

### Άσκηση 1

Ποια είναι η έξοδος του παρακάτω προγράμματος;

```
#include <stdio.h>

int main(void)
{
    int *ptr, i=10, j=20, k=30;

    ptr = &i;
    *ptr = 40;

    ptr = &j;
    *ptr += i;

    ptr = &k;
    *ptr += i+j;

    printf("%d%d%d\n", i, j, k);

    return 0;
}
```

Απάντηση: 40 60 130

### Άσκηση 2

Να γραφεί ένα πρόγραμμα το οποίο να χρησιμοποιεί έναν δείκτη για να διαβάσει έναν πραγματικό αριθμό τύπου **double** και να εμφανίσει το δεκαδικό του μέρος. Για παράδειγμα, αν ο χρήστης εισαγάγει -7.21, το πρόγραμμα να εμφανίσει 0.21.

```
#include <stdio.h>

int main(void)
{
    double *ptr, x;

    ptr = &x;

    printf("Enter a number: ");
    scanf("%lf", ptr);

    if(*ptr < 0)
        *ptr = -*ptr;

    printf("%f\n", *ptr - (int)*ptr);

    return 0;
}
```

### Άσκηση 3

Ποια είναι η έξοδος του παρακάτω προγράμματος;

```
#include <stdio.h>

int main(void)
{
    int *ptr, i, j=1, a[4] = {j, j+1, j+2, j+3};

    for(i=0; i<4; i++)
    {
        ptr = a+i;
        printf("%d ",a[*ptr]%4);
    }

    return 0;
}
```

Απάντηση: 2 3 4 1

### Άσκηση 4

Να γραφεί ένα πρόγραμμα το οποίο να θέτει τυχαίες τιμές (στο [-5, 5]) στα στοιχεία ενός 5×5 πίνακα ακεραίων και να ελέγχει αν το άθροισμα των στοιχείων της κύριας διαγωνίου του είναι ίσο με αυτό της δευτερεύουσάς του. Να χειριστείτε τον πίνακα με σημειογραφία δείκτη.

```
#include <stdio.h>
#include <stdlib.h> /* rand() is defined here */
#define SIZE 5

int main(void)
{
    int i, j, sum1, sum2, arr[SIZE][SIZE];

    for(i=sum1=sum2=0; i < SIZE; i++)
    {
        for(j=0; j < SIZE; j++)
        {
            *(arr[i]+j) = rand()%11 - 5;
            if(i == j)
                sum1 += *(arr[i]+j);
            if(i+j == SIZE-1)
                sum2 += *(arr[i]+j);
        }
    }

    if(sum1 == sum2)
        printf("Same\n");
    else
        printf("Different\n");

    return 0;
}
```

### Άσκηση 5

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει ακεραίους και να τους αποθηκεύει σε έναν 5×5 πίνακα και να εμφανίζει το άθροισμα των στοιχείων της στήλης με το μεγαλύτερο άθροισμα, καθώς και το άθροισμα των στοιχείων της γραμμής με το μεγαλύτερο άθροισμα. Να χειριστείτε τον πίνακα με σημειογραφία δείκτη.

```
#include <stdio.h>
#define SIZE 5

int main(void)
{
    int i, j, rowsum, colsum, rowmax, colmax, arr[SIZE][SIZE];

    for(i=0; i < SIZE; i++)
        for(j = 0; j < SIZE; j++)
        {
            printf("Enter number: ");
            scanf("%d", arr[i]+j);
        }

    for(i=0, rowsum=colsum=0; i < SIZE; i++)
    {
        rowsum += *(arr[0]+i); /* Άθροισμα στοιχείων 1ης γραμμής */
        colsum += *arr[i];    /* Άθροισμα στοιχείων 1ης στήλης */
    }
}
```

```
rowmax = rowsum;
colmax = colsum;

for(i=1; i < SIZE; i++)
{
    for(j=0, rowsum=colsum=0; j < SIZE; j++)
    {
        rowsum += *(arr[i]+j);
        colsum += *(arr[j]+i);
    }
    if(rowsum > rowmax)
        rowmax = rowsum;
    if(colsum > colmax)
        colmax = colsum;
}

printf("Max row sum = %d\n", rowmax);
printf("Max column sum = %d\n", colmax);

return 0;
}
```

### Άσκηση 6

Να γράψετε ένα πρόγραμμα που να δημιουργεί έναν 2-Δ πίνακα ( $M \times N$ ) ακεραίων στη `main()` και θα τον αρχικοποιεί με συνάρτηση ενός ορίσματος (ο πίνακας) από το πληκτρολόγιο. Οι γραμμές ( $M$ ) και οι στήλες ( $N$ ) του πίνακα θα δηλώνονται με `#define`. Τέλος, θα καλεί επαναληπτικά για κάθε γραμμή μια συνάρτηση που θα υπολογίζει το άθροισμα των τετραγώνων της γραμμής του πίνακα και θα το εκχωρεί στο αντίστοιχο στοιχείο ενός δεύτερου πίνακα.

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

#define M 10
#define N 15

void initialize_matrix(int A[][N]);
int sumsqr(int A[]);

int main(void)
{
    int i, j, arr[M][N], ssq[M] = {0};
    int Fnorm;

    initialize_matrix(arr);

    for(i=0; i<M; i++)
    {
        for(j=0; j<N; j++)
            printf("%d ",arr[i][j]);
        printf("\n");
    }

    for(i=0; i<M; i++)
        ssq[i] = sumsqr(arr[i]); /* <==> ssq[i] = sumsqr(*(arr+i)); */

    for(i=0, Fnorm=0; i<M; i++)
        Fnorm += ssq[i];

    printf("Frobenius norm of matrix arr = %d\n",Fnorm);

    return 0;
}
```

```

void initialize_matrix(int A[][N])
{
    int i,j;

    for(i=0; i<M; i++)
        for(j=0; j<N; j++)
            scanf("%d",A[i]+j);
}

#define SQR(X) ((X) * (X))

int sumsqr(int A[])
{
    int i, ssq=0;

    for(i=0; i<N; i++)
        ssq += SQR(A[i]);
    return ssq;
}

```

#### Άσκηση 7 (Για το σπίτι)

Δημιουργήστε μία συνάρτηση που να δέχεται σαν παράμετρο έναν πίνακα και να ελέγχει αν περιέχει τιμές που να επαναλαμβάνονται. Αν ναι, η συνάρτηση να επιστρέφει έναν δείκτη στο στοιχείο που επαναλαμβάνεται τις περισσότερες φορές, αλλιώς να επιστρέφει NULL. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 100 ακεραίους, να τους αποθηκεύει σε έναν πίνακα, να καλεί τη συνάρτηση και να χρησιμοποιεί την τιμή επιστροφής της για να εμφανίσει την τιμή του στοιχείου με τις περισσότερες εμφανίσεις. (Σημ. Αν υπάρχουν παραπάνω από ένα στοιχεία με τον ίδιο αριθμό μέγιστων εμφανίσεων, το πρόγραμμα να εμφανίζει το πρώτο από αυτά).