

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΑΘΗΜΑ : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΥ
ΕΞΑΜΗΝΟ : Α'
ΦΥΛΛΟ ΑΣΚΗΣΕΩΝ : 7 Διαδικασίες (Procedures)
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. Τα αριθμητικά δεδομένα αναπαρίστανται πάντα από μεταβλητές ακέραιου τύπου (`Longint` κατά κύριο λόγο ή `Integer`). Σε αντίθετη περίπτωση (μεταβλητές τύπου `Real`) θα γίνονται οι απαραίτητες διευκρινήσεις.
 4. Το αποτέλεσμα πράξης στην οποία συμμετέχει αριθμός τύπου `Real`, θα είναι πάντα ακέραιος αριθμός. Σε αντίθετη περίπτωση θα γίνονται οι απαραίτητες διευκρινήσεις.
 5. Για την εμφάνιση των τιμών μεταβλητών τύπου `Real` θα χρησιμοποιείται πάντα η εξής στοίχιση: μεταβλητή:3:1, δηλαδή τα δεδομένα θα είναι στοιχημένα στα αριστερά χωρίς κενά και με ένα δεκαδικό ψηφίο. Σε περίπτωση που απαιτείται διαφορετική στοίχιση θα δίνονται οι απαραίτητες διευκρινήσεις.
 6. Για την εμφάνιση πολλών δεδομένων στην ίδια γραμμή θα τυπώνεται ένας κενός χαρακτήρας ανά δεδομένο. Για την αναπαράσταση του κενού χαρακτήρα στις εκφωνήσεις χρησιμοποιείται η κάτω παύλα `-underscore-` “ ”.
-

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, w: real;
  Code, Class: char;

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

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

- i) `Calculate(u, v, two, p, q, Code);`
- ii) `Calculate(pi, u, w, two, Day, Year, Class);`
- iii) `Calculate(Hours, pi, two, Day, Year, Class);`
- iv) `while u>0 do Calculate(Rate, u, Day+2, p, q, 'm');`
- v) `Calculate(0, Hours, (p+1)/2, Day, year, Code);`
- vi) `while Amount>0 do Calculate(two, Amount, Day, p+q, Day, Class);`

2. Πληκτρολογήστε και εκτελέστε το παρακάτω πρόγραμμα . Εξηγήστε τα αποτελέσματα που προκύπτουν.

```
PROGRAM Hard_to_believe;
VAR
  Number :integer;

PROCEDURE Double_and_add(VAR first,second :integer);
```

```

BEGIN
    first:=2*first;
    second:=1+second;
END; (*OF DOUBLE_AND_ADD*)

PROCEDURE Add_And_Double(VAR first,second:integer);
BEGIN
    first:=1+first;
    second:=2*second;
END;(*OF ADD_AND_DOUBLE*)

```

```

BEGIN
    Number:=3;
    Double_and_add(number,number);
    writeln(number);
    Number:=3;
    Add_and_Double(number,number);
    writeln(number);
END.

```

3. Εξηγήστε τα αποτελέσματα των παρακάτω προγραμμάτων:

A.

```

PROGRAM nosense;
VAR
    thing :integer;

PROCEDURE cheat(var hee,haw:integer);
begin
    hee:=-1;
    haw:=-hee;
end;

BEGIN
    thing:=1;
    cheat(thing,thing);
    writeln(thing);
END.

```

B.

```

PROGRAM rubbish;
VAR
    thing : integer;

PROCEDURE liar(VAR hee : integer; haw : integer);
begin
    hee:=10*haw;
end;

BEGIN
    thing:=10;
    liar(thing,thing);
    writeln(thing);
END.

```

4. Να γραφεί πρόγραμμα που θα περιλαμβάνει 3 διαδικασίες με ονόματα:
- leftjust(S, L, X, Y),
 - rightjust(S, L, X, Y) και
 - centerjust(S, L, X, Y)
- που θα στοιχίζουν αντίστοιχα ένα αλφαριθμητικό S αριστερά, δεξιά και στο κέντρο ενός πεδίου καθορισμένου μήκους L (byte) σε μία συγκεκριμένη θέση της οθόνης με συντεταγμένες X, Y (byte). Το μήκος του πεδίου μπορεί να είναι ίσο ή μεγαλύτερο του μήκους του αλφαριθμητικού (μέγιστο μήκος 80 χαρακτήρες). Το αλφαριθμητικό θα εμφανίζεται μετά την στοίχιση σε reverse video (μαύρο χρώμα χαρακτήρων σε άσπρο φόντο). Τα αλφαριθμητικά, το μήκος του πεδίου και οι συντεταγμένες X,Y στην οθόνη θα διαβάζονται από το κυρίως πρόγραμμα και στη συνέχεια θα εκτελούνται διαδοχικά όλες οι παραπάνω διαδικασίες.
5. Να γραφεί πρόγραμμα που θα περιλαμβάνει μια διαδικασία με όνομα Delspaces(OldS, NewS) η οποία θα αφαιρεί τους κενούς χαρακτήρες (#32) που υπάρχουν στην αρχή και το τέλος ενός αλφαριθμητικού επιστρέφοντας ένα νέο αλφαριθμητικό που θα έχει προφανώς διαφορετικό μήκος. Το αρχικό αλφαριθμητικό (OldS) θα διαβάζεται από το κυρίως πρόγραμμα και μετά την εκτέλεση του προγράμματος θα εμφανίζονται το τελικό αλφαριθμητικό (NewS) και το μήκος του.
6. a) Να γίνει πρόγραμμα που θα περιλαμβάνει μια διαδικασία η οποία θα σχεδιάζει στην οθόνη ένα ορθογώνιο πλαίσιο, αν δίνονται το πλάτος και το ύψος του ($3 \leq \text{πλάτος} \leq 80$ και $3 \leq \text{ύψος} \leq 25$), με τη χρήση των κατάλληλων χαρακτήρων του κώδικα ASCII. Το πλαίσιο θα πρέπει να εμφανίζεται κεντραρισμένο στην οθόνη και θα υπάρχει η δυνατότητα επιλογής απλών ή διπλών γραμμών περιγράμματος.
- b) Να γραφεί μια διαδικασία με όνομα Random_Gen(A,B), η οποία χρησιμοποιώντας τη συνάρτηση Random θα δημιουργεί τυχαίους αριθμούς μεταξύ των ορίων A,B (βλέπε βιβλίο Κ. Κ.Τσούρου σελ. 26 τύπος (1)). Στη συνέχεια να γραφεί πρόγραμμα που θα καλεί την Random_Gen για τη δημιουργία τυχαίων τιμών ύψους - πλάτους αποδεκτών για τις διαστάσεις της οθόνης (25x80) και δεδομένου του τύπου γραμμής περιγράμματος θα σχεδιάζει το αντίστοιχο πλαίσιο στην οθόνη.
7. Να γραφεί μια διαδικασία με όνομα PETC(X,Y) (από τα αρχικά των 'PRESS ENTER TO CONTINUE' που σημαίνει πληκτρολογήστε ENTER για συνέχεια) η οποία θα τυπώνει το παραπάνω μήνυμα σε μια συγκεκριμένη θέση (X, Y) (byte) της οθόνης και θα περιμένει μέχρις ότου ο χρήστης να πληκτρολογήσει ENTER. Στη συνέχεια γράψτε ένα πρόγραμμα, για να ελέγξετε την διαδικασία, που θα εμφανίζει σε σελίδες οθόνης ανά 20 τους χαρακτήρες του κώδικα ASCII.
8. Δίνονται οι ισοτιμίες των παρακάτω νομισμάτων με την δραχμή (οι τιμές υπάρχουν στις οικονομικές στήλες των εφημερίδων) :
- ECU (Ευρωπαϊκή Νομισματική Μονάδα)
 DEM (Μάρκο Γερμανίας)
 FRF (Φράγκο Γαλλίας)
 USD (Δολάριο Αμερικής)
- Θεωρώντας ότι οι τιμές αυτές είναι σταθερές του κυρίως προγράμματος να γραφεί μια procedure που θα δέχεται ως παράμετρο εισόδου ένα ποσό σε οποιοδήποτε από τα πέντε νομίσματα και θα επιστρέφει τις ισοτιμίες στα υπόλοιπα τέσσερα. Στη συνέχεια να κληθεί η διαδικασία αυτή για την εμφάνιση ενός πίνακα ισοτιμιών για οποιοδήποτε νόμισμα ως βασικό αν δίνονται: η τιμή εκκίνησης (π.χ. 10000 Δρ.), το βήμα (π.χ. 10000 Δρ) και η τελευταία τιμή του πίνακα (π.χ. 500000 Δρ). Η μορφή του πίνακα θα είναι η εξής:
- επικεφαλίδα με πρώτο το βασικό νόμισμα και στη συνέχεια τα υπόλοιπα τέσσερα σε δεξιά στοίχιση των 12 χαρακτήρων
 - στήλες με τις ισοτιμίες σε δεξιά στοίχιση των 12 χαρακτήρων με ακρίβεια 2 δεκαδικών ψηφίων
- π.χ. αν βασικό νόμισμα είναι η δραχμή με τιμή εκκίνησης 1000, τελική τιμή 10000 και βήμα 1000 τότε τα αποτελέσματα θα εμφανιστούν ως εξής:

