

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

```
#include <stdio.h>

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

    ptr = &x;

    printf("Enter 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[] = {j, j+1, j+2, j+3};

    for(i = 0; i < 3; i++)
    {
        ptr = a+i;
        printf("%d ", a[*ptr]);
    }

    return 0;
}
```

Απάντηση: 2 3 4

Άσκηση 4

Χρησιμοποιήστε τους δείκτες p1 και p2 και τη μεταβλητή temp και συμπληρώστε το παρακάτω πρόγραμμα, ώστε να αντιστρέψετε τα στοιχεία του πίνακα arr. Στη συνέχεια, χρησιμοποιήστε τον δείκτη p1 για να τα εμφανίσετε. Δηλαδή, η νέα μορφή του πίνακα να είναι: 2.5 9.4 -3.8 -4.1 1.3

```
#include <stdio.h>

int main(void)
{
    double arr[]={1.3,-4.1,-3.8,9.4,2.5}, temp, *p1=arr, *p2=arr+4;

    while(p1 < p2)
    {
        temp = *p1;
        *p1 = *p2;
        *p2 = temp;
        p1++; /* Δείχνει στο επόμενο στοιχείο του πίνακα. */
        p2--; /* Δείχνει στο προηγούμενο στοιχείο του πίνακα. */
    }

    printf("Reversed Array: ");
    for(p1 = arr; p1 < arr+5; p1++)
        printf("%.1f ", *p1);

    return 0;
}
```

Άσκηση 5

Να γραφεί ένα πρόγραμμα το οποίο να θέτει τυχαίες τιμές (στο [-5, 5]) στα στοιχεία ενός 5x5 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) = (int) (11*rand()/RAND_MAX) - 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 6

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει ακεραίους και να τους αποθηκεύει σε έναν 5x5 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 7

Να γράψετε ένα πρόγραμμα που να δημιουργεί έναν 2-Δ πίνακα 20 x 50 ακεραίων και να τον αρχικοποιεί με τυχαίους αριθμούς στο διάστημα [-10, 10]. Στη συνέχεια θα ελέγξει αν δύο γραμμές του έχουν το ίδιο άθροισμα και θα βγάλει σχετικό μήνυμα. Το πρόγραμμα θα δημιουργεί τον πίνακα στη main(), στη συνέχεια θα τον αρχικοποιεί με συνάρτηση ενός ορίσματος (ο πίνακας) καθώς οι γραμμές και οι στήλες του θα έχουν μνημονικά ονόματα με #define και τέλος θα καλεί επαναληπτικά για κάθε γραμμή μια συνάρτηση που θα υπολογίζει το άθροισμα των τετραγώνων της γραμμής του πίνακα το οποίο και θα εκχωρεί στο αντίστοιχο στοιχείο ενός δεύτερου πίνακα.

```

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

#define ROWS 3
#define COLS 4

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

int main(void)
{
    int i, j, arr[ROWS][COLS], ssq[ROWS] = {0};
    int Fnorm, (*p)[COLS] = arr;

    initialize_matrix(arr);
}

```

```

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

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

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

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

return 0;
}

```

```

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

    for(i=0; i<ROWS; i++)
        for(j=0; j<COLS; j++)
            scanf("%d",&A[i][j]);
}

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

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

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

```

Άσκηση 8

Να γράψετε συνάρτηση που να διαβάζει τα στοιχεία ενός 2D πίνακα γραμμικά και να την χρησιμοποιήσετε στη main() ώστε να διαβάσετε δύο 2D πίνακες με δεικτοδότηση 1D πινάκων. Στη συνέχεια να γράψετε άλλη συνάρτηση που να υπολογίζει το γινόμενο των 2 πινάκων πάλι με γραμμικούς δείκτες. Τέλος, η main() να εμφανίζει το αποτέλεσμα του γινομένου ως 2D πίνακα.

Άσκηση 9

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 γράφει 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).

```
#include <stdio.h>
#define SIZE 100
int *find(int arr[]);

int main(void)
{
    int *ptr, i, arr[SIZE];

    for(i=0; i < SIZE; i++)
    {
        printf("Enter number: ");
        scanf("%d", &arr[i]);
    }
    ptr = find(arr);
    if(ptr == NULL)
        printf("No duplicated value is found\n");
    else
        printf("The number %d appears most times\n", *ptr);
    return 0;
}
```

```
int *find(int arr[])
{
    int i, j, cnt, max, pos;

    max = 0;

    for(i = 0; i < SIZE; i++)
    {
        cnt = 0;
        for(j = i+1; j < SIZE; j++)
        {
            if(arr[i] == arr[j])
                cnt++;
        }

        if(cnt > max)
        {
            max = cnt;
            pos = i;
        }
    }

    if(max == 0)
        return NULL;
    else
        return arr+pos;
}
```