376, 890625, 2890625, 7109376, 12890625, 87109376, 212890625, 787109376, 1787109376, 8212890625, ...

105. Στα μαθηματικά, ένας ετερομεκτικός (heteromecic) αριθμός είναι ένας θετικός ακέραιος που ισούται με το γινόμενο 2 συνεχόμενων ακεραίων. Είναι δηλ. της μορφής $n*(n+1)$. π.χ. ο $342 = 18*19$. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και ελέγχει αν είναι ετερομεκτικός (heteromecic_arithmos.psc).

106. Να γραφεί πρόγραμμα που διαβάζει 3 θετικούς ακέραιους και εμφανίζει τον μεγαλύτερο χωρίς χρήση των δομών Αν και Επίλεξε (max3WithoutIf.psc). Παρομοίως να εμφανίζει τον μικρότερο.

111. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο και εμφανίζει το **πρώτο και το τελευταίο ψηφίο** του (prwto_pshfio.psc)

115. Πρόγραμμα το οποίο διαβάζει έναν θετικό ακέραιο και x και ελέγχει αν αποτελεί **δύναμη του 2** και αν ναι, ποια είναι αυτή (dynamh_toy_2.psc). Π.χ. $x = 256 \rightarrow$ ναι 2^8

116. Πρόγραμμα που εμφανίζει όλους τους 3ψήφιους θετικούς ακέραιους που **ισούνται με το γινόμενο του αθροίσματος επί του γινομένου των ψηφίων τους** (tripshfioi_isei_me_athroisma_epi_ginomeno_pshfiwn.psc). π.χ. $135 = (1+3+5) \times (1 \times 3 \times 5)$

121. Πρόγραμμα που διαβάζει έναν ακέραιο x (≥ 1) και υπολογίζει τη μεγαλύτερη δύναμη του 2 (n) που είναι μικρότερη ή ίση του x (max_dynamh_toy_2.psc)

```
70000
16 because 2^16 = 65536 <= 70000
```

123. Πρόγραμμα που διαβάζει έναν ακέραιο και ελέγχει αν όλα τα ψηφία του είναι ίσα. π.χ. 999, 11111, 33 (ola_ta_pshfia_isa.psc)

131. Να γραφεί τμήμα προγράμματος ισοδύναμο με το παρακάτω χωρίς τη χρήση των δομών Αν/Επίλεξε (An_Oso.psc):

```
ΔΙΑΒΑΣΕ x, y
ΑΝ x < y ΤΟΤΕ
  ΓΡΑΨΕ -1
ΑΛΛΙΩΣ_ΑΝ x > y ΤΟΤΕ
  ΓΡΑΨΕ 1
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 0
ΤΕΛΟΣ_ΑΝ
```

133. Να γραφεί πρόγραμμα το οποίο διαβάζει έναν θετικό ακέραιο και ελέγχει εάν έχει τα ψηφία του σε αύξουσα σειρά. π.χ. 12489. (pshfiaSeAyksoysaSeira.psc)

```
60 : 12
72 : 12
84 : 12
90 : 12
96 : 12
48 : 10
80 : 10
36 : 9
100 : 9
24 : 8
...
43 : 2
47 : 2
53 : 2
59 : 2
61 : 2
67 : 2
71 : 2
73 : 2
79 : 2
83 : 2
```

128. **Ευγενείς αριθμοί (Polite)**: ορίζονται οι θετικοί ακέραιοι αριθμοί που μπορούν να γραφτούν ως το άθροισμα δύο ή περισσότερων διαδοχικών θετικών ακεραίων. π.χ. $6=1+2+3$, $14=2+3+4+5$, το 15 γράφεται με 2 τρόπους $15=1+2+3+4+5=7+8$, ενώ το 16 δεν είναι ευγενής αριθμός. Να γράψετε πρόγραμμα το οποίο: να διαβάζει έναν θετικό ακέραιο αριθμό a . Να ελέγχει αν ο a αποτελεί Ευγενή αριθμό. Αν αυτό δεν συμβαίνει να εμφανίζει σχετικό μήνυμα διαφορετικά να εμφανίζει τους διαδοχικούς θετικούς ακέραιους των οποίων το άθροισμα ισούται με τον αριθμό a . Για αριθμούς που γράφονται ως άθροισμα διαδοχικών ακεραίων με περισσότερους από έναν τρόπους, να εμφανίζει τη σειρά με το μεγαλύτερο πλήθος διαδοχικών όρων. (π.χ. για το 15 τη σειρά 1,2,3,4,5 και όχι 7,8). Υπόδειξη με 2 παραδείγματα: α) Έστω ότι θέλουμε να ελέγξουμε τον αριθμό 7. Ελέγχουμε διαδοχικά τα αθροίσματα: $1+2$, $1+2+3$, $1+2+3+4$, εδώ σταματάμε γιατί έχουμε $\text{άθροισμα}=10>7$. Ξεκινάμε πάλι ελέγχοντας $2+3$, $2+3+4$, εδώ σταματάμε γιατί έχουμε $\text{άθροισμα}=9>7$. Ξεκινάμε πάλι $3+4=7$, τέλος, άρα το 7 είναι ευγενής αριθμός. β) Έστω ότι θέλουμε να ελέγξουμε τον αριθμό 16. Ελέγχουμε διαδοχικά τα αθροίσματα: $1+2$, $1+2+3$, $1+2+3+4$, $1+2+3+4+5$, $1+2+3+4+5+6$, εδώ σταματάμε γιατί έχουμε $\text{άθροισμα}=21>16$. Ξεκινάμε πάλι ελέγχοντας $2+3$, $2+3+4$, $2+3+4+5$, $2+3+4+5+6$, εδώ σταματάμε γιατί έχουμε $\text{άθροισμα}=20>16$. Ξεκινάμε πάλι $3+4$, $3+4+5$, $3+4+5+6=18>16$ άρα σταματάμε. Ξεκινάμε πάλι ελέγχοντας $4+5$, $4+5+6$, $4+5+6+7=22>16$ άρα σταματάμε. Ξεκινάμε πάλι ελέγχοντας $5+6$, $5+6+7=18>16$. Ξανά $6+7$, $6+7+8=21>16$ άρα σταματάμε. Ξανά $7+8$, $7+8+9=24>16$ stop. Ξανά $8+9=17>16$. Εδώ τελειώνουμε διότι το επόμενο άθροισμα που πρέπει να ελέγξουμε είναι το $9+10=19>16$ και όλες οι επόμενες διαδοχές $10+11$, $11+10$, ... θα μας δίνουν όλο και μεγαλύτερα αθροίσματα. (Σκεφτείτε γενικότερα πότε πρέπει να τελειώνουμε τον έλεγχο για να κάνουμε πιο αποδοτικό τον αλγόριθμό μας). Άρα το 16 δεν είναι ευγενής αριθμός. (polite_numbers.psc)

```
Give a number
15
Polite number. Sequence found:
1 + 2 + 3 + 4 + 5
Polite number. Sequence found:
4 + 5 + 6
Polite number. Sequence found:
7 + 8
-----
max sequence:
1 + 2 + 3 + 4 + 5
```

136. Να γραφεί πρόγραμμα το οποίο διαβάζει έναν ακέραιο (1-39) και τον εμφανίζει στη ρωμαϊκή γραφή (ArabicToRoman.psc):

Arabic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
Roman	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV	XVI	XVII	XVIII	XIX	XX	XXI	XXII	XXIII	XXIV	XXV	XXVI	XXVII	XXVIII	XXIX	XXX	XXXI	XXXII	XXXIII	XXXIV	XXXV	XXXVI	XXXVII	XXXVIII	XXXIX

137. Εμφανίστε επαναληπτικά τις παρακάτω δυνάμεις του 2 (DynamisToyDyo.psc):

```

1 . 1 ^2 = 1
2 . 11 ^2 = 121
3 . 111 ^2 = 12321
4 . 1111 ^2 = 1234321
5 . 11111 ^2 = 123454321
6 . 111111 ^2 = 12345654321
7 . 1111111 ^2 = 1234567654321

```

```

1 . 3 ^2 = 9
2 . 33 ^2 = 1089
3 . 333 ^2 = 110889
4 . 3333 ^2 = 11108889
5 . 33333 ^2 = 1111088889
6 . 333333 ^2 = 111110888889
7 . 3333333 ^2 = 11111108888889

```

```

1 . 6 ^2 = 36
2 . 66 ^2 = 4356
3 . 666 ^2 = 443556
4 . 6666 ^2 = 44435556
5 . 66666 ^2 = 4444355556
6 . 666666 ^2 = 444443555556
7 . 6666666 ^2 = 44444435555556

```

```

1 . 9 ^2 = 81
2 . 99 ^2 = 9801
3 . 999 ^2 = 998001
4 . 9999 ^2 = 99980001
5 . 99999 ^2 = 9999800001
6 . 999999 ^2 = 999998000001
7 . 9999999 ^2 = 99999980000001

```

Πίνακες

- Πρόγραμμα που διαβάζει έναν ακέραιο πίνακα $A[10]$ και εμφανίζει τη μέγιστη τιμή του και των πλήθος των εμφανίσεών της (maxA10.psc)
- Πρόγραμμα εισαγωγής σε πίνακες της μάζας και της θερμοκρασίας 10 σωμάτων και υπολογισμός του σταθμικού μέσου όρου θερμοκρασίας σταθμικός μέσος = $(\mu_1 \times \theta_1 + \mu_2 \times \theta_2 + \dots + \mu_{10} \times \theta_{10}) / (\mu_1 + \mu_2 + \dots + \mu_{10})$ (stathmikosMesos.psc)
- Πρόγραμμα που διαβάζει έναν ακέραιο $A[10]$ και μία τιμή αναζήτησης. Με τη σειριακά αναζήτηση ψάχνει την τιμή στον πίνακα (seiriakh.psc)
- Πρόγραμμα αναζήτησης τιμής σε ακέραιο $A[100]$ ταξινομημένο σε φθίνουσα σειρά με τον αλγόριθμο της δυαδικής αναζήτησης (binary_search.psc)
- Εισαγωγή τιμών σε ακέραιο $A[10]$. Εισαγωγή μίας τιμής αναζήτησης. Εύρεση του πλήθους εμφάνισης της τιμής αναζήτησης στον $A[10]$ (search_all.psc)
- Πρόγραμμα που διαβάζει έναν ακέραιο $A[10]$ και τον εμφανίζει ταξινομημένο κατά αύξουσα σειρά (sort.psc)
- Πρόγραμμα συγχώνευσης 2 πινάκων $A[3]$, $B[5]$ σε 3ο $\Gamma[8]$ (merge.psc)
- Πρόγραμμα που διαβάζει έναν ακέραιο $A[10]$ και τον εμφανίζει ταξινομημένο κατά αύξουσα σειρά με τον αλγόριθμο της έξυπνης φυσαλίδας (eksyprnh_fysalida.psc)
- Πρόγραμμα που διαβάζει έναν ακέραιο $A[10]$ και τον εμφανίζει ταξινομημένο κατά αύξουσα σειρά με τον αλγόριθμο ταξινόμησης selection sort (selection_sort.psc)

- Πρόγραμμα που διαβάζει έναν θετικό ακέραιο και εμφανίζει πόσες φορές περιέχει καθένα από τα ψηφία 0-9 (μόνο για αυτά που περιέχει. Π.χ. Δώσε έναν θετικό ακέραιο: 10003423 ==> Ο αριθμός περιέχει 3 φορές το ψηφίο: 0, 1 φορές το ψηφίο: 1, 1 φορές το ψηφίο: 2, 2 φορές το ψηφίο: 3, 1 φορές το ψηφίο: 4 (NumberAnalyzer.psc)
- Πρόγραμμα το οποίο γεμίζει έναν ακέραιο τετραγωνικό πίνακα C[8,8] με 1 τις δύο κύριες διαγωνιές του και με 0 τα υπόλοιπα στοιχεία του. Να τον εμφανίζει σαν πίνακα ως εξής (diagwnioi_pirakaka.psc):

Πρόγραμμα που διαβάζει τις ενδείξεις ρίψης ενός ζαριού (1-6) μέχρι να δοθεί η τιμή -1. Να εμφανίζει τη συχνότητα εμφάνισης της κάθε ένδειξης (1-6)

```

1 : 2 φορές
2 : 2 φορές
3 : 4 φορές
4 : 2 φορές
5 : 2 φορές
6 : 2 φορές

```

(pinakas_syxnothtwn.psc)

C	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0
3	0	0	1	0	0	1	0	0
4	0	0	0	1	1	0	0	0
5	0	0	0	1	1	0	0	0
6	0	0	1	0	0	1	0	0
7	0	1	0	0	0	0	1	0
8	1	0	0	0	0	0	0	1

! Δίνεται ο ακέραιος A[11,11] γεμάτος με μηδενικά εκτός
! των διαγωνίων του που είναι γεμάτος με την τιμή 1
! Διαβάζει έναν ακέραιο T[3,3] και τον αναζητά στον A[11,11] π.χ.

T

1	0	1
0	1	0
1	0	1

(anazhtsh_pinaka_se_pinaka.psc)

A

	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	0	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	0	1	0
3	0	0	1	0	0	0	0	0	1	0	0
4	0	0	0	1	0	0	0	1	0	0	0
5	0	0	0	0	1	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	1	0	1	0	0	0	0
8	0	0	0	1	0	0	0	1	0	0	0
9	0	0	1	0	0	0	0	0	1	0	0
10	0	1	0	0	0	0	0	0	0	1	0
11	1	0	0	0	0	0	0	0	0	0	1

Βρέθηκε στη θέση 5, 5

27. (Άσκηση 2.18 Βιβλίο Πληροφορικής Κύπρου) Η Αριάδνη έχει κρύψει τον κωδικό της μέσα σε ένα κείμενο. Δεν είναι σίγουρη όμως αν το έκανε σωστά. Να δημιουργήσετε πρόγραμμα το οποίο να δέχεται χαρακτήρα προς χαρακτήρα τον κωδικό (4 χαρακτήρες) και το κείμενο (10 χαρακτήρες). Αν ο κωδικός υπάρχει μέσα στο κείμενο να εμφανίζει τις θέσεις των χαρακτήρων του κειμένου που εμφανίζονται οι χαρακτήρες του κωδικού. Διαφορετικά, να εμφανίζει κατάλληλο μήνυμα. Οι χαρακτήρες του κωδικού πρέπει να εμφανίζονται μέσα στη συμβολοσειρά με τη σωστή σειρά, αλλά όχι κατά ανάγκη ο ένας δίπλα από τον άλλο (krymmenos_kwdikos.psc). π.χ. κωδικός='STOP', κείμενο='ABDSTKLOTP' →

Βρέθηκε :
4
5
8
10

33. Καταχωρείστε σε πίνακα N[9] τα 9 (χαρτο)νομίσματα του €: 500, 200, ..., 5, 2, 1. Εύρεση ελάχιστου αριθμού κερμάτων και χαρτονομισμάτων που ισοδυναμούν με ένα ποσό σε € (minimum_coins.psc).

789

```
-----
1 των 500 €
1 των 200 €
1 των 50 €
1 των 20 €
1 των 10 €
1 των 5 €
2 των 2 €
```

42. Να γραφεί πρόγραμμα που διαβάζει ένα θετικό ακέραιο αριθμό και κάνει καταμέτρηση των ψηφίων του π.χ. ως εξής (syxnohtes_pshfiwn_arithmoy.psc):

```
457265200099
Ψηφίο: 0 Πλήθος: 3
Ψηφίο: 2 Πλήθος: 2
Ψηφίο: 4 Πλήθος: 1
Ψηφίο: 5 Πλήθος: 2
Ψηφίο: 6 Πλήθος: 1
Ψηφίο: 7 Πλήθος: 1
Ψηφίο: 9 Πλήθος: 2
```

46. Να γραφεί πρόγραμμα που διαβάζει έναν πίνακα ακεραίων μονοδιάστατο 10 θέσεων και υπολογίζει την επικρατούσα τιμή, δηλαδή την τιμή(ές) που εμφανίζεται τις περισσότερες φορές. (epikratoysa_timh.psc) π.χ. με είσοδο των τιμών: 7, 4, 3, 7, 3, 3, 6, 7, 1, 9 → Επικρατούσα: 3, Επικρατούσα: 7

Υπόδειξη: ταξινομήστε τον πίνακα

47. Να γραφεί πρόγραμμα που διαβάζει έναν πίνακα ακεραίων μονοδιάστατο 10 θέσεων και υπολογίζει τις 3 μεγαλύτερες διαφορετικές τιμές (αν υπάρχουν). (max_different.psc) Παραδείγματα εκτέλεσης:

1	1	1	1
1	1	1	2
1	1	1	3
2	2	1	4
2	2	1	5
2	2	1	7
2	2	1	7
2	2	1	8
2	2	1	10
3	2	1	10
max= 3	max= 2	max= 1	max= 10
max= 2	max= 1		max= 8
max= 1			max= 7

50. Να γραφεί πρόγραμμα που διαβάσει σε πίνακα A[N], N ακέραιες τιμές και ελέγχει (ΝΑΙ/ΟΧΙ) εάν είναι όλες διαφορετικές μεταξύ τους. (ola_diaforetika.psc). Π.χ. για N=5:

```

1
2
3
5
6
different values: 5
All Different!
1
2
3
1
2
different values: 3

```

59. Να γραφεί πρόγραμμα το οποίο για όλες τις πιθανές τιμές ρίψης 3 ζαριών, να εμφανίζει: α) τη μεγαλύτερη συχνότητα εμφάνισης αθροίσματος, β) τα πιο συχνά αθροίσματα και γ) τις τιμές των ζαριών με τα αθροίσματα αυτά (tria_zaria.psc).

```

max sykhnothta athroismatos 3 zariwn = 27 twn athroismatwn:
10
11
Times 3 zariwn me ta pio syxna athroismata:
-----
1 + 3 + 6 + = 10
1 + 4 + 5 + = 10
1 + 4 + 6 + = 11
1 + 5 + 4 + = 10
1 + 5 + 5 + = 11

```

```

5 + 4 + 1 + = 10
5 + 4 + 2 + = 11
5 + 5 + 1 + = 11
6 + 1 + 3 + = 10
6 + 1 + 4 + = 11
6 + 2 + 2 + = 10
6 + 2 + 3 + = 11
6 + 3 + 1 + = 10
6 + 3 + 2 + = 11
6 + 4 + 1 + = 11

```

71. Ένας πίνακας $A[N]$ λέγεται μονοκόρυφος εάν για κάποιο στοιχείο p ($2 \leq p \leq N-1$) έχουμε $A[1] < A[2] < \dots < A[p]$ και $A[p] > A[p+1] > \dots > A[N]$. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο $A[10]$ και ελέγχει εάν είναι μονοκόρυφος ή όχι. Σε περίπτωση που είναι να εμφανίζει το στοιχείο p (`monokoryfos1D.psc`) π.χ.

```
A = [ 1  2  3  4  5  5  3  2  1  0 ]
oxi monokoryfos
```

```
A = [ 1  2  3  4  5  4  3  2  1  0 ]
monokoryfos sto stoixeiο A[ 5 ]= 5
```

73. Να γραφεί πρόγραμμα που αρχικά γεμίζει έναν πίνακα χαρακτήρων $MC[10]$ με τον κώδικα Μορς των ψηφίων 0-9 και στη συνέχεια διαβάζει έναν θετικό ακέραιο και εμφανίζει τα ψηφία του κωδικοποιημένα στον κώδικα Μορς (`morse_code.psc`).

1	●	■	■	■	■
2	●	●	■	■	■
3	●	●	●	■	■
4	●	●	●	●	■
5	●	●	●	●	●
6	■	●	●	●	●
7	■	■	●	●	●
8	■	■	■	●	●
9	■	■	■	■	●
0	■	■	■	■	■

1970
· - - - -
- - - - - ·
- - · · ·
- - - - -

77. Ως **πανψηφιακός αριθμός** ονομάζεται ο φυσικός αριθμός στον οποίο περιέχονται τουλάχιστον μια φορά το κάθε ένα ψηφίο (0-9). 76. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και ελέγχει εάν είναι

1234509876123
ΑΛΗΘΗΣ

πανψηφιακός (*pandigitalNumber.psc*). π.χ.

78. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο $A[10]$ και τον **ταξινομεί με βάση τον αριθμό των ψηφίων** κάθε τιμής και κατά φθίνουσα σειρά, ενώ οι τιμές με τον ίδιο αριθμό ψηφίων τα ταξινομούνται με αύξουσα σειρά. Π.χ. 3242, 435, 4534, 2, 8, 9, 567, 23456, 98765, 23 → 23456, 98765, 3242, 4534, 435, 567, 23, 2, 8, 9 (*sortByNumOfDigits.psc*)

81. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο $A[N]$ και εντοπίζει τα στοιχεία του που είναι "σημεία ισορροπίας". Δηλ. το άθροισμα των στοιχείων στα αριστερά ισούται με το άθροισμα των στοιχείων στα δεξιά. π.χ. για τα στοιχεία 2,3,4,1,4,5 σημείο ισορροπίας είναι το 4ο (1) διότι τα στοιχεία στα αριστερά (2,3,4) και τα στοιχεία στα δεξιά (4,5) έχουν το ίδιο άθροισμα (9). (*balanceElementOfArray.psc*)

82. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο τριψήφιο και εμφανίζει τη **λεκτική περιγραφή** του (*sayTheNumber.psc*).

π.χ.

658 Εξακόσια πενήντα οκτώ	720 Επτακόσια είκοσι	309 Τριακόσια εννέα
400 Τετρακόσια	211 Διακόσια έντεκα	712 Επτακόσια δώδεκα

83. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και εμφανίζει τον μεγαλύτερο ακέραιο που μπορεί να φτιαχτεί χρησιμοποιώντας τα ψηφία του αρχικού.

4353543901
Μέγιστος ακέραιος: 9554433310

π.χ.

Υπόδειξη (λύση 1^η): συλλέξτε τα ψηφία του ακεραίου σε πίνακα $A[100]$, ταξινομήστε τα κατά φθίνουσα σειρά και σχηματίστε τον ζητούμενο ακέραιο (*makeMaxNumber.psc*).

Υπόδειξη (λύση 2^η): φτιάξτε τον πίνακα συχνοτήτων $A[10]$ με τις συχνότητες των ψηφίων του ακεραίου ($A[1]$: οι άσσοι, ..., $A[9]$: τα εννιάρια, $A[10]$: τα μηδενικά) και σχηματίστε τον ζητούμενο ακέραιο από τα εννιάρια μέχρι τους άσσους και τελευταία τα μηδενικά (*makeMaxNumber2.psc*)

85. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο $A[N]$ (N : σταθερά) και τον ανακατεύει στο μέγιστο βαθμό. Τοποθετεί δηλ. στο $A[1]$ το 1ο μεγαλύτερο, στο $A[2]$ το 1ο μικρότερο, στο $A[3]$ το 2ο μεγαλύτερο, στο $A[4]$ το 2ο μικρότερο κ.ο.κ. π.χ. εάν ο A έχει τις τιμές: 1,2,3,4,5,6,7,8,9,10 καταλήγει με τις τιμές: 10,1,9,2,8,3,7,4,6,5 (anakatemaPinaka.psc)

87. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο και ελέγχει αν στον αριθμό κάθε ψηφίο εμφανίζεται μία και μοναδική φορά. Για παράδειγμα ο αριθμός 38213 περιέχει 2 φορές το ψηφίο 3 ενώ στον αριθμό 953 κάθε ψηφίο εμφανίζεται μία και μοναδική φορά (pshfia_aro_mia_fora.psc).

88. Ποια μέρα είναι Χριστούγεννα; Έστω ET η χρονολογία που μας ενδιαφέρει. Χωρίζουμε το ET σε δύο μέρη: ονομάζουμε A τις εκατονταετίες της χρονολογίας, δηλαδή τα 2 αριστερά ψηφία του ET . Ονομάζουμε B τα υπόλοιπα χρόνια της χρονολογίας, δηλαδή τα 2 δεξιά ψηφία του ET . Διαιρούμε το A με τον αριθμό 4 και κρατάμε μόνο το ακέραιο μέρος. Αυτό το ονομάζουμε K . Διαιρούμε το B με τον αριθμό 4 και κρατάμε μόνο το ακέραιο μέρος. Αυτό το ονομάζουμε L . Για να βρούμε την ημέρα της εβδομάδας εφαρμόζουμε τον τύπο: $X = 50 + B + K + L - 2A$. Διαιρούμε το X με το 7 και το υπόλοιπο το ονομάζουμε M . Αν $M = 0$ σημαίνει ότι τα Χριστούγεννα θα πέσουν Κυριακή, αν $M = 1$ θα πέσουν Δευτέρα, ... κ.ο.κ. Να γραφεί πρόγραμμα που εμφανίζει την ημέρα των Χριστουγέννων από το 2000 μέχρι το 2025 (christmas_day.psc).

2020	Παρασκευή
2021	Σάββατο
2022	Κυριακή
2023	Δευτέρα
2024	Τετάρτη
2025	Πέμπτη

97. Να γραφεί πρόγραμμα που γεμίζει έναν ακέραιο $A[N, N]$ (N : περιττός) με τις τιμές $1-N^2$ και τον εμφανίζει «κατά δακτυλίου» δεξιόστροφα από το κέντρο και βγαίνοντας προς τα έξω ξεκινώντας από το επάνω αριστερά στοιχείο του κάθε δακτυλίου (sarwsh_kata_daktyliouy.psc). Π.χ.

```
Gemisma pinaka. Please wait...
Pinakas A( 3 x 3 ):
-----
```

```
1  2  3
4  5  6
7  8  9
-----
```

```
sarwsh_kata_daktyliouy:
daktylios 1: 5
daktylios 2 : 1 2 3 6 9 8 7 4
```

```
Gemisma pinaka. Please wait...
Pinakas A( 7 x 7 ):
-----
```

```
1  2  3  4  5  6  7
8  9  10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35
36 37 38 39 40 41 42
43 44 45 46 47 48 49
-----
```

```
sarwsh_kata_daktyliouy:
daktylios 1: 25
daktylios 2 : 17 18 19 26 33 32 31 24
daktylios 3 : 9 10 11 12 13 20 27 34 41 40 39 38 37 30 23 16
daktylios 4 : 1 2 3 4 5 6 7 14 21 28 35 42 49 48 47 46 45 44 43 36 29 22 15 8
```

```
Gemisma pinaka. Please wait...
Pinakas A( 9 x 9 ):
-----
```

```
1  2  3  4  5  6  7  8  9
10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54
55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80 81
-----
```

```
sarwsh_kata_daktyliouy:
daktylios 1: 41
daktylios 2 : 31 32 33 42 51 50 49 40
daktylios 3 : 21 22 23 24 25 34 43 52 61 60 59 58 57 48 39 30
daktylios 4 : 11 12 13 14 15 16 17 26 35 44 53 62 71 70 69 68 67 66 65 56 47 38 29 20
daktylios 5 : 1 2 3 4 5 6 7 8 9 18 27 36 45 54 63 72 81 80 79 78 77 76 75 74 73 64 55 46 37 28 19 10
```

```
Gemisma pinaka. Please wait...
Pinakas A( 5 x 5 ):
-----
```

```
1  2  3  4  5
6  7  8  9  10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
-----
```

```
sarwsh_kata_daktyliouy:
daktylios 1: 13
daktylios 2 : 7 8 9 14 19 18 17 12
daktylios 3 : 1 2 3 4 5 10 15 20 25 24 23 22 21 16 11 6
```


100. Ένας **φοιτητής** δίνει 4 μαθήματα σε ένα εξάμηνο. Περνάει το εξάμηνο εάν: (a) έχει σε όλα τουλάχιστον τη βάση (50) (b) έχει σε ένα κάτω από τη βάση και μέσο όρο άνω του 60 (c) έχει σε δύο κάτω από τη βάση αλλά πάνω από 40 και μέσο όρο άνω του 70. Να γραφεί πρόγραμμα που διαβάζει τους βαθμούς του φοιτητή (0-100) και ελέγχει εάν περνάει το εξάμηνο (foithths.psc). Σενάρια ελέγχου: a. 50, 50, 50, 50 (ok) b. 49, 50, 50, 100 (ok) c. 49, 49, 90, 100 (ok) d. 49, 49, 80, 100 / 40, 49, 100, 100 (failed)

102. Να γραφεί πρόγραμμα το οποίο διαβάζει τον πραγματικό A[9] και βρίσκει τον 1ο **υποπίνακα συνεχόμενων στοιχείων του με το μέγιστο άθροισμα**. π.χ. A={-8, 3, 8, -5, 4, 5, -4, 3, -5} → μέγιστο άθροισμα = 15 για τον υποπίνακα A[2]-A[6] (largest_sum_of_contiguous_subarray.psc)

107. Να γραφεί πρόγραμμα που διαβάζει τον ακέραιο A[10] τοποθετώντας κάθε στοιχείο με **παρεμβολή** στη σωστή θέση ώστε ο πίνακας να είναι ταξινομημένος κατά αύξουσα σειρά. π.χ. Αν ο A περιέχει τα στοιχεία 5,7,8 και εισαχθεί η τιμή 6, αυτή να παρεμβληθεί μεταξύ του 5 και του 7. (parembolh_stoixeiwn.psc)

108. Να γραφεί πρόγραμμα το οποίο υπολογίζει τις **συχνότητες των αθροισμάτων** (2-12) και των **γινομένων** (1-36) της ρίψης **2 ζαριών**. Ποιο το συχνότερο άθροισμα και γινόμενο; (syxnothtes_2_zariwn.psc)

109. Σε μία **φάση κυπέλλου ποδοσφαίρου συμμετέχουν 16 ομάδες με σκοπό να προκριθούν στους 8**. Να γραφεί πρόγραμμα που διαβάζει στον OM1[16] τα ονόματα των ομάδων. Έστω ότι τα ζευγάρια των αναμετρήσεων είναι ανά δύο (1-2, 3-4, ..., 15-16). Για κάθε μία από τις 8 αναμετρήσεις διαβάζει τον αριθμό του νικητή με έλεγχο εγκυρότητας. Γεμίζει τον OM2[8] με τις ομάδες που προκρίθηκαν και τον εμφανίζει. (eromenh_fash_kypelloy.psc)

```
syxnotero athroisma: 7  syxnothta: 6
syxnotero ginomeno: 6  syxnothta: 4
syxnotero ginomeno: 12 syxnothta: 4
```


110. Έστω **ανελκυστήρας** προγραμματισμένος να εξυπηρετεί τα **αιτήματα** με την εξής σειρά: αν κατευθύνεται προς τα πάνω και βρίσκεται στον όροφο x , εξυπηρετεί πρώτα τα αιτήματα από τους ορόφους που είναι μεγαλύτεροι του x και κατά αύξουσα σειρά και έπειτα τα αιτήματα από τους ορόφους που είναι μικρότεροι του x και κατά φθίνουσα σειρά. π.χ. αν ο ανελκυστήρας βρίσκεται στον 9ο όροφο και έχουν καταγραφεί τα αιτήματα από τους ορόφους: 11, 3, 20, 19, 1, 4, 7, 8, 12, 5 τότε αυτά θα εξυπηρετηθούν με την εξής σειρά: 11,12,19,20,8,7,5,4,3,1. Να γραφεί πρόγραμμα το οποίο διαβάζει τον όροφο x που βρίσκεται ένας ανελκυστήρας σε 20όροφο κτίριο καθώς και τα 10 αιτήματα που έστω ότι έχουν καταγραφεί (1-20 και διαφορετικά του x). Να εμφανίζει τη σειρά εξυπηρέτησης των αιτημάτων (anelkysthras.psc)

112. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο $A[N]$ (έστω με διαφορετικές τιμές) και μία τιμή αναζήτησης x . Να **αναζητά το x στον A και από τις δύο κατευθύνσεις ταυτόχρονα** (από την αρχή και από το τέλος του) και να εμφανίζει το αποτέλεσμα της αναζήτησης καθώς και τον αριθμό των επαναλήψεων που χρειάστηκε (search_both_directions.psc)

114. Να γραφεί πρόγραμμα που προσομοιώνει τη **μετάδοση ενός ιού** ως εξής: αρχικοποιεί έναν πίνακα χαρακτήρων $A[N,M]$ με την τιμή 'Y' (υγιές). Διαβάζει τις συντεταγμένες (x,y) ενός μολυσμένου στοιχείου και του εκχωρεί την τιμή 'M' (μολυσμένο). Εάν καθημερινά μολύνεται κάθε κελί που γειτονεύει με κάποιο ήδη μολυσμένο, να ενημερώνει τον πίνακα μέχρι να εξαπλωθεί ο ιός σε όλα τα στοιχεία του. Κάθε ημέρα να εμφανίζει το πλήθος των μολυσμένων στοιχείων. Τελικά να εμφανίζει το πλήθος των ημερών που απαιτήθηκαν για να συμβεί το γεγονός αυτό. (virus_spread.psc). π.χ. για $A[3,5]$ και αρχικό μολυσμένο στοιχείο το $A[1,1] \rightarrow 4$ ημέρες, για $A[3,5]$ και αρχικό μολυσμένο στοιχείο το $A[2,3] \rightarrow 2$ ημέρες

118. Να γραφεί πρόγραμμα το οποίο διαβάζει έναν ακέραιο $A[N]$ και υπολογίζει το **% ποσοστό ανακατέματός** του το οποίο προκύπτει από τον αριθμό των αντιμεταθέσεων που απαιτήθηκαν ώστε να ταξινομηθεί κατά αύξουσα σειρά δια του μέγιστου πλήθους των απαιτούμενων αντιμεταθέσεων για την ταξινόμησή του επί 100. Δηλ. $pososto_anakatematos = antimetatheseis_taksinomhshs / maxAntimetatheseis * 100$, $maxAntimetatheseis = N*(N-1)/2$ (pososto_anakatematatos.psc)

119. Να γραφεί πρόγραμμα το οποίο από τους N πρώτους ακέραιους εμφανίζει αυτούς με το **μεγαλύτερο πλήθος διαιρετών** (max_diairetes.psc). π.χ. για N=100 →

```
0 60 exei 12 diairetes
0 72 exei 12 diairetes
0 84 exei 12 diairetes
0 90 exei 12 diairetes
0 96 exei 12 diairetes
```

122. Να γραφεί πρόγραμμα το οποίο διαβάζει έναν ακέραιο πίνακα A[100, 100] τιμές 0/1. Να εμφανίζει τους 3x3 υποπίνακες του A που έχουν **σχήμα σταυρού** από άσσους: καθώς και πόσοι είναι (find_crosses.psc) π.χ.

0	1	0
1	1	1
0	1	0

```
0 1 0 1 0 1 0
1 1 1 1 1 1 1
0 1 0 1 0 1 0
1 1 1 1 1 1 1
0 1 0 1 0 1 0
1 1 1 1 1 1 1
0 1 0 1 0 1 0
cross at ( 2 , 2 )
cross at ( 2 , 4 )
cross at ( 2 , 6 )
cross at ( 4 , 2 )
cross at ( 4 , 4 )
cross at ( 4 , 6 )
cross at ( 6 , 2 )
cross at ( 6 , 4 )
cross at ( 6 , 6 )
Crosses found: 9
```

127. **Διανομέας καφέ** έχει να παραδώσει καφέδες σε 4 σημεία της πόλης. Χρειάζεται μία εφαρμογή η οποία να του συστήνει τη συντομότερη διαδρομή που ξεκινάει από το μαγαζί που εργάζεται, περνάει από τα 4 σημεία διανομής και επιστρέφει στη βάση του. Να γραφεί πρόγραμμα το οποίο: α) καταχωρεί στον D[5, 5] τις αποστάσεις μεταξύ των 5 σημείων (γραμμές/στήλες 1-4: τα 4 σημεία διανομής και γραμμή/στήλη 5: το σημείο εκκίνησης και τερματισμού). Ενδεικτικά: $D[i, j] = i * j$ για $i > j$ και μηδέν(0) στη διαγώνιο β) υπολογίζει το μήκος όλων των διαφορετικών διαδρομών ($4! = 1 \times 2 \times 3 \times 4 = 24$) γ) εμφανίζει το μικρότερο μήκος διαδρομής και τη(τις) διαδρομές που το έχουν.

Travelling cafe delivery problem

Distances

D	1	2	3	4	5
1	0	2	3	4	5
2	2	0	6	8	10
3	3	6	0	12	15
4	4	8	12	0	20
5	5	10	15	20	0

$D[i, j] = i \times j, i \neq j$

Paths (starting from and ending to 5)

1. 1 2 3 4 S=45
2. 1 2 4 3 S=42
3. 1 3 2 4 S=42
4. 1 3 4 2 S=38
5. 1 4 2 3 S=38
6. 1 4 3 2 S=37 (5+4+12+6+10)
7. 2 1 3 4 S=47
8. 2 1 4 3 S=43
9. 2 3 1 4 S=43
10. 2 3 4 1 S=37 (10+6+12+4+5)
11. 2 4 1 3 S=40
12. 2 4 3 1 S=38
13. 3 1 2 4 S=48
14. 3 1 4 2 S=40
15. 3 2 1 4 S=47
16. 3 2 4 1 S=38
17. 3 4 1 2 S=43
18. 3 4 2 1 S=42
19. 4 1 2 3 S=47
20. 4 1 3 2 S=43
21. 4 2 1 3 S=48
22. 4 2 3 1 S=42
23. 4 3 1 2 S=47
- (24=4!) 24. 4 3 2 1 S=45

min= 37

traveling_salesman4.psc

130. **Ταξινόμηση με Κάδους - BucketSort:** Έστω ότι ο πίνακας A η στοιχείων περιέχει στοιχεία που ανήκουν στο διάστημα [1..m]. Ο αλγόριθμος BucketSort βασίζεται πάνω στα ακόλουθα βήματα: 1. Δημιουργούμε ένα πίνακα buckets μήκους m και θέτουμε buckets[i]=0, για όλα τα i (Αυτά τα είναι τα buckets - κάδοι). 2. Διαβάζουμε τον πίνακα A ξεκινώντας από το πρώτο στοιχείο. Αν διαβάσουμε το στοιχείο α, τότε αυξάνουμε την τιμή του buckets[α] κατά ένα. Επαναλαμβάνουμε το βήμα μέχρι το τελευταίο στοιχείο. 3. Τέλος, διαβάζουμε γραμμικά τον πίνακα buckets, ο οποίος περιέχει αναπαράσταση του ταξινομημένου πίνακα, και θέτουμε τα στοιχεία του πίνακα A με την ταξινομημένη ακολουθία. Να γραφεί πρόγραμμα που διαβάζει έναν ακέραιο A[20] με τιμές στο [1, 6] και τον ταξινομεί με τον παραπάνω αλγόριθμο (BucketSort.psc)

PS[1]	4
PS[2]	4
PS[3]	3
PS[4]	3
PS[5]	3
PS[6]	3

```
A: 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6 1 2
A sorted: 1 1 1 1 2 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6
```

π.χ.

135. Να γραφεί πρόγραμμα το οποίο διαβάζει τον ακέραιο A[10] με θετικές τιμές και τον ταξινομεί κατά φθίνουσα σειρά με βάση το άθροισμα των ψηφίων τους. Οι τιμές με ίσο άθροισμα ψηφίων να ταξινομούνται κατά φθίνουσα σειρά (sort_by_sum_of_digits.psc). Π.χ. για τιμές εισόδου: 12, 23, 34, 45, 56, 67, 78, 89, 100, 121:

```
Arithmos 89 athroisma pshfiwn 17
Arithmos 78 athroisma pshfiwn 15
Arithmos 67 athroisma pshfiwn 13
Arithmos 56 athroisma pshfiwn 11
Arithmos 45 athroisma pshfiwn 9
Arithmos 34 athroisma pshfiwn 7
Arithmos 23 athroisma pshfiwn 5
Arithmos 121 athroisma pshfiwn 4
Arithmos 12 athroisma pshfiwn 3
Arithmos 100 athroisma pshfiwn 1
```

Υποπρογράμματα

- Να γραφεί πρόγραμμα το οποίο υπολογίζει το εμβαδό του κύκλου από την ακτίνα του. (γρ_paradeigma_biblioug.psc)
- Συνάρτηση υπολογισμού του μεγίστου τριών τιμών. Συνάρτηση υπολογισμού του ελαχίστου τριών τιμών. Πρόγραμμα που διαβάζει τρεις τιμές και με χρήση των συναρτήσεων εμφανίζει τον μέγιστο και τον ελάχιστο (γρ1.psc)
- Διαδικασία που δέχεται τρεις τιμές και επιστρέφει τον ελάχιστο, τον μέγιστο και τον μέσο όρο τους. Πρόγραμμα που διαβάζει τρεις τιμές και με χρήση της διαδικασίας εμφανίζει τον μέγιστο και τον μέσο όρο τους (γρ2.psc)
- α) Να γραφεί συνάρτηση η οποία δέχεται σαν παραμέτρους 2 πραγματικούς και επιστρέφει τη μεγαλύτερη τιμή. β) Να γραφεί πρόγραμμα που διαβάζει 4 πραγματικούς και με χρήση της παραπάνω συνάρτησης υπολογίζει και εμφανίζει τη μεγαλύτερη και τη μικρότερη τιμή. (γρ3.psc)
- Διαδικασία που δέχεται σαν είσοδο έναν ακέραιο A[10] και μία λογική μεταβλητή *agksoysa* και επιστρέφει τον πίνακα ταξινομημένο κατά αύξουσα σειρά εάν η *agksoysa* είναι Αληθής, διαφορετικά τον επιστρέφει ταξινομημένο κατά φθίνουσα σειρά. Διαδικασία που δέχεται σαν είσοδο έναν ακέραιο A[10] και τον εμφανίζει. Πρόγραμμα που διαβάζει έναν ακέραιο A[10] και με τη βοήθεια των διαδικασιών να τον εμφανίζει κατά αύξουσα και κατά φθίνουσα σειρά (γρ4.psc)
- Πρόγραμμα που διαβάζει τους όρους ενός κλάσματος, υπολογίζει το μέγιστο κοινό τους διαιρέτη και απλοποιεί το κλάσμα (AploroihshKlasmatos.psc)

12. Εισαγωγή 2 θετικών ακέραιων και έλεγχος αν είναι φίλιοι: το άθροισμα των γνήσιων διαιρετών του καθενός (δηλ. εκτός του εαυτού τους) ισούται με τον άλλο. (filioi_arithmoi.psc)



```
Enter 1st number
1184
Enter 2nd number
1210
ΕΙΝΑΙ!
```

45. Ο αριθμός 202 ονομάζεται super παλινδρομικός επειδή και αυτός και το τετράγωνό του $202^2 = 40804$ είναι παλινδρομικοί αριθμοί (δηλαδή διαβάζονται το ίδιο είτε από αριστερά είτε από δεξιά). Να γραφεί πρόγραμμα που θα εμφανίζει όλους τους super παλινδρομικούς αριθμούς από το 1 μέχρι το 1111 (super_palindromikoι.psc). Υπόδειξη: φτιάξτε τη Συνάρτηση palindromikos(x): Λογική

```
1 ( 1 )
2 ( 4 )
3 ( 9 )
11 ( 121 )
22 ( 484 )
101 ( 10201 )
202 ( 40804 )
212 ( 44944 )
111 ( 12321 )
121 ( 14641 )
1001 ( 1002001 )
1111 ( 1234321 )
```


52. Να γραφεί πρόγραμμα (*diagwnismos_tragoydiou.psc*) που προσομοιώνει έναν **διαγωνισμό τραγουδιού** στον οποίο συμμετέχουν 5 χώρες. Το κύριο πρόγραμμα να καλεί τις παρακάτω διαδικασίες:

- **Arxikopoihsh**(Χwres, Bathmologies) η οποία αρχικοποιεί τον πίνακα Χwres με τα ονόματα 5 χωρών και τον παράλληλο πίνακα Bathmologies με το μηδέν(0)
- **Bathmologhsh**(Χwres, Bathmologies) η οποία ζητάει από κάθε χώρα τον αριθμό των χωρών (1-5) στις οποίες δίνει 3 βαθμούς, το 10 το 11 και το 12. Η εισαγωγή να γίνεται με έλεγχο εγκυρότητας, ώστε:
 - ο αριθμός της προς βαθμολόγηση χώρας να είναι 1-5
 - να μην επιτρέπεται η αυτοβαθμολόγηση (βαθμολόγηση της χώρας που βαθμολογεί)
 - να μην επιτρέπεται η επαναβαθμολόγηση (πολλαπλή βαθμολόγηση της ίδιας χώρας)

```
Bathmologies ths xwras Greece
1 . Greece : 0
2 . England : 0
3 . France : 0
4 . Italy : 0
5 . Spain : 0
Poy dineis th bathmologia 10 (1- 5 )?
```

Να ενημερώνει τον πίνακα Bathmologies με τις συνολικές βαθμολογίες των χωρών.

- **Taksinomhsh**(Χwres, Bathmologies) η οποία ταξινομεί κατά φθίνουσα σειρά τον πίνακα Bathmologies και παράλληλα τον πίνακα Χwres
- **Emfanish**(Χwres, Bathmologies) η οποία εμφανίζει την τελική κατάταξη των χωρών

```
Telikh katataksh
1 . France : 46
2 . England : 43
3 . Greece : 41
4 . Italy : 24
5 . Spain : 11
```

66. Στη θεωρία των αριθμών, ένας **ναρκισσιστικός αριθμός** (narcissistic number) είναι ένας ακέραιος αριθμός που ισούται με το άθροισμα των ψηφίων του υψωμένα στον αριθμό των ψηφίων του. π.χ. $153 = 1^3 + 5^3 + 3^3$, $1634 = 1^4 + 6^4 + 3^4 + 4^4$, $54748 = 5^5 + 4^5 + 7^5 + 4^5 + 8^5$. Οι ναρκισσιστικοί αριθμοί είναι: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, ... Να γράψετε πρόγραμμα που εμφανίζει όλους τους ναρκισσιστικούς αριθμούς στο διάστημα 1-N, όπου N στοιχείο εισόδου (*Narcissistic_number.psc*).

$$153 = 1^3 + 5^3 + 3^3$$

$$370 = 3^3 + 7^3 + 0^3$$

$$371 = 3^3 + 7^3 + 1^3$$

$$407 = 4^3 + 0^3 + 7^3$$

```

1 Narcissistic
2 Narcissistic
3 Narcissistic
4 Narcissistic
5 Narcissistic
6 Narcissistic
7 Narcissistic
8 Narcissistic
9 Narcissistic
153 Narcissistic
370 Narcissistic
371 Narcissistic
407 Narcissistic

```

89. Ένας αριθμός έχει ένα **σημείο διακοπής (breakpoint)** εάν μπορεί να διαχωρισθεί με τέτοιο τρόπο ώστε τα ψηφία στην αριστερή πλευρά και τα ψηφία στη δεξιά πλευρά να έχουν το ίδιο άθροισμα. Για παράδειγμα, ο αριθμός 35291 έχει ένα breakpoint διότι μπορεί να διαχωρισθεί μεταξύ των ψηφίων 352 και 91, και $3 + 5 + 2 = 10$ και $9 + 1 = 10$. Ο αριθμός 1234 δεν έχει breakpoint. Να γραφεί πρόγραμμα το οποίο διαβάσει έναν θετικό ακέραιο και εμφανίζει όλα τα σημεία διακοπής που έχει ή κατάλληλο μήνυμα εάν δεν έχει κανένα. (*breakpoint.psc*)

```

35291
Checking: 3 5291
Checking: 35 291
Checking: 352 91
Breakpoint with sum of digits: 10
Checking: 3529 1

```

```

1234
Checking: 1 234
Checking: 12 34
Checking: 123 4
No breakpoint found

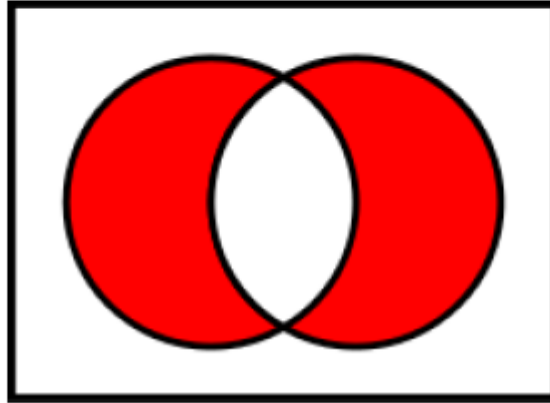
```

```

1203
Checking: 1 203
Checking: 12 3
Breakpoint with sum of digits: 3
Checking: 120 3
Breakpoint with sum of digits: 3

```

91. Η **συμμετρική διαφορά** δύο συνόλων A, B ορίζεται ως τα στοιχεία του A που δεν ανήκουν στο B και τα στοιχεία του B που δεν ανήκουν στο A. π.χ. $A = \{1,2,3,4,5\}$ $B = \{4,5,6,7,8\} \Rightarrow$ συμμετρική διαφορά = $\{1,2,3,6,7,8\}$. Να γραφεί πρόγραμμα που διαβάζει τους πίνακες A[5] και B[5] και εμφανίζει τη συμμετρική διαφορά τους καθώς και το πλήθος των στοιχείων της. Υπόδειξη: φτιάξτε κατάλληλη διαδικασία που δέχεται ως παραμέτρους 2 πίνακες και εμφανίζει τα στοιχεία του 1ου που δεν ανήκουν στον 2ο και επιστρέφει το πλήθος τους. Καλέστε την 2 φορές. (*symmetrikh_diafora.psc*)



92. Έστω ότι στον πίνακα A[N,N] καταγράφονται με 1 τα υγιή άτομα και με 3 τα άτομα που νοσούν από έναν μεταδοτικό ιό, σε έναν πληθυσμό NxN ατόμων. Κάθε ημέρα ο ιός μεταδίδεται από κάθε ασθενές άτομο προς όλα τα γειτονικά με αυτό άτομα. Να γραφεί πρόγραμμα που αρχικοποιεί έναν A[N, N] με την τιμή 1 (όλοι αρχικά υγιείς) και διαβάζει ένα στοιχείο (x,y) που θα αποτελέσει το 1ο ασθενές άτομο του πληθυσμού θέτοντας στο A[x,y] την τιμή 3. Να προσομοιώνει τη διαδικασία μετάδοσης του ιού εμφανίζοντας κάθε ημέρα πόσοι νοσούν, μέχρι να νοσήσει όλος ο πληθυσμός. Εμφανίζει πόσες ημέρες χρειάστηκαν για να συμβεί αυτό (covid19.psc). Π.χ. για N=11 και x=6, y=6:

Day: 0	Day: 1	Day: 2	Day: 3	Day: 4	Day: 5
1111111111	1111111111	1111111111	1111111111	1111111111	3333333333
1111111111	1111111111	1111111111	1111111111	1333333331	3333333333
1111111111	1111111111	1111111111	1133333311	1333333331	3333333333
1111111111	1111111111	1113333311	1133333311	1333333331	3333333333
1111131111	1111333111	1113333311	1133333311	1333333331	3333333333
1111111111	1111333111	1113333311	1133333311	1333333331	3333333333
1111111111	1111111111	1113333311	1133333311	1333333331	3333333333
1111111111	1111111111	1111111111	1133333311	1333333331	3333333333
1111111111	1111111111	1111111111	1111111111	1333333331	3333333333
1111111111	1111111111	1111111111	1111111111	1111111111	3333333333

Πρόγραμμα που διαβάζει έναν θετικό άρτιο (>2) και επαληθεύει την εικασία του [Goldbach](#): Κάθε άρτιος θετικός ακέραιος μεγαλύτερος του 2 μπορεί να γραφεί ως άθροισμα δύο πρώτων αριθμών, έτσι ώστε για κάθε $n > 2$, $n=p + q$, όπου p, q πρώτοι αριθμοί. Π.χ.

```
Όσο θετικό άρτιο μεγαλύτερο του 2
10
Άθροισμα πρώτων αριθμών: 3 + 7
Άθροισμα πρώτων αριθμών: 5 + 5
```

```
Όσο θετικό άρτιο μεγαλύτερο του 2
100
Άθροισμα πρώτων αριθμών: 3 + 97
Άθροισμα πρώτων αριθμών: 11 + 89
Άθροισμα πρώτων αριθμών: 17 + 83
Άθροισμα πρώτων αριθμών: 29 + 71
Άθροισμα πρώτων αριθμών: 41 + 59
Άθροισμα πρώτων αριθμών: 47 + 53
```

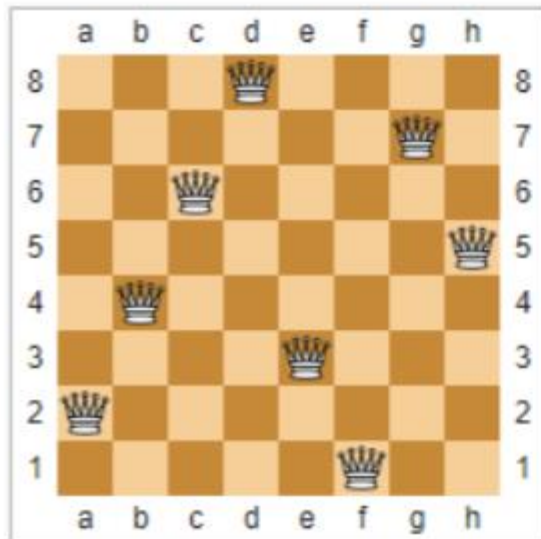
(goldbach.psc)

Πρόγραμμα που αρχικά ζητάει από τον χρήστη να τοποθετήσει τους ακέραιους αριθμούς 1-9 σε τετραγωνικό πίνακα $S[3,3]$. Στη συνέχεια να ελέγχει εάν σχηματίστηκε ένα μαγικό 3×3 τετράγωνο: όλες οι γραμμές, στήλες και διαγώνιοι έχουν το ίδιο άθροισμα. Π.χ.

```
S   1   2   3
-----
1 | 8 | 1 | 6 |
-----
2 | 3 | 5 | 7 |
-----
3 | 4 | 9 | 2 |
-----
Άθροίσματα γραμμών, στηλών και διαγωνίων:
15 15 15 15 15 15 15
ΜΑΓΙΚΟ ΤΕΤΡΑΓΩΝΟ!!!!
```

(magic_square.psc)

Ο γρίφος των 8 βασιλισσών υπάρχει από τη δεκαετία του 1850 και μας προκαλεί να τοποθετήσουμε 8 βασίλισσες σε μια σκακιέρα έτσι ώστε καμία από αυτές να μην μπορεί να επιτεθεί στην άλλη. Πρόγραμμα που ζητάει από τον χρήστη να τοποθετήσει 8 βασίλισσες σε μια σκακιέρα (πίνακας 8x8) και να ελέγχει τελικά αν οι βασίλισσες είναι τοποθετημένες έτσι ώστε να λύνουν το πρόβλημα. Π.χ.



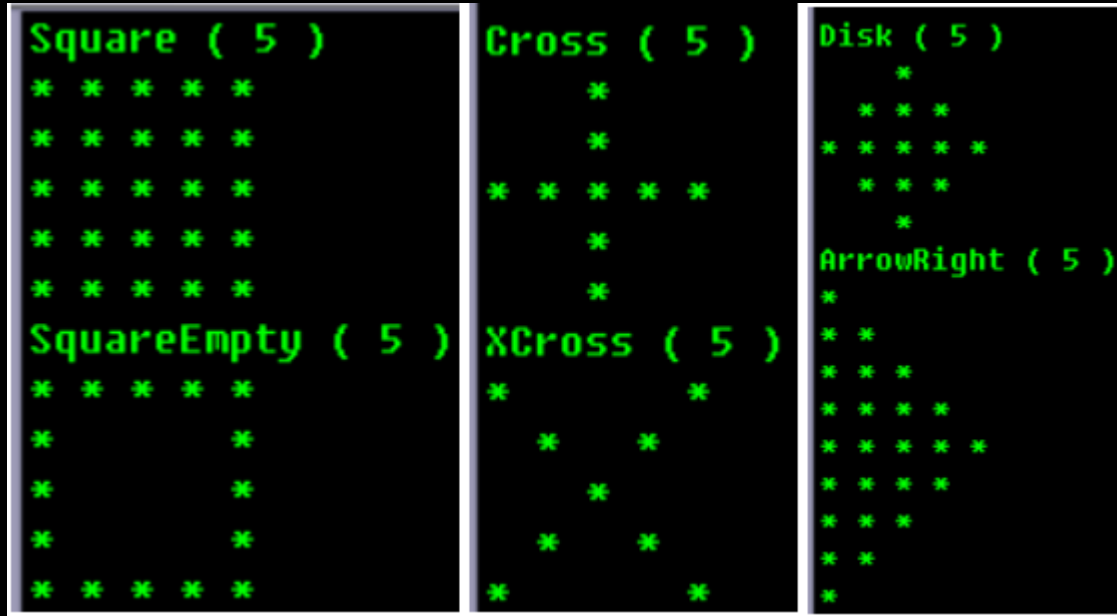
```

C   1   2   3   4   5   6   7   8
-----
1  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
-----
2  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
-----
3  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
-----
4  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
-----
5  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
6  | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
-----
7  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
-----
8  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
-----
Βασίλισσα 1  ok
Βασίλισσα 2  ok
Βασίλισσα 3  ok
Βασίλισσα 4  ok
Βασίλισσα 5  ok
Βασίλισσα 6  ok
Βασίλισσα 7  ok
Βασίλισσα 8  ok
Σωστό!

```

(eight_queens.psc)

98. Να γραφεί πρόγραμμα που σχεδιάζει τα παρακάτω με κλήση ξεχωριστών διαδικασιών για το κάθε σχήμα. Χρησιμοποιείτε την εντολή "ΓΡΑΨΕ_ ..." η οποία κάνει εμφάνιση χωρίς να αλλάξει γραμμή και την εντολή "ΓΡΑΨΕ" η οποία απλά αλλάζει γραμμή (draw_stars.psc)



101. "Δυνατός" είναι ένας θετικός ακέραιος με άθροισμα των παραγοντικών των ψηφίων του ίσο με τον αριθμό. π.χ. ο $145 = 1! + 4! + 5!$. Να γραφεί πρόγραμμα που διαβάζει έναν θετικό ακέραιο και ελέγχει αν είναι "δυνατός" (dynatos_arithmos.psc). Υπόδειξη: να γράψετε συνάρτηση που δέχεται έναν θετικό ακέραιο x και υπολογίζει το παραγοντικό του ($x! = 1*2*...*x$)

125. Η διαδικασία της σταθεράς του Karrekar (σταματά όταν προκύψει ο αριθμός 6174):

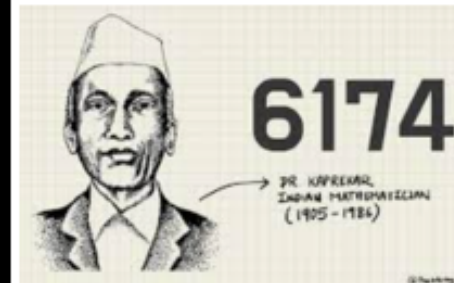
1. Παίρνουμε έναν ακέραιο K
2. Αναδιατάσσουμε τα ψηφία του K ώστε να προκύψει ο μέγιστος (και ο ελάχιστος) δυνατός αριθμός. Π.χ. αν ο K είναι ο 2309 τότε ο μέγιστος είναι ο 9320 και ο ελάχιστος ο 239 (θεωρείστε leading zero άρα $239 = 0239$)
3. Αφαιρούμε τον ελάχιστο από τον μέγιστο και με τη διαφορά τους εκτελούμε το βήμα 1

Παράδειγμα 1: Έστω ο αριθμός 2600. Ο μέγιστος ακέραιος που μπορεί να προκύψει από αναδιάταξη των ψηφίων του 2600 είναι ο 6200 και ο ελάχιστος είναι ο 26 (0026). Αφαιρώντας τους δύο ακέραιους προκύπτει η σταθερά Karrekar μετά από 1 επανάληψη ($6200 - 0026 = 6174$).

Παράδειγμα 2: Έστω ο αριθμός 1542. Ο μέγιστος ακέραιος που μπορεί να προκύψει από αναδιάταξη των ψηφίων του 1542 είναι ο 5421 και ο ελάχιστος είναι ο 1245. Αφαιρώντας τους δύο ακέραιους προκύπτει ο αριθμός 4176. Ο μέγιστος ακέραιος που μπορεί να προκύψει από αναδιάταξη των ψηφίων του 4176 είναι ο 7641 και ο ελάχιστος είναι ο 1467. Αφαιρώντας τους δύο ακέραιους προκύπτει η σταθερά Karrekar μετά από 2 επαναλήψεις ($7641 - 1467 = 6174$). (Karrekar_Constant.psc)

Είσοδος	Εξοδος
0	Wrong Input
3333	Wrong Input
2600	1
1542	2
3215	7
9899	5

```
Dwise K
9899
999
8991
8082
8532
6174
steps = 5
```



134. Να γραφεί πρόγραμμα το οποίο διαβάζει έναν θετικό ακέραιο N και στη συνέχεια **διαβάζει N θετικούς ακέραιους και τους συνενώνει σε έναν**. (concatNumbers.psc) π.χ. για $N=3$ και τιμές εισόδου: 32, 4, 128 => αποτέλεσμα = 324128. Υπόδειξη: να φτιάξετε συνάρτηση $Digits(x)$: Ακέραια η οποία επιστρέφει το πλήθος των ψηφίων ενός ακέραιου x