



**Δ.Π.Θ**

**Εισαγωγή στην Επιστήμη  
των Υπολογιστών**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ**

# **2ο Φυλλάδιο Ασκήσεων**

## **(Ενδεικτικές Λύσεις)**

**Dr. Αθανάσιος Μπαλαφούτης**  
Εργαστηριακό Διδακτικό Προσωπικό  
Τομέας Συστημάτων Παραγωγής  
Εργαστήριο Ρομποτικής και Αυτοματισμών  
abalafou@pme.duth.gr  
Γραφείο 304, τηλ.: 25410 – 79892

# Άσκηση 1



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Να γραφεί πρόγραμμα σε γλώσσα C που θα υπολογίζει τον δείκτη ψυχρότητας αέρα  $w$  (wind chill factor) από τον τύπο:

$$w = 91.4 - (0.474677 - 0.020425 \cdot v + 0.303107 \cdot \sqrt{v}) \cdot (91.4 - t)$$

όπου  $v$  η ταχύτητα αέρα σε **mph** και  $t$  η θερμοκρασία του αέρα σε **°Fahrenheit**.

Η συνάρτηση `main()`:

1. θα δέχεται από το πληκτρολόγιο (χρήση της `scanf`) τη θερμοκρασία σε **°Celsius** και την ταχύτητα του ανέμου σε **km/h**
2. θα μετατρέπει τη θερμοκρασία και την ταχύτητα στις μονάδες που απαιτεί ο παραπάνω τύπος
3. θα εμφανίζει με χρήση της `printf` τις τιμές των δεδομένων (θερμοκρασία σε **°Celsius** και **°Fahrenheit**, ταχύτητα αέρα σε **km/h** και **mph**) και την τιμή του δείκτη ψυχρότητας αέρα  $w$ .

(ΥΠΟΔΕΙΞΗ : πρέπει να αναζητήσετε τις αντιστοιχίες των μονάδων που απαιτούνται)

# Άσκηση 1 - Μετατροπές Μονάδων



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Μετατροπή βαθμών Κελσίου σε βαθμούς Φαρενάιτ:

$$\text{Fahrenheit } (^{\circ}\text{F}) = \text{Celsius} * 9/5 + 32$$

Μετατροπή Χιλιομέτρων σε Μίλια:

$$\text{mph} = \text{kmph} / 1.60934$$

# Άσκηση 1 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <math.h>
int main(){
    float C, F, mph, kmph, w;
    printf("Δώσε τη θερμοκρασία σε βαθμούς Κελσίου: ");
    scanf("%f", &C);
    printf("Δώσε την ταχύτητα σε χιλιόμετρα: ");
    scanf("%f", &kmph);

    F = C * 9 / 5 + 32;
    mph = kmph / 1.60934;

    printf("Θερμοκρασία: %.2f βαθμοί Κελσίου, %.2f βαθμοί Φαρενάιτ \n", C, F);
    printf("Ταχύτητα: %.2f kmph, %.2f mph \n", kmph, mph);

    w = 91.4 - (0.474677 - 0.020425 * mph + 0.303107 * sqrt(mph)) * (91.4 - F);
    printf("Δείκτης ψυχρότητας αέρα: %.2f \n", w);
    return 0;
}
```

# Άσκηση 2



Να γραφεί πρόγραμμα σε γλώσσα C που θα υλοποιεί τα βήματα που ακολουθούν :

- Έστω  $M=44$  και  $N=28$  ( $M, N$  ακέραιοι αριθμοί)
- Όσο (while)  $M$  διάφορο του  $N$  επανέλαβε:
  - όσο  $M > N$  επανέλαβε :
    - αντικατέστησε το  $M$  με  $M-N$
  - όσο  $N > M$  επανέλαβε :
    - αντικατέστησε το  $N$  με  $N-M$
- Εμφάνισε το  $M$
- Τέλος
- 

Το πρόγραμμα θα πρέπει να εμφανίζει όλες τις τιμές των  $M$  και  $N$  κατά τη διάρκεια εκτέλεσης του προγράμματος.

Να επαναλάβετε την εκτέλεση για τιμές  $M=14$  και  $N=24$ .

# Άσκηση 2 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

int main(){
    int M = 44, N = 28;

    while(M != N){
        printf("M = %d, N = %d \n", M, N);

        if(M > N)
            M = M - N;
        else if (N > M)
            N = N - M;
    }
    printf("M = %d, N = %d \n", M, N);
    return 0;
}
```

# Άσκηση 3



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Η παρακάτω μέθοδος υπολογισμού του αριθμού  $n$  αποδίδεται στον Αρχιμήδη :

Έστω  $A = 1$  και  $N = 6$

Επανάλαβε 10 φορές:

Αντικατέστησε το  $N$  με  $2N$

Αντικατέστησε το  $A$  με  $\sqrt{2 - \sqrt{4 - A^2}}$

Θέσε  $L = NA / 2$

Θέσε  $U = L / \sqrt{1 - A^2 / 2}$

Θέσε  $P = (U + L) / 2$  (υπολογισμός του  $n$ )

Θέσε  $E = (U - L) / 2$  (υπολογισμός του σφάλματος)

Εμφάνισε τα  $N, P, E$

Να γραφεί πρόγραμμα σε γλώσσα C για την εφαρμογή του αλγορίθμου.

# Άσκηση 3 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <math.h>

int main(){
    float A = 1, N = 6, L, U, P, E;

    for (int i = 0; i < 10; i++){
        N = 2 * N;
        A = sqrt(2 - sqrt(4 - pow(A, 2)));
        L = N * A / 2;
        U = L / sqrt(1 - pow(A, 2) / 2);
        P = (U + L) / 2;
        E = (U - L) / 2;
    }
    printf("N = %f, P = %f, E = %f \n", N, P, E);
    return 0;
}
```



# Άσκηση 4



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Να γραφεί ένα πρόγραμμα σε γλώσσα C που θα βρίσκει ποιος από τους παρακάτω αριθμούς είναι μεγαλύτερος:

$$\pi^{\pi^e}, e^{e^\pi}$$

ΥΠΟΔΕΙΞΕΙΣ:

- να συμπεριληφθεί η βιβλιοθήκη math.h
- οι μεταβλητές πρέπει να δηλωθούν ως double και να εμφανίζονται με ακρίβεια τουλάχιστον 7 δεκαδικών ψηφίων
- να χρησιμοποιηθεί η συνάρτηση pow(x,y) που αντιστοιχεί στην πράξη  $x^y$

# Άσκηση 4 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <math.h>

int main(){
    double a, b;

    a = pow(M_PI, pow(M_PI, M_E));
    b = pow(M_E, pow(M_E, M_PI));

    if(a > b)
        printf("Το π^(π^e) είναι μεγαλύτερο από το e^(e^π) \n");
    else
        printf("Το π^(π^e) είναι μικρότερο από το e^(e^π) \n");

    return 0;
}
```

# Άσκηση 5



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Οι ηλεκτρικές αντιστάσεις έχουν τρεις ζώνες χρωμάτων. Κάθε χρώμα αντιστοιχεί σε έναν ακέραιο αριθμό:

Μαύρο → 0, Καφέ → 1, Κόκκινο → 2, Πορτοκαλί → 3, Κίτρινο → 4, Πράσινο → 5, Μπλε → 6, Ιώδες → 7, Γκρι → 8 και Λευκό → 9.

Η αντίσταση υπολογίζεται ως το άθροισμα :

- 10 φορές ο αριθμός που αντιστοιχεί στο πρώτο χρώμα
- ο αριθμός που αντιστοιχεί στο δεύτερο χρώμα
- πλήθος μηδενικών τόσα όσα υποδηλώνει ο αριθμός του τρίτου χρώματος

Για παράδειγμα αν τα τρία χρώματα είναι Κόκκινο, Κίτρινο και Πορτοκαλί (με τιμές 2,4,3) η αντίσταση είναι:

$2 \cdot 10 + 4 = 24$  και 3 μηδενικά δηλ. 24000.

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει την αντίσταση για οποιοδήποτε δεδομένη διαφορετική τριάδα χρωμάτων.

# Άσκηση 5 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

int main(){
    int first, second, third, res;

    printf("Δώσε τους 3 αριθμούς των χρωμάτων: ");
    scanf("%d %d %d", &first, &second, &third);

    res = first * 10 + second;

    for(int i = 0; i < third; i++)
        res *= 10;

    printf("Η αντίσταση έχει τιμή: %d \n", res);

    return 0;
}
```

# Άσκηση 6



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Για τον υπολογισμό της τιμής του  $n!$  όταν η τιμή του  $n$  είναι μεγάλη χρησιμοποιείται ο προσεγγιστικός τύπος του Stirling :

$$n! \approx e^{-n} n^n \sqrt{2\pi n}$$

Να γραφεί πρόγραμμα σε γλώσσα C, σύμφωνα με τον παραπάνω τύπο, για τον υπολογισμό της τιμής  $n!$ .

Η τιμή του  $n$  εισάγεται με τη χρήση της scanf (να χρησιμοποιηθεί η συνάρτηση pow της math.h).

Στο ίδιο πρόγραμμα, να γραφεί το αντίστοιχο τμήμα προγράμματος για τον υπολογισμό του  $n!$  σύμφωνα με τον τύπο:

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n,$$

και να βρεθούν οι διαφορές (ως ποσοστό %) που παρουσιάζουν οι δύο μέθοδοι υπολογισμού, για διάφορες τιμές του  $n$ .

# Άσκηση 6 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <math.h>
int main(){
    double factorial_S, factorial, pososto;
    int n;
    printf("Δώσε τον αριθμό n: ");
    scanf("%d", &n);
    factorial_S = exp(-n) * pow(n, n) * sqrt(2 * M_PI * n);
    factorial = 1;
    for (int i = 2; i <= n; i++) {
        factorial *= i;
    }
    pososto = fabs(factorial - factorial_S) / factorial * 100;
    printf("Παραγοντικό του %d κατά προσέγγιση: %.21f\n", n, factorial_S);
    printf("Παραγοντικό του %d: %.01f\n", n, factorial);
    printf("Ποσοστιαία διαφορά: %.21f %%\n", pososto);
    return 0;
}
```

# Άσκηση 7



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Μια εταιρεία παραγωγής ξυλείας θέλει να δημιουργήσει ένα πίνακα για τις μηχανικές ιδιότητες των προϊόντων της. Οι διαστάσεις του ξύλου (που αποτελεί ένα συμπαγές ορθογώνιο παραλληλεπίπεδο) δίνονται ως βάση (base) και ύψος (height) σε cm, ενώ οι ιδιότητες που πρέπει να υπολογιστούν είναι :

Διατομή :	$base \times height$
Ροπή αδρανείας :	$\frac{base \times height^3}{12}$
Ροπή αντίστασης :	$\frac{base \times height^2}{6}$

Οι διαστάσεις που διατίθενται είναι : για τη βάση 2,4,6,8 και 10 cm και για το ύψος 2,4,6,8,10 και 12 cm. Να εμφανίσετε έναν πίνακα που θα περιλαμβάνει τις τιμές των μηχανικών ιδιοτήτων για όλους τους δυνατούς συνδυασμούς των παραπάνω διαστάσεων.

# Άσκηση 7 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <math.h>

int main(){
    int base, height, diatomi;
    float torque1, torque2;

    for(base = 2; base<=10; base+=2){
        for(height = 2; height <=12; height+=2){
            diatomi = base * height;
            torque1 = base * pow(height, 3)/12;
            torque2 = base * pow(height, 2)/6;
            printf("(%d, %d) Διατομή: %d Ροπή αδρανείας: %.2f Ροπή αντίστασης: %.2f\n", base, height,
            diatomi, torque1, torque2);
        }
    }
    return 0;
}
```



# Άσκηση 8



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Μια τριάδα θετικών ακεραίων αριθμών ονομάζεται **Πυθαγόρεια τριάδα** αν ικανοποιεί το Πυθαγόρειο θεώρημα. Το πλήθος τέτοιων τριάδων είναι άπειρο. Αν δοθούν **δύο θετικοί ακέραιοι αριθμοί**  $m$  και  $n$ , με τον περιορισμό  $m > n$ , μια Πυθαγόρεια τριάδα μπορεί να δημιουργηθεί από τους τύπους :

$$side1 = m^2 - n^2$$

$$side2 = 2mn$$

$$hypotenuse = \sqrt{side1^2 + side2^2}$$

Να γράψετε ένα πρόγραμμα σε γλώσσα C που :

- Θα δέχεται (με χρήση της scanf) τις τιμές  $m$  και  $n$
- Θα ελέγχει αν ικανοποιείται ο περιορισμός  $m > n$ 
  - Αν ισχύει ο περιορισμός τότε θα υπολογίζει και θα εμφανίζει τις τιμές της Πυθαγόρειας τριάδας που δημιουργείται από τους αριθμούς αυτούς, χρησιμοποιώντας τους παραπάνω τύπους
  - Αν δεν ισχύει ο περιορισμός θα εμφανίζει ένα κατάλληλο μήνυμα

# Άσκηση 8 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <math.h>
int main(){
    int m, n;
    float side1, side2, hypotenuse;
    printf("Δώσε τα m, n: ");
    scanf("%d %d", &m, &n);
    if(m > n){
        side1 = pow(m, 2) - pow(n, 2);
        side2 = 2 * m * n;
        hypotenuse = sqrt(pow(side1, 2) + pow(side2, 2));
        printf("side1: %.2f, side2: %.2f, hypotenuse: %.2f\n", side1, side2, hypotenuse);
    }
    else
        printf("Δεν ισχύει m > n \n");

    return 0;
}
```

# Άσκηση 9



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Η αύξηση του πληθυσμού μιας πόλης για το επόμενο έτος, **NextYr**, υπολογίζεται με βάση τον πληθυσμό του τρέχοντος έτους **CurrentYr** και τον ρυθμό αύξησης **Rate** (επιτρεπτές τιμές από 0.0 έως 4.0, αντιστοιχούν σε ποσοστά αύξησης 0.0% - 4.0%) σύμφωνα με τον παρακάτω τύπο:

$$NextYr = Rate * CurrentYr * \left(1 - \frac{CurrentYr}{1000000}\right)$$

Ο τύπος ισχύει για τιμές του πληθυσμού τρέχοντος έτους, **CurrentYr**, από 0 -1000000.

Να γραφεί ένα πρόγραμμα σε γλώσσα C που θα

- Εισάγει την τρέχουσα τιμή του πληθυσμού (0-1000000)
- Εισάγει τον ρυθμό αύξησης (επιτρεπτές τιμές 0.0 – 4.0, αντιστοιχούν σε ποσοστά αύξησης 0.0% - 4.0%)
- Θα υπολογίζει και θα εμφανίζει μετά από πόσα έτη ο πληθυσμός της πόλης θα διπλασιαστεί. Σε περίπτωση που αυτό δεν επιτυγχάνεται για τα επόμενα 30 έτη, το πρόγραμμα, μετά από 30 επαναλήψεις, θα σταματά.

# Άσκηση 9 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
int main(){
    int years;
    float rate, nextYr, currentYr, initYr;
    printf("Δώσε την τρέχουσα τιμή πληθυσμού: ");
    scanf("%f", &currentYr);
    printf("Δώσε το ρυθμό αύξησης πληθυσμού: ");
    scanf("%f", &rate);
    years = 0;
    initYr = currentYr;
    do{
        nextYr = rate * currentYr * (1- currentYr / 1000000);
        currentYr = nextYr;
        years++;
    }while(nextYr < 2 * initYr && years < 30);
    printf("Νέος πληθυσμός: %f \n", nextYr);
    printf("Χρόνια: %d \n", years);
    return 0;
}
```

# Χωρίζοντας ακέραιο στα ψηφία του



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

int main(){
    int x, digits;

    printf("Δώσε έναν ακέραιο αριθμό: ");
    scanf("%d", &x);

    digits = 0;
    while (x != 0){
        printf("%d \n", x % 10);
        x = x / 10;
        digits++;
    }
    printf("Ο αριθμός έχει %d ψηφία.\n", digits);

    return 0;
}
```

# Άσκηση 10



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Για ορισμένους ακέραιους και θετικούς αριθμούς ισχύει ότι:

ο αριθμός είναι ίσος με το γινόμενο του αθροίσματος των ψηφίων του επί το γινόμενο των ψηφίων του. Π.χ.

$$135 = (1 + 3 + 5) \cdot (1 \cdot 3 \cdot 5)$$

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει όλους τους αριθμούς που πληρούν την παραπάνω ιδιότητα και είναι μικρότεροι του 1000.

Στο τέλος πρέπει να εμφανίζεται και το πλήθος τους.

# Άσκηση 10 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
int main(){
    int x, sum, mult, result, count = 0;
    for(int i = 1; i < 1000; i++){
        x = i; sum = 0; mult = 1;
        while (x != 0){
            sum += x % 10;
            mult *= x % 10;
            x = x / 10;
        }
        result = sum * mult;
        if(result == i){
            printf("%d \n", result);
            count++;
        }
    }
    printf("Σύνολο αριθμών με αυτή την ιδιότητα: %d \n", count);
    return 0;
}
```

# Άσκηση 11



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Ένας ακέραιος αριθμός λέγεται αυτομορφικός αν το τετράγωνό του τελειώνει με τον ίδιο αριθμό

π.χ.  $5^2 = 25$ ,  $25^2 = 625$  κλπ.

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει, θα μετρά και θα εμφανίζει όλους τους αυτομορφικούς αριθμούς το τετράγωνο των οποίων είναι μικρότερο του 10000.

Ο αλγόριθμος θα εμφανίζει στο τέλος και το πλήθος αυτών των αριθμών.



# Άσκηση 11 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    int x, sq_x, count = 0;
    bool is;

    for(int i = 0; i < 100; i++){
        x = i;
        sq_x = i * i;
        is = true;
        while (x != 0){
            if(x % 10 != sq_x % 10)
                is = false;
            x = x /10;
            sq_x = sq_x /10;
        }
    }
}
```

```
        if (is){
            printf("%d \n", i);
            count++;
        }
    }
    printf("Πλήθος αυτομορφικών αριθμών: %d \n",
count);
    return 0;
}
```

# Άσκηση 12



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Ένας ακέραιος αριθμός λέγεται τριμορφικός όταν τα τελευταία ψηφία του κύβου του αριθμού είναι ο ίδιος ο αριθμός

π.χ.  $49^3 = 117649$ ,  $25^3 = 15625$  κλπ.

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει, θα μετρά και θα εμφανίζει όλους τους τριμορφικούς αριθμούς που είναι μεγαλύτεροι του 9 και μικρότεροι του 100.

Το πρόγραμμα θα εμφανίζει στο τέλος και το πλήθος αυτών των αριθμών.

# Άσκηση 12 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    int x, x_3, count = 0;
    bool is;

    for(int i = 9; i < 100; i++){
        x = i;
        x_3 = i * i * i;
        is = true;
        while (x != 0){
            if(x % 10 != x_3 % 10)
                is = false;
            x = x /10;
            x_3 = x_3 /10;
        }
    }
}
```

```
        if (is){
            printf("%d \n", i);
            count++;
        }
    }
    printf("Πλήθος τριμορφικών αριθμών: %d \n", count);
    return 0;
}
```

# Άσκηση 13



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Για ορισμένους ακέραιους αριθμούς ισχύει ότι το άθροισμα των ψηφίων του αριθμού διαιρεί τον ίδιο τον αριθμό

π.χ. για τον αριθμό 1729 ισχύει :  $1+7+2+9=19$  και  $1729/19=91$ .

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει, θα μετρά και θα εμφανίζει όλους τους αριθμούς που πληρούν την παραπάνω ιδιότητα και είναι μικρότεροι του 10000.

Ο αλγόριθμος θα εμφανίζει στο τέλος και το πλήθος αυτών των αριθμών.

# Άσκηση 13 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
int main(){
    int x, sum, count = 0;

    for(int i = 1; i < 10000; i++){
        x = i; sum = 0;

        while (x != 0){
            sum += x % 10;
            x = x / 10;
        }
        if(i % sum == 0){
            printf("%d \n", i);
            count++;
        }
    }
    printf("Σύνολο αριθμών με αυτή την ιδιότητα: %d \n", count);
    return 0;
}
```

# Άσκηση 14



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Ένας αριθμός λέγεται παλίνδρομος αν μπορεί να διαβαστεί το ίδιο από την αρχή και το τέλος του

π.χ. 2772, 4444, 9119

Να γραφεί ένα πρόγραμμα σε γλώσσα C που θα υλοποιεί τα εξής:

- Θα διαβάζει από το πληκτρολόγιο (χρήση της scanf) έναν ακέραιο θετικό τετραψήφιο αριθμό (1000 - 9999)
- Θα ελέγχει αν ο αριθμός αυτός ανήκει στο παραπάνω διάστημα τιμών
- Αν ο αριθμός δεν ανήκει στο παραπάνω διάστημα τιμών θα εμφανίζεται ένα κατάλληλο μήνυμα
- Αν ο αριθμός ικανοποιεί τους περιορισμούς θα εμφανίζονται τα ψηφία του αριθμού σε απόσταση 4 κενών το ένα από το άλλο και το κατάλληλο μήνυμα για το αν ο αριθμός είναι παλίνδρομος ή όχι.

# Άσκηση 14 - Λύση



```
#include <stdio.h>
int main(){

    int x, original_x, reversed_x;

    printf("Δώσε έναν τετραψήφιο ακέραιο:");
    scanf("%d", &x);

    if(x < 1000 || x > 9999)
        printf("Ο αριθμός δεν είναι τετραψήφιος \n");
    else{
        original_x = x;
        reversed_x = 0;

        while (x != 0){
            printf("%5d", x % 10);
            reversed_x = reversed_x * 10 + x % 10;
            x = x / 10;
        }
    }
}
```

```
if(original_x == reversed_x)
    printf("\nΟ αριθμός είναι παλίνδρομος");
else
    printf("\nΟ αριθμός δεν είναι παλίνδρομος");

}
return 0;
}
```

# Άσκηση 15

---



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Να γραφεί ένα πρόγραμμα σε γλώσσα C που θα εμφανίζει όλες τις δυνάμεις του 2 που είναι μικρότερες από μια δεδομένη τιμή **maxpower**.

Σε κάθε γραμμή θα εμφανίζεται η δύναμη **n** και η αντίστοιχη τιμή **2<sup>n</sup>**



# Άσκηση 15 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <math.h>

int main(){

    int n, maxpower;

    printf("Δώσε το maxpower: ");
    scanf("%d", &maxpower);

    n = 1;

    while(pow(2, n) < maxpower){
        printf("n: %d, 2^n: %d \n", n, (int)pow(2, n));
        n++;
    }

    return 0;
}
```

# Άσκηση 16



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Να βρεθούν και να εμφανιστούν όλοι οι μονοψήφιοι και διψήφιοι θετικοί ακέραιοι αριθμοί που ικανοποιούν την παρακάτω ιδιότητα:

Το άθροισμα των ψηφίων του τετραγώνου του αριθμού για τους μονοψήφιους θετικούς ακέραιους αριθμούς ή το άθροισμα του αριθμού που προκύπτει από τα 2 τελευταία ψηφία του τετραγώνου του αριθμού και του αριθμού που προκύπτει από τα υπόλοιπα αριστερότερα ψηφία για τους διψήφιους να είναι ίσο με τον αριθμό.

Παραδείγματα:

Μονοψήφιος : 9	$9^2=81$	$8+1=9$	ΕΙΝΑΙ
Διψήφιος : 14	$14^2=196$	$1+96=97$	ΔΕΝ ΕΙΝΑΙ !
Διψήφιος : 45	$45^2=2025$	$20+25=45$	ΕΙΝΑΙ

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει αυτούς τους αριθμούς. Στο τέλος θα πρέπει να εμφανίζεται και το πλήθος αυτών των αριθμών.

# Άσκηση 16 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <math.h>

int main(){

    int x, x_2, sum, count = 0;

    for(x = 1; x < 100; x++){
        x_2 = pow(x, 2);

        if(x < 10)
            sum = (x_2 / 10) + (x_2 % 10);
        else
            sum = (x_2 / 100 + x_2 % 100);
```

```
        if(sum == x){
            printf("%d \n", x);
            count++;
        }
    }
    printf("Πλήθος αριθμών: %d \n", count);

    return 0;
}
```

# Άσκηση 17



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Δύο πρώτοι αριθμοί λέγονται δίδυμοι αν διαφέρουν κατά 2 (π.χ. 3 και 5, 101 και 103).

Να γραφεί ένα πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει όλους τους θετικούς δίδυμους πρώτους αριθμούς που είναι μικρότεροι του 10000.

ΥΠΟΔΕΙΞΗ-1:

για να ελέγξετε το πρόγραμμά σας : οι δύο τελευταίοι δίδυμοι που είναι μικρότεροι από 10000 είναι οι αριθμοί **9929** και **9931**.

ΥΠΟΔΕΙΞΗ-2 :

Ο παρακάτω κώδικας προγράμματος βρίσκει και εμφανίζει τους θετικούς πρώτους αριθμούς που υπάρχουν μέχρι ένα δεδομένο όριο  $n$  ( $n = \text{γνωστό}$ ).

# Πρώτοι Αριθμοί



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

void main(){
    int posprime, posDiv,n;

    do{
        printf("limit=? ( > 0 please) ");
        scanf("%d", &n);
    }while (n <= 0);

    printf("Primes < = %5d\n", n);

    for ( posprime = 2; posprime <= n; posprime++){
        for (posDiv = 2; posDiv < posprime; posDiv++){
            if ( posprime % posDiv == 0) break;
            if (posDiv == posprime)
                printf("%3d ", posprime);
        }
    }
}
```

# Άσκηση 17 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

int main(){
    int posprime, posDiv,n, prev_posprime;

    prev_posprime = 2;

    for (posprime = 2; posprime < 10000; posprime++){
        for (posDiv = 2; posDiv < posprime; posDiv++){
            if ( posprime % posDiv == 0) break;
            if (posDiv == posprime){
                if(posprime - prev_posprime == 2)
                    printf("%d - %d \n", prev_posprime, posprime);
                prev_posprime = posprime;
            }
        }
    }
    return 0;
}
```

# Άσκηση 18



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει όλους τους θετικούς τετραψήφιους ακέραιους αριθμούς ( [1000,9999] ) που ικανοποιούν μία από τις παρακάτω δύο ιδιότητες:

- Το άθροισμα των τετραγώνων των δύο αριθμών που προκύπτουν από τα 2 πρώτα και τα δύο τελευταία ψηφία του αριθμού ισούται με τον αριθμό π.χ.  $1233 = 12^2 + 33^2$
- Ο κύβος του αθροίσματος των ψηφίων του αριθμού ισούται με τον αριθμό π.χ.  $4913 = (4+9+1+3)^3$

Στο τέλος θα πρέπει να εμφανίζεται και το πλήθος αυτών των αριθμών.

# Λογικοί Τελεστές



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

NOT Truth Table

A	B
0	1
1	0

Operator	Description
&&	AND
	OR
!	NOT
!=	NOT EQUAL TO
&	BITWISE AND
	BITWISE OR
^	BITWISE XOR



# Άσκηση 18 - Λύση



```
#include <stdio.h>
#include <stdbool.h>
#include <math.h>

int main(){

    int x1, x2, sum;
    bool property1, property2;

    for(int i = 1000; i < 10000; i++){
        property1 = false;
        property2 = false;

        x1 = i / 100;
        x2 = i % 100;
        if(pow(x1, 2) + pow(x2, 2) == i)
            property1 = true;
```

```
        sum = 0;
        x1 = i;
        while(x1 != 0){
            sum += x1 % 10;
            x1 = x1 / 10;
        }
        if(pow(sum, 3) == i)
            property2 = true;

        // XOR operator
        if(property1 ^ property2 == true)
            printf("%d \n", i);
    }
    return 0;
}
```

# Άσκηση 19



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Ένα δάνειο 8000€ αποπληρώνεται με μηνιαίες δόσεις των 300€ και με ετήσιο επιτόκιο 10%. Να γραφεί πρόγραμμα σε γλώσσα C που θα εμφανίζει τον παρακάτω μηνιαίο πίνακα αποπληρωμής του δανείου μέχρι αυτό να εξοφληθεί. Για κάθε μήνα ο τόκος υπολογίζεται στο ποσό του αρχικού κεφαλαίου που παραμένει ανεξόφλητο.

A/A ΜΗΝΑ	ΑΡΧΙΚΟ ΠΟΣΟ	ΠΟΣΟ ΤΟΚΩΝ	ΠΟΣΟ ΔΑΝΕΙΟΥ	ΥΠΟΛΕΙΠΟΜΕΝΟ ΠΟΣΟ
1	8000.000000	66.666667	233.333333	7766.666667
2	7766.666667	64.722223	235.277777	7531.388889
3	7531.388889	.....	.....	.....
...	....	.....	.....	....
...	.....	.....	.....	0.000000

Σημείωση. Αν το ετήσιο επιτόκιο είναι 10%, το μηνιαίο θα είναι:  $10 / 12 = 0.8333\%$

# Άσκηση 19 - Λύση (1/2)



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

int main(){

    float arxiko_poso, tokoi, poso_daneiou, ypoloipo;
    int month;

    arxiko_poso = 8000;
    month = 0;
    ypoloipo = 8000;
    printf("ΜΗΝΑΣ \t ΑΡΧΙΚΟ ΠΟΣΟ \t ΠΟΣΟ ΤΟΚΩΝ \t ΠΟΣΟ ΔΑΝΕΙΟΥ \t ΥΠΟΛΕΙΠΟΜΕΝΟ ΠΟΣΟ \n");

    // ... συνεχίζεται ...
```

# Άσκηση 19 - Λύση (2/2)



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
//...

while(1){
    tokoi = arxiko_poso * 0.833333 / 100;
    poso_daneiou = 300 - tokoi;
    ypoloipo -= poso_daneiou;
    month++;
    if(ypoloipo < 0) break;
    printf("%d \t %f \t %f \t %f \t %f \n", month, arxiko_poso, tokoi, poso_daneiou, ypoloipo);
    arxiko_poso -= poso_daneiou;
}
poso_daneiou = arxiko_poso + tokoi;
ypoloipo = 0;
printf("%d \t %f \t %f \t %f \t %f \n", month, arxiko_poso, tokoi, poso_daneiou, ypoloipo);

return 0;
}
```

# Άσκηση 20



Να γράψετε πρόγραμμα σε γλώσσα C για την εύρεση της τετραγωνικής ρίζας ενός αριθμού σύμφωνα με τα παρακάτω βήματα (μέθοδος Newton).

Το πρόγραμμα :

1. Θα διαβάζει από το πληκτρολόγιο τον αριθμό  $x$  για τον οποία αναζητούμε την τετραγωνική ρίζα. Αν ο αριθμός είναι αρνητικός το πρόγραμμα θα σταματά.
2. Θα χρησιμοποιεί ως πρώτη τιμή για την τετραγωνική ρίζα το  $\frac{1}{2}$  του αριθμού  $x$ .
3. Θα χρησιμοποιεί μια δομή `while` για να βρει τη ρίζα με διαδοχικές προσεγγίσεις με μια δοσμένη ακρίβεια  $\epsilon=10^{-5}$ . Η νέα τιμή σε κάθε επανάληψη θα υπολογίζεται από τον τύπο:  
νέα τιμή =  $0.5 \cdot (x/\text{προηγούμενη τιμή} + \text{προηγούμενη τιμή})$
4. Θα εμφανίζει σε κάθε βήμα τον αύξοντα αριθμό επανάληψης, την νέα τιμή, την αμέσως προηγούμενη τιμή και τη διαφορά αυτών των δύο τιμών.

# Άσκηση 20 - Λύση



Δ.Π.Θ

Εισαγωγή στην Επιστήμη  
των Υπολογιστών

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(){
    float x, new_value, old_value;
    int i = 1;
    printf("Δώσε τον αριθμό: ");
    scanf("%f", &x);
    if(x < 0) exit(0);
    new_value = x / 2;
    old_value = 0;
    while(fabs(new_value - old_value) > 1e-5){
        old_value = new_value;
        new_value = 0.5 * (x / old_value + old_value);
        printf("%d, %f, %f, %f \n", i, new_value, old_value, fabs(new_value - old_value));
        i++;
    }
    return 0;
}
```