

Факултет за електротехника и
информациски технологии



решени задачи во
програмски јазик

C

СОДРЖИНА:

1. *Циклуси во C*
2. *Дводимензионални полиња [матрици] во C*
3. *Итератици и рекурзивни функции во C*
4. *Текстуални низи и покажувачи во C*
5. *Датотеки во C*
6. *Структури во C*

ЦИКЛУСИ во С

1. Да се напише програма во која од тастатура се внесува еден број. На екран да се испечатат сите четирицифрени броеви кои го содржат цел број пати внесениот број.

Се бараат четирицифрени броеви кои се деливи на дадениот број

```
#include <stdio.h>
int main ()
{
    int l, x1;
    scanf %x;
    printf ("Vnesi broj x= ");
    scanf ("%d",&x);
    x1=(int)x; /* Se bara cel broj bidejki operacijata % raboti samo za celi broevi */
    for (i=1000;i<=9999;i++)
    {
        If (i%x1==0)
            printf("%d",i);
    }
    return 0;
}
```

2. Да се напише програма која ќе ги изброи сите броеви од даден интервал (почетокот и крајот на интервалот се внесуваат од тастатура) во кои најзначајната цифра е парна и поголема од најмалку значајната цифра.

Пример: 843, 431, 6292

```
#include <stdio.h>
int main ()
{
    int dg, gg, najmnoguZ, najmalkuZ, i, a, cifra;
    printf("Vnesuvanje na intervalot:\n");
    printf("Dolna granica: ");
    scanf ("%d",&dg);
    printf("Gorna granica: ");
    scanf ("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
```

```

        a=i;
        while (a)
        {
            cifra=a%10;
            a/=10;
        }
        najmnoguZ=cifra;
        najmalkuZ=i%10;
        if ((najmnoguZ%2==0) && (najmnoguZ>najmalkuZ))
            printf("%d\n",i);
    }
    return 0;
}

```

3. Да се напише програма која од непознат број на цели броеви кои се внесуваат преку тастатура ќе ја испечати позицијата и вредноста на минималниот елемент.

Пример: 8 4 6 8 10 5 7 3 5 7 6 4 11. Позицијата на мин. елемент е 8 а неговата вредност е 3

```

#include <stdio.h>
int main ()
{
    int poz, min, broj, k=1;
    printf("Za kraj da se vnese znak!\n");
    scanf("%d\n",&broj);
    poz=1;
    min=broj;
    while (scanf("%d\n",&broj))
    {
        k++;
        if (broj<min)
        {
            min=broj;
            poz=k;
        }
    }
    printf("Miniumum e %d i se naogja na pozicija %d\n", min, poz);
    return 0;
}

```

4. Да се напише програма која од непознат број на цели броеви кои се внесуваат преку тастатура ќе ја испечати вредноста и позицијата на максималниот елемент.

Пример: 2 4 6 8 4 5 7 2 5 7 6 4 2. Позицијата на мак. елемент е 4 а неговата позиција е 8

```

#include <stdio.h>
int main ()
{
    int poz,max,broj,k=1;
    printf("Za kraj da se vnese znak!\n");    scanf("%d\n",&broj);
    poz=1;
    max=broj;
    while (scanf("%d\n",&broj))
    {
        k++;

```

```

        if (broj>max)
        {
            max=broj;
            poz=k;
        }
    }
    printf("Maksimum e %d i se naogja na pozicija %d\n",max,poz);
    return 0;
}

```

5. Да се напише програма која од непознат број на цели броеви кои се внесуваат преку тастатура ќе ги испише максималниот и минималниот елемент.

Пример: 2 42 4 16 48 96 101 202. Ќе ги испечати броевите 2 и 202

```

#include <stdio.h>
int main ()
{
    int max,broj,min;
    printf("Za kraj da se vnese znak!\n");    scanf("%d\n",&broj);
    min=broj; max=broj;
    while (scanf("%d\n",&broj))
    {
        if (broj<min) min=broj;
        if (broj>max) max=broj;
    }
    printf("Minimum e %d, maksimum e %d\n", min, max);
    return 0;
}

```

6. Да се напише програма која ќе ги испечати сите броеви од даден интервал (почетокот и крајот на интервалот се внесуваат од тастатура) чија сума на парните цифри е делива со 3.

```

#include <stdio.h>
int main ()
{
    int dg,gg,i,a,suma,cifra;
    printf("Vnesuvanje na intervalot:\n");
    printf("Dolna granica: ");    scanf ("%d",&dg);
    printf("Gorna granica: ");    scanf ("%d",&gg);
    for (i=dg; i<=gg; i++)
    {
        a=i; suma=0;
        while (a)
        {
            cifra=a%10;
            a/=10;
            if (cifra%2==0) suma+=cifra;
        }
        if ((suma) && (suma%3==0)) printf("%d\n",i);
    }
    return 0;
}

```

7. Да се напише програма во која од тастатура се внесува еден број. Ако внесениот број е двоцифрен да се испечати првиот број поголем од него делив со 3, а ако бројот е троцифрен да се испечати првиот број помал од него делив со 4.

Пример: За 14 ќе испечати 15, а за 243 ќе испечати 240

```
#include <stdio.h>
int main ()
{
    int n,i;
    printf("n= ");
    scanf("%d",&n);
    if ((n>=10) && (n<=99))
    {
        for (i=n+1; i<=99; i++)
        {
            if (i%3==0)
            {
                printf ("Prviot broj pogolem od %d i deliv so 3 e %d\n",n,i);
                break;
            }
        }
    }
    else if ((n>=100) && (n<=999))
    {
        for (i=n-1; i>=100; i--)
        {
            if (i%4==0)
            {
                printf ("Prviot broj pomal od %d i deliv so 4 e %d\n",n,i);
                break;
            }
        }
    }
    else printf ("Ne e vnesen dvocifren ili trocifren broj!\n");
    return 0;
}
```

8. Да се напише програма која ќе ги изброи сите броеви од даден интервал (почетокот и крајот на интервалот се внесуваат од тