

```

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

////////////////////////////////////
////////////////////////////////////
///citirea datelor de la tastatura folosind anumite functii
predefinite

void ex1() {
    char tab[255];
    int zi, luna, an;

    printf("introduceti data:\n");
    gets_s(tab);
    sscanf_s(tab, "%d %d %d", &zi, &luna, &an); //citeste ziua,
luna, anul
    puts(tab); //afiseaza noua valoare a lui tab cu datele
introduse de la tastatura

    char ch[50];
    printf("\nIntroduceti un sir de caractere:\n");
    gets_s(ch); //asteapta un sir de caractere introduse de la
tastatura (necesitand enter la sf) si afiseaza in consola sirul
introdus
    printf("\nCaracterele sunt: %s", ch);

    _getch();
}
////////////////////////////////////
////////////////////////////////////
//calculeaza volumul unui bazin cu datele introduse de la tastatura
//fct main contine numai declaratii de variabile si apelul de
functii

int calcul_volum(int lungime, int latime, int inaltime) {
    int volum = 0;

    volum = lungime*latime*inaltime;

    return volum;
}

void ex2() {
    int latime = 0, lungime = 0, inaltime = 0, r;

    printf("Dati lungimea bazinului: \nL= ");
    scanf_s("%d", &lungime);
    printf("\nDati latimea bazinului:\nI= ");
    scanf_s("%d", &latime);
    printf("\nDati inaltimea bazinului: \nH= ");
    scanf_s("%d", &inaltime);
}

```

```

        r = calcul_volum(lungime, latime, inaltime);
        printf("\nVolumul bazinului este: \nVolum= %d", r);

        _getch();
    }
    ///////////////////////////////////////////////////////////////////
    //o functie care returneaza suma cifrelor unui numar

int suma_cifrelor(int numar) {
    int suma = 0;

    do {
        suma += numar % 10;
        numar = numar / 10;
    } while (numar >= 1);

    return suma;
}

void ex3() {
    int nr = 0;
    int x = 0;

    printf("Introduceti numarul:\nnumar= ");
    scanf_s("%d", &nr);

    x = suma_cifrelor(nr);
    printf("\nSuma cifrelor numarului %d este = %d", nr, x);

    _getch();
}
/////////////////////////////////////////////////////////////////
//calculul lui C^y_x (combinari de c(x,y)) folosind formula

float calcul_combinari(int x, int y) {
    float c = 1.0;

    for (int i = 0; i < y; i++) {
        c = c * ((x - i) / (i + 1));
    }

    return c;
}

void ex4() {
    int x = 0, y = 0;
    float a;

    printf("Introduceti primul numar:\nx= ");
    scanf_s("%d", &x);
    printf("\nIntroduceti cel de-al doilea numar(trebuie sa fie

```

```

mai mic ca x ):\ny= ");
scanf_s("%d", &y);
while(y>=x){
    printf("\nReintroduceti: ");
    scanf_s("%d",&y);
}

a = calcul_combinari(x, y);
printf("\nCalculul lui c(%d,%d) dupa formula, este = %.2f",
x, y, a);

    _getch();
}
////////////////////////////////////
/////
//citeste un intreg pozitiv de tip long, stabileste daca este prim
si afiseaza un mesaj corespunzator

int numar_prim(unsigned long nr) {
    for (int i = 2; i <= (nr / 2); i++) {
        if ((nr%i) == 0)//daca restul impartii la i
incrementat este 0 atunci nr nu este prim
            return 0;
        else
            return 1;
    }
}

void ex5() {
    unsigned long numar = 0;

    printf("Introduceti numarul(sa nu aiba mai mult de 10
cifre): \nnumar= ");
    scanf_s("%d", &numar);

    int x = numar_prim(numar);
    printf("\nNumarul introdus %d este %d unde 1=prim si
0=neprim", numar, x);

    _getch();
}
////////////////////////////////////
/////
////concatenarea a doua siruri

void ex6() {
    char s1[50], s2[50];

    printf("Primul sir: ");
    gets_s(s1);//citeste caractere de la tastatura si se
opreste la apasarea tastei enter
    printf("\nAl doilea sir: ");
    gets_s(s2);
}

```

```

        strcat_s(s1, s2);//la primul sir adauga si valorile celui
de-al doilea
        printf("\nSirurile sunt:\n%s\n", s1);

        _getch();
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///
//maximul dintr-un sir/vector
/*

int v[50];

int citire() {
    int dim;

    printf("Dati dimensiunea vectorului: ");
    scanf_s("%d", &dim);
    printf("\nIntroduceti elementele vectorului:\n");
    for (int i = 0; i < dim; i++) {
        printf("v[%d] = ", i);
        scanf_s("%d", &v[i]);
    }
    printf("\nVectorul introdus este: \nv=[ ");
    for (int i = 0; i < dim; i++)
        printf("%d ", v[i]);
    printf("]");

    return dim;
}

int max_vector(int n) {
    int max=v[0];
    for (int i = 1; i < n; i++){
        if (max < v[i])
            max=v[i];
    }
    return max;
}

void main() {
    int n;
    n=citire();

    printf("\nElementul maxim este: %d",max_vector(n));

    _getch();
}
*/
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//inlocuirea unui element x_i din vector cu un altul
/*

int v[20];

```

```

int citire() {
    int dim;

    printf("Dati dimensiunea vectorului: ");
    scanf_s("%d", &dim);
    printf("\nIntroduceti elementele vectorului:\n");
    for (int i = 0; i < dim; i++) {
        printf("v[%d] = ", i);
        scanf_s("%d", &v[i]);
    }
    printf("\nVectorul introdus este: \nv=[ ");
    for (int i = 0; i < dim; i++)
        printf("%d ", v[i]);
    printf("]");

    return dim;
}

void replace(int dim, int p, int e) {
    int vechi = v[p];
    v[p] = e;
    printf("\nNoul vector este:\nv'=[ ");
    for (int i = 0; i < dim; i++)
        printf("%d ", v[i]);
    printf("]");

    printf("\nElementul de pe pozitia %d era %d si a fost
inlocuit cu elemetul %d", p, vechi, e);
}

void main() {
    int n, poz, elem;
    n = citire();

    printf("\nCe pozitie doriti sa inlocuiti?\n");
    scanf_s("%d", &poz);

    printf("\nCare este numarul pe care il introduceti?\n");
    scanf_s("%d", &elem);

    replace(n,poz,elem);

    _getch();
}
*/
////////////////////////////////////
//suma elementelor diagonalei secundare
/*

int m[20][20];

void citire_matrice(int lc) {

```

```

    int i, j;
    printf("\nIntroduceti elementele matricii:\n");
    for (i = 0; i < lc; i++) {
        for (j = 0; j < lc; j++) {
            printf("a[%d][%d] = ", i, j);
            scanf_s("%d", &m[i][j]);
        }
    }
    printf("\nMatricea introdusa este:\n");
    for (i = 0; i < lc; i++) {
        for (j = 0; j < lc; j++)
            printf("%d ", m[i][j]);
        printf("\n");
    }
}

int suma_diag_secundara(int l) {
    int d=0, i, j=0;

    for (i = l-1; i >= 0; i--)
        d = d + m[i][j++];

    return d;
}

void main() {
    int l, suma;

    printf("Nr de linii si coloane: ");
    scanf_s("%d", &l);
    while (l > 20 || l < 0) {
        printf("\nReintroduceti: ");
        scanf_s("%d", &l);
    }

    citire_matrice(l);
    suma=suma_diag_secundara(l);

    printf("\nSuma elementelor diagonalei secundare este: %d",
suma);

    _getch();
}
*/
///// //////////////////////////////////////
///
//ecuatia de gradul II

int delta(int a, int b, int c) {
    int d;
    d = (int)(pow(b,2)) - 4 * a * c;

    return d;
}

```

```

void solutii(int a, int b, int c) {
    int s1, s2;
    int d=delta(a,b,c);

    if (d < 0) {
        printf("\nEcuatia nu are solutii reale!");
    }else{
        if (d > 0) {
            s1 = (-b + sqrt(d)) / (2 * a);
            printf("\nSolutia x1= %d",s1);
            s2 = (-b - sqrt(d)) / (2 * a);
            printf("\nSolutia x2= %d",s2);
        }else {
            s1 = -b / (2 * a);
            printf("\nEcuatia are o singura solutie
x1=x2= %d",s1);
        }
    }
}

void main() {
    int a, b, c;

    printf("Ecuatia de gradul II cu numere intregi\n");
    printf("\nIntroduceti a= ");
    scanf_s("%d", &a);
    while (a == 0) {
        printf("\nReintroduceti a= ");
        scanf_s("%d", &a);
    }
    printf("Introduceti b= ");
    scanf_s("%d", &b);
    printf("Introduceti c= ");
    scanf_s("%d", &c);

    solutii(a, b, c);

    _getch();
}

/*
void main() {

    //ex1();
    //ex3();
    //ex4();
    //ex5();
    //ex6();

}
*/

```