DRH	ECU	DEM	FRF	USD
1000.00	3.04	5.96	20.00	3.61
2000.00	6.08	11.93	40.00	7.21
.....
10000.00	30.40	59.65	200.00	36.07

- Η είσοδος των δεδομένων γίνεται με την εξής σειρά:
κωδικός βασικού νομίσματος (DRH \Rightarrow 1, ECU \Rightarrow 2, DEM \Rightarrow 3, FRF \Rightarrow 4, USD \Rightarrow 5) (byte)
τιμή εκκίνησης του πίνακα ισοτιμιών (real)
τελική τιμή του πίνακα ισοτιμιών (real)
βήμα (real)
9. Να γραφεί μια διαδικασία με όνομα $\text{swap}(x, y)$ που θα εναλλάσσει τις τιμές 2 ακεραίων μεταβλητών x και y (η τιμή της x θα δοθεί στην μεταβλητή y και της y στην x). Στη συνέχεια να γίνει ένα πρόγραμμα το οποίο θα διαβάζει τρεις ακέραιους αριθμούς και θα τους καταχωρεί στις μεταβλητές A, B, C , θα τους ταξινομήσει κατά αύξουσα σειρά εναλλάσσοντας τις τιμές τους μεταξύ των μεταβλητών με τη βοήθεια της διαδικασίας $\text{swap}(x, y)$, έτσι ώστε $A < B < C$. Τέλος θα εμφανίζει τις ταξινομημένες τιμές των μεταβλητών A, B και C .
10. Η Ε.Μ.Υ. καταγράφει ανά 8 ώρες τις θερμοκρασίες 10 πόλεων της Ελλάδος (τα ονόματα των πόλεων είναι κωδικοποιημένα με αριθμούς: 1=Θεσσαλονίκη, 2=Αθήνα κλπ). Να γραφεί πρόγραμμα που θα διαβάζει τις θερμοκρασίες (μεταβλητές τύπου Real) ενός εικοσιτετραώρου για κάθε πόλη (θερμοκρασία 1^{ου} 8ώρου, 2^{ου} 8ώρου, 3^{ου} 8ώρου για κάθε πόλη). Στη συνέχεια να υπολογίζει και να εμφανίζει τον εθνικό μέσο όρο (μεταβλητή τύπου Real) και για κάθε πόλη τη μέση θερμοκρασία της και τη μέγιστη απόκλιση της από τον εθνικό μέσο όρο (μεταβλητές τύπου Real).
- Τα αποτελέσματα να εμφανίζονται με την παρακάτω μορφή:
εθνικός μέσος όρος θερμοκρασίας
μέση θερμοκρασία 1^{ης} πόλης_μέγιστη απόκλιση της 1^{ης} πόλης από τον εθνικό μέσο όρο
μέση θερμοκρασία 2^{ης} πόλης_μέγιστη απόκλιση της 2^{ης} πόλης από τον εθνικό μέσο όρο
.....
μέση θερμοκρασία 10^{ης} πόλης_μέγιστη απόκλιση της 10^{ης} πόλης από τον εθνικό μέσο όρο.
11. Να γραφεί ένα πρόγραμμα που θα περιλαμβάνει μια διαδικασία με όνομα menuP , η οποία θα επιτρέπει την επιλογή εργασίας από ένα μενού επιλογών με τη χρήση των βελών κίνησης (πάνω-κάτω) του δρομέα. Η τρέχουσα επιλογή θα είναι φωτισμένη (reverse video) και με ENTER θα εμφανίζεται σε μια νέα οθόνη μόνο ένα χαρακτηριστικό μήνυμα (για παράδειγμα 'KANATE THN EPILOGH 1') και στη συνέχεια θα εμφανίζεται και πάλι το μενού επιλογών. Οι διαθέσιμες επιλογές θα καταχωρούνται σε μονοδιάστατο πίνακα N θέσεων, όπου N ο αριθμός επιλογών, δεδομένος και σταθερός. Το πρόγραμμα θα τερματίζει όταν επιλεγεί η N -οστή επιλογή π.χ. 'ΤΕΛΟΣ'.
12. Να γραφεί ένα πρόγραμμα που θα περιλαμβάνει μια διαδικασία με το όνομα menuL , η οποία θα επιτρέπει την επιλογή εργασίας από ένα μενού επιλογών με τη χρήση των βελών κίνησης (αριστερά ή δεξιά) του δρομέα. Η τρέχουσα επιλογή θα είναι φωτισμένη και με ENTER θα εμφανίζεται σε μια νέα οθόνη μόνο ένα χαρακτηριστικό μήνυμα (για παράδειγμα 'KANATE THN EPILOGH 1') και στη συνέχεια θα εμφανίζεται και πάλι το μενού επιλογών. Οι διαθέσιμες επιλογές θα καταχωρούνται σε μονοδιάστατο πίνακα N θέσεων, όπου N ο αριθμός επιλογών, δεδομένος και σταθερός. Το πρόγραμμα θα τερματίζει όταν επιλεγεί η N -οστή επιλογή π.χ. 'EXIT'.
13. Γράψτε ένα πρόγραμμα καθοδηγούμενο από πίνακα επιλογών (menu-driven program) το οποίο μετατρέπει διάφορα μεγέθη από ένα είδος μονάδων σε άλλο:
- από βαθμούς Fahrenheit (real) σε βαθμούς Celsius (real) ($C = 5/9 * (F - 32)$) κι αντιστρόφως,
 - από λεπτά (real) σε ώρες (real) και
 - από πόδια (real) σε μέτρα (real) ($1 \text{ πόδι} = 0.3048 \text{ μέτρα}$).
- Το μενού επιλογών να έχει την παρακάτω μορφή:
- AVAILABLE OPTIONS:
1. CONVERT MINUTES TO HOURS
 2. CONVERT FEET TO METERS
 3. CONVERT DEGREES CELSIUS TO DEGREES FAHRENHEIT
 4. CONVERT DEGREES FAHRENHEIT TO DEGREES CELSIUS
 5. QUIT
14. Γράψτε μια διαδικασία η οποία: θα στρογγυλοποιεί έναν δεδομένο ακέραιο αριθμό στην πλησιέστερη δεκάδα, εκατοντάδα, χιλιάδα κλπ (ο χρήστης θα προσδιορίζει την επιθυμητή στρογγυλοποίηση) και θα

βρίσκει το υπόλοιπο -το λεγόμενο "σφάλμα στρογγυλοποίησης"- που θα προκύπτει κάθε φορά. Αν για παράδειγμα ο αριθμός 17579 στρογγυλοποιηθεί στην πλησιέστερη εκατοντάδα (3ο ψηφίο από δεξιά), θα επιστρέφεται ο αριθμός 17600 και το υπόλοιπο -21, ενώ αν ο αριθμός 17579 στρογγυλοποιηθεί στην πλησιέστερη δεκάδα (2ο ψηφίο από δεξιά), θα επιστρέφεται ο αριθμός 17580 και το υπόλοιπο -1. Η κλήση της διαδικασίας θα γίνεται με τέσσερις παραμέτρους: τον ακέραιο αριθμό που δίνεται, το ψηφίο στο οποίο θα γίνεται η στρογγυλοποίηση, το νέο αριθμό που προκύπτει και το υπόλοιπο.

- Η είσοδος των δεδομένων θα γίνεται με την εξής σειρά:
ακέραιος αριθμός προς στρογγυλοποίηση (integer), ψηφίο στρογγυλοποίησης (byte)
- Τα αποτελέσματα που θα εμφανίζονται είναι:
ο στρογγυλοποιημένος αριθμός (integer), το υπόλοιπο στρογγυλοποίησης (integer)

15. Γράψτε μια διαδικασία με το όνομα `ValidDate` η οποία θα δέχεται 3 ακέραιους αριθμούς που θα δηλώνουν κατά σειρά την ημέρα, το μήνα και το έτος μιας ημερομηνίας και θα επιστρέφει μια Boolean μεταβλητή με την τιμή `TRUE` ή `FALSE` ανάλογα με το αν η ημερομηνία αυτή είναι αποδεκτή ή όχι (αν δηλαδή υπάρχει ή όχι στο ημερολόγιο). Θα πρέπει να συμπεριληφθούν και τα δίσεκτα έτη. Στο τέλος το πρόγραμμα θα εμφανίζει την τιμή της Boolean μεταβλητής.

16. Γράψτε ένα πρόγραμμα το οποίο θα δέχεται αγνώστου πλήθους τετράδες δεδομένων. Κάθε τετράδα θα περιλαμβάνει τα εξής:

- τρεις ακέραιους αριθμούς (τύπου `Integer`) που θα αντιστοιχούν σε μια ημερομηνία (ημέρα, μήνας, έτος ⇒ θα δίνεται ολόκληρο, π.χ. 1999)
- έναν αριθμό που αντιστοιχεί σε ένα ποσό ΦΠΑ (`longint`)

Οι ημερομηνίες πρέπει να ελέγχονται αν είναι αποδεκτές με τη διαδικασία `ValidDate` (βλ. άσκηση 15) και να ανήκουν οπωσδήποτε στο ίδιο έτος. Το τρέχον έτος θα είναι σταθερά του προγράμματος. Η εισαγωγή των δεδομένων θα συνεχίζεται ή θα τελειώνει με τη χρήση μιας διαδικασίας `continue` η οποία θα εμφανίζει σε συγκεκριμένη θέση (κάτω δεξιά) στην οθόνη το μήνυμα 'NEA ΕΙΣΑΓΩΓΗ ΝΑΙ-ΟΧΙ' και θα επιστρέφει έναν από τους χαρακτήρες `N` ή `O` (ελληνικοί ή αγγλικοί χαρακτήρες) ανάλογα με το αν η διαδικασία εισαγωγής θα συνεχιστεί ή όχι.

Μετά την εισαγωγή όλων των δεδομένων, το πρόγραμμα θα εμφανίζει τα αποτελέσματα κάθε δίμηνου του έτους καθώς και τα συνολικά ποσά για το τρέχον έτος σύμφωνα με την παρακάτω μορφή:

ΔΙΜΗΝΟ	ΠΛ. ΕΓΓΡΑΦΩΝ	ΣΥΝ. ΠΟΣΟ ΦΠΑ

		--
1	4	1256480
2	0	0
..
6	6	456500

		--
ΣΥΝΟΛΑ: