

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΑΘΗΜΑ : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΥ
ΕΞΑΜΗΝΟ : Α'
ΦΥΛΛΟ ΑΣΚΗΣΕΩΝ : 8 Συναρτήσεις (Functions), Μονάδες (Units)
ΥΠΕΥΘΥΝΟΣ ΜΑΘΗΜΑΤΟΣ : Μάγια Σατρατζέμη

Παρατηρήσεις:

1. Τα δεδομένα εισόδου διαβάζονται πάντα με ξεχωριστές εντολές `Readln()` το καθένα και με τη σειρά που δηλώνονται στις εκφωνήσεις.
 2. Αντίστοιχα για τα δεδομένα εξόδου και όπου δεν υπάρχουν περαιτέρω διευκρινήσεις για τη μορφή τους, αυτά θα εμφανίζονται με ξεχωριστές εντολές `Writeln()` το καθένα και με τη σειρά που δηλώνονται στις εκφωνήσεις.
 3. Το αποτέλεσμα πράξης στην οποία συμμετέχει αριθμός τύπου `Real`, θα είναι πάντα ακέραιος αριθμός. Σε αντίθετη περίπτωση θα γίνονται οι απαραίτητες διευκρινήσεις.
-

1. Δίνεται ο παρακάτω σκελετός ενός προγράμματος Pascal:

```
program demo;
const
    pi=3.14159;
    two=2;

var
    Month, Day, Year, p, q: Integer;
    Hours, Rate, Amount, u, v: real;
    Code, Class: char;

function f(x, y: real; d: integer): real;
begin
    .....
    .....
end;

procedure Calculate (a: real; var B: real; m: integer; var k,n: integer; var c: integer);
begin
    .....
    .....
end;
```

Προσδιορίστε ποιες από τις επόμενες εντολές μπορούν να χρησιμοποιηθούν μέσα σε αυτό το πρόγραμμα (δηλαδή δεν περιέχουν συντακτικά λάθη):

- a) `Amount:= f(pi, Rate, month);`
- b) `Rate:= f(Hours, Day, two);`
- c) `writeln(f(0, 0, 0));`
- d) `f(Hours, Rate, Month);`
- e) `Calculate(u, v, p, two, Day, code);`
- f) `Hours:= two*f(pi, amount) / (2.71828*Rate);`
- g) `Amount:= f(pi*Hours, (2.71828+Day) / Rate, two);`
- h) `if Month= two then`
 `Year:= f(Hours, f(Rate, pi, two), Day);`
- i) `if u > 0 then`
 `Amount:= Calculate(u, v, two, p, q, Code);`

```

j) if calculate(0, u, 1, p, year, class) > 0 then writeln('Okey');
k) while f(Amount, 0, 0) < pi do
    Amount:= f(Amount, pi, 1);
l) repeat
    Amount:= f(Amount, 0, code)
until Amount > 0;
m) Calculate(f(u, v, day), rate, 7, p, q, Code);
n) Amount := f(a, b, day);

```

2. Να γραφεί ένα πρόγραμμα που θα διαβάζει έναν αριθμό τύπου Real (μεταβλητή number) και έναν αριθμό τύπου Byte (μεταβλητή places) και θα περιλαμβάνει μια συνάρτηση με όνομα Round_Dec(number, places), η οποία θα επιστρέφει την τιμή της μεταβλητής number στρογγυλοποιημένη στον καθορισμένο αριθμό δεκαδικών ψηφίων places (π.χ. Round_Dec(12.547,0), Round_Dec(12.547,1), Round_Dec(12.547,2) θα επιστρέφουν 13, 12.5 και 12.55 αντίστοιχα). Η εμφάνιση της στρογγυλοποιημένης τιμής της μεταβλητής number θα γίνεται από το κυρίως πρόγραμμα και θα είναι στοιχημένη στα αριστερά χωρίς κενά.
3. Να γραφεί μια συνάρτηση με όνομα YesNo που θα διαβάζει την απάντηση του χρήστη στο (N/O) και θα επιστρέφει τη boolean τιμή true ή false αν πατήσει το πλήκτρο N ή O αντίστοιχα. Η συνάρτηση να δουλεύει για όλες τις καταστάσεις των πλήκτρων N και O (πεζοί-κεφαλαίοι, αγγλικοί-ελληνικοί χαρακτήρες).
4. Να γραφεί πρόγραμμα που θα διαβάζει τις τιμές τριών μεταβλητών Kefalaio (τύπου Longint), Epitokio (τύπου Real, π.χ. για επιτόκιο 12% \Rightarrow 0.12) και Eti (τύπου Integer) και στη συνέχεια θα υπολογίζει και θα εμφανίζει το συνολικό κεφάλαιο. Η εισαγωγή των στοιχείων θα γίνεται με τη βοήθεια διαδικασίας ενώ ο υπολογισμός του κεφαλαίου με τη βοήθεια συνάρτησης.
5. Να γραφεί πρόγραμμα που θα αποτελείται από τις παρακάτω συναρτήσεις οι οποίες θα δέχονται ένα χαρακτήρα και θα ελέγχουν αν είναι:

- αραβικός αριθμός ('ARABIC'),
- γράμμα του αγγλικού αλφαβήτου ('ENGLISH'),
- αγγλικό κεφαλαίο γράμμα ('ENGLISH CAPITAL'),
- αγγλικό πεζό (μικρό) γράμμα ('ENGLISH SMALL'),
- αγγλικό φωνήεν (πεζό ή κεφαλαίο) ('ENGLISH VOWEL'),
- γράμμα του ελληνικού αλφαβήτου ('GREEK'),
- ελληνικό κεφαλαίο γράμμα ('GREEK CAPITAL'),
- ελληνικό πεζό (μικρό) γράμμα ('GREEK SMALL'),
- ελληνικό φωνήεν (πεζό ή κεφαλαίο) ('GREEK VOWEL'),

Θα επιστρέφουν μια τιμή Boolean ανάλογα με το χαρακτήρα που δέχονται. Το πρόγραμμα θα καλεί διαδοχικά όλες τις παραπάνω συναρτήσεις και θα εμφανίζει ανάλογα μηνύματα για τις συναρτήσεις που αληθεύουν. (π.χ. για το χαρ. 'α' θα εμφανιστούν σε διαφορετικές γραμμές τα μηνύματα 'GREEK', 'GREEK SMALL', 'GREEK VOWEL')

6. Να γραφεί πρόγραμμα που θα περιέχει τις παρακάτω συναρτήσεις:
 - Μια συνάρτηση με όνομα Maxdate που θα δέχεται δύο ημερομηνίες και θα βρίσκει ποια από τις δύο είναι χρονικά μεγαλύτερη. Θα επιστρέφεται ο αύξων αριθμός της μεγαλύτερης ημερομηνίας. Ο αύξων αριθμός προκύπτει από τη σειρά με την οποία έγινε η εισαγωγή των ημερομηνιών.
 - Μια συνάρτηση με όνομα Difference που θα δέχεται δύο ημερομηνίες και θα επιστρέφει τη διαφορά των ημερών που χωρίζουν τις δύο αυτές ημερομηνίες (βλέπε βιβ. Κ. Κ. Τσούρου σελ. 95).
 Κάθε ημερομηνία θα αποτελείται από τρεις ακέραιους αριθμούς που θα δηλώνουν αντίστοιχα ημέρα, μήνα και έτος. Το έτος θα δίνεται ολόκληρο, δηλαδή 1999 και όχι 99. Στη συνέχεια, οι ημερομηνία που δίνονται θα ελέγχονται αν είναι ημερολογιακά σωστές με τη χρήση της διαδικασίας Valid_Date και αν δεν είναι τότε θα εμφανίζεται το μήνυμα λάθους 'ERROR' και το πρόγραμμα θα τερματίζει. Αν και οι δύο ημερομηνίες είναι έγκυρες τότε θα εμφανίζονται οι τιμές των συναρτήσεων Maxdate και Difference.

7. Να γραφεί μια συνάρτηση με όνομα AbsDate που θα δέχεται μια ημερομηνία και θα επιστρέφει τον απόλυτο αριθμό της (1-365 ή 366) μέσα στο έτος. Η ημερομηνία θα διαβάζεται από το κυρίως πρόγραμμα και θα αποτελείται από τρεις ακέραιους αριθμούς που θα δηλώνουν αντίστοιχα ημέρα, μήνα και έτος. Το έτος θα δίνεται ολόκληρο, δηλαδή 1999 και όχι 99. Να χρησιμοποιηθεί η διαδικασία Valid_Date για τον έλεγχο της

ημερομηνίας. Αν η ημερομηνία δεν είναι έγκυρη τότε το κυρίως πρόγραμμα θα εμφανίζει το μήνυμα λάθους 'ERROR', αλλιώς θα εμφανίζει την τιμή της συνάρτησης.

8. Να γραφεί ένα πρόγραμμα που θα διαβάζει έναν ακέραιο αριθμό τύπου Longint και θα περιλαμβάνει μια συνάρτηση με όνομα Reverse, η οποία θα δέχεται τον ακέραιο αυτό και θα επιστρέφει τον αριθμό που προκύπτει από την αντιστροφή των ψηφίων του (π.χ. 271 ⇒ 172, 256400 ⇒ 4652). Η τιμή της συνάρτησης θα εμφανίζεται από το κυρίως πρόγραμμα.

9. Οι αριθμοί για τους οποίους το άθροισμα των κύβων των ψηφίων τους είναι ίσο με τον ίδιο τον αριθμό λέγονται αριθμοί Armstrong. (π.χ. $153 = 1^3 + 5^3 + 3^3$) Να γραφεί πρόγραμμα που θα περιλαμβάνει τα παρακάτω :

- την συνάρτηση Cube που θα επιστρέφει την τρίτη δύναμη ενός μονοψηφίου ακέραιου αριθμού,
- την διαδικασία Separate που θα δέχεται έναν ακέραιο αριθμό και θα επιστρέφει τα ψηφία από τα οποία αποτελείται και
- την συνάρτηση Sum_Cube που θα δέχεται έναν ακέραιο αριθμό και θα επιστρέφει το άθροισμα των κύβων των ψηφίων του.

Το κυρίως πρόγραμμα θα βρίσκει και θα εμφανίζει ποιοι είναι αριθμοί Armstrong στο διάστημα (1-999).

10. Να γραφεί πρόγραμμα που θα περιλαμβάνει τα παρακάτω :

- μια συνάρτηση με όνομα Parag που θα δέχεται έναν ακέραιο αριθμό μικρότερο του 12 και θα επιστρέφει το παραγοντικό του (longint) και
- μια συνάρτηση με όνομα Force που θα βρίσκει τη n-οστή δύναμη ($n = \text{ακέραιος}$) ενός πραγματικού αριθμού x . Θα επιστρέφει real τύπου αριθμό.

Το πρόγραμμα θα βρίσκει την τιμή του $\sin x$ από τον τύπο :

$$\sin x = x - x^3/3! + x^5/5! - x^7/7! + \dots$$

Ο υπολογισμός θα σταματά όταν υπολογιστεί ο όρος $x^{11}/11!$ ή όταν η ακρίβεια του αποτελέσματος γίνει μικρότερη του 0.0001. Στη συνέχεια το κυρίως πρόγραμμα θα εμφανίζει το αποτέλεσμα σε δεξιά στοίχιση των 12 χαρακτήρων με ακρίβεια 10 δεκαδικών ψηφίων, καθώς και τον εκθέτη του τελευταίου όρου που υπολογίστηκε (3, 5, 7 ή 11). Το x (real) θα δίνεται σε ακτίνια (rads) και όχι σε μοίρες.

11. Να γραφεί πρόγραμμα που θα διαβάζει δύο ακέραιους αριθμούς τύπου Integer (μεταβλητές a και m) και θα περιλαμβάνει μια συνάρτηση με όνομα Int_to_String(a, m) που θα μετατρέπει τον θετικό ακέραιο a σε αλφαριθμητικό μήκους m τοποθετώντας μηδενικά στις τυχόν κενές θέσεις που μένουν μπροστά από τον αριθμό (π.χ. Int_to_String(4437,7) θα δώσει αποτέλεσμα '0004437'). Η τιμή της συνάρτησης να εμφανίζεται από το κυρίως πρόγραμμα.

12. Να γραφούν 3 διαφορετικά προγράμματα καθένα από τα οποία θα περιλαμβάνει και μια από τις παρακάτω συναρτήσεις:

- a) String_to_Integer ⇒ μετατρέπει ένα αλφαριθμητικό σε αριθμό τύπου Integer,
- b) String_to_Longint ⇒ μετατρέπει ένα αλφαριθμητικό σε αριθμό τύπου Longint και
- c) String_to_Real ⇒ μετατρέπει ένα αλφαριθμητικό σε αριθμό τύπου Real.

Θα ελέγχεται η ύπαρξη μη αριθμητικών χαρακτήρων στο αλφαριθμητικό οπότε θα επιστρέφεται τιμή 0, ενώ στην περίπτωση πραγματικού αριθμού θα διαβάζεται και ο αριθμός των δεκαδικών ψηφίων που θα έχει με στρογγυλοποίηση του τελευταίου ψηφίου. Το κυρίως πρόγραμμα θα εμφανίζει την τιμή που επιστρέφει η συνάρτηση. Οι δεκαδικοί αριθμοί θα εμφανίζονται στοιχημένοι στα αριστερά χωρίς κενά.

13. Να γραφεί ένα πρόγραμμα που θα διαβάζει έναν αριθμό τύπου Integer και θα περιλαμβάνει μια συνάρτηση με όνομα Binstring, η οποία θα μετατρέπει τον αριθμό αυτό σε δυαδικό αλφαριθμητικό (π.χ. $44 \Rightarrow '101100'$). Η τιμή της συνάρτησης θα εμφανίζεται από το κυρίως πρόγραμμα.

14. Να γραφεί ένα πρόγραμμα που θα διαβάζει έναν δυαδικό αριθμό υπό μορφή αλφαριθμητικού και θα περιλαμβάνει μια συνάρτηση με όνομα From_Bin_to_Integer, η οποία θα μετατρέπει το αλφαριθμητικό αυτό σε αριθμό τύπου Integer (π.χ. '101100' ⇒ 44). Η τιμή που επιστρέφει η συνάρτηση θα εμφανίζεται από το κυρίως πρόγραμμα.

15. Να γραφεί ένα πρόγραμμα που θα διαβάζει έναν αριθμό τύπου Integer και θα περιλαμβάνει μια συνάρτηση με όνομα Hexstring, η οποία θα μετατρέπει τον αριθμό αυτό σε δεκαεξαδικό αλφαριθμητικό (π.χ. 32767 ⇒ '7FFF'). Η τιμή της συνάρτησης θα εμφανίζεται από το κυρίως πρόγραμμα.
16. Να γραφεί ένα πρόγραμμα που θα διαβάζει ένα δεκαεξαδικό αριθμό υπό μορφή αλφαριθμητικού και θα περιλαμβάνει μια συνάρτηση με όνομα From_Hex_to_Integer, η οποία θα μετατρέπει το αλφαριθμητικό αυτό σε αριθμό τύπου Integer (π.χ. '7FFF' ⇒ 32767). Η τιμή που επιστρέφει η συνάρτηση θα εμφανίζεται από το κυρίως πρόγραμμα.
17. Να γραφεί ένα πρόγραμμα που θα διαβάζει ένα αλφαριθμητικό και θα περιλαμβάνει μια συνάρτηση με όνομα Capital, η οποία θα μετατρέπει τους πεζούς χαρακτήρες του αλφαριθμητικού αυτού σε κεφαλαίους. Η τιμή της συνάρτησης θα εμφανίζεται από το κυρίως πρόγραμμα. Για τους αγγλικούς χαρακτήρες να χρησιμοποιηθεί η συνάρτηση Urcase ενώ για τους ελληνικούς να βρεθεί η ακριβής αντιστοιχία (τονισμένοι, άτοννοι, με διαλυτικά) από τον κώδικα ASCII.
18. Να γραφεί συνάρτηση με επικεφαλίδα:
 - Συνάρτηση Password : Boolean;
 που θα επιτρέπει την εισαγωγή ενός συνθηματικού μήκους 7 χαρακτήρων σε συγκεκριμένη θέση της οθόνης, αόρατα, με δυνατότητα 3 το πολύ δοκιμών. Η εισαγωγή θα γίνει με χρήση της Readkey και οι 7 θέσεις θα δείχνουν συνεχώς το χαρακτήρα #176. Αν η εισαγωγή ολοκληρωθεί επιτυχώς θα εμφανίζεται το κατάλληλο μήνυμα ενώ σε περίπτωση αποτυχίας θα εμφανίζεται ανάλογο μήνυμα.
19. Να γραφεί ένα πρόγραμμα που θα διαβάζει τις τιμές 2 ακέραιων μεταβλητών x και y και θα περιλαμβάνει τις συναρτήσεις MAX(x,y:integer):integer και MIN(x,y:integer):integer, οι οποίες υπολογίζουν το μέγιστο και τον ελάχιστο όρο μεταξύ δυο ακεραίων. Οι τιμές των συναρτήσεων θα εμφανίζονται με την παραπάνω σειρά από το κυρίως πρόγραμμα.
20. Να γραφεί ένα πρόγραμμα που θα διαβάζει τις τιμές 2 ακέραιων μεταβλητών x, y και θα περιέχει τις παρακάτω συναρτήσεις:
 • ADD(x, y: integer): integer
 • SUB(x, y: integer): integer
 • MULT(x, y: integer): integer
 • DIVD(x, y: integer): integer
 οι οποίες θα υπολογίζουν αντίστοιχα το άθροισμα, τη διαφορά, το γινόμενο και το ακέραιο πηλίκο δύο ακέραιων αριθμών. Τέλος να υπολογίζονται οι τιμές των:
 • Mult(Add(x,y), Sub(x, Divd(x,y))) και
 • Divd(Sub(Mult(x, y), x), Add(x, y)).
 Το κυρίως πρόγραμμα θα εμφανίζει τις τιμές των παραπάνω συναρτήσεων με αντίστοιχη σειρά.
21. Να γίνει πρόγραμμα που θα διαβάζει τις τιμές 2 ακέραιων μεταβλητών και θα περιλαμβάνει τις συναρτήσεις:
 • MKD (x, y: integer): integer και
 • EKP (x, y: integer): integer,
 οι οποίες θα υπολογίζουν το μέγιστο κοινό διαιρέτη (MKΔ) και το ελάχιστο κοινό πολλαπλάσιο (EKP) δύο ακέραιων αριθμών. Οι συναρτήσεις να γραφούν με τη βοήθεια των ασκήσεων 19 και 20. Για τον υπολογισμό του MKΔ στηριχθείτε στις παρακάτω συνθήκες:
 * ο MKΔ των A και 0 είναι A
 * αν $A > B$ τότε ο $MKΔ(A, B) = MKΔ(A-B, B)$
 Στη συνέχεια το κυρίως πρόγραμμα θα εμφανίζει τις τιμές των παραπάνω συναρτήσεων με αντίστοιχη σειρά.
22. Με τη βοήθεια της άσκησης 21 να γράψετε μια συνάρτηση με όνομα SIMPLE η οποία θα δέχεται ως είσοδο ένα αλφαριθμητικό (String) της μορφής 'A/B' με A και B ακεραίους και θα επιστρέφει το κλάσμα απλοποιημένο, διαιρώντας τους δυο όρους του με το MKΔ τους (π.χ. 6/12 ⇒ 1/2). Το κυρίως πρόγραμμα θα διαβάζει το αλφαριθμητικό και στη συνέχεια θα εμφανίζει την τιμή της συνάρτησης SIMPLE.

23. Να γραφεί μια μονάδα (unit), η 'u23f2.pas' που θα περιλαμβάνει διαδικασίες ή συναρτήσεις για να προσδιορίσει τις παρακάτω πληροφορίες αν ως παράμετρος εισόδου θεωρηθεί ένας ακέραιος αριθμός τύπου Integer.

- Είναι ο ακέραιος αριθμός πολλαπλάσιο του 7, 11 ή του 13;
- Είναι το πλήθος των ψηφίων του περιττό ή άρτιο;
- Είναι ο αριθμός πρώτος;

Στη συνέχεια να γραφεί πρόγραμμα που θα ελέγχει τα υποπρογράμματα της μονάδας που υλοποιήσατε. Ο ακέραιος αριθμός θα διαβάζεται από το πρόγραμμα και θα εμφανίζονται με αντίστοιχη σειρά τα μηνύματα που ισχύουν για τον αριθμό αυτό:

‘ΠΟΛΛΑΠΛΑΣΙΟ ΤΟΥ 7’

‘ΠΟΛΛΑΠΛΑΣΙΟ ΤΟΥ 11’

‘ΠΟΛΛΑΠΛΑΣΙΟ ΤΟΥ 13’

‘ΑΡΤΙΟ ΠΛΗΘΟΣ ΨΗΦΙΩΝ’

‘ΠΕΡΙΤΤΟ ΠΛΗΘΟΣ ΨΗΦΙΩΝ’

‘ΠΡΩΤΟΣ ΑΡΙΘΜΟΣ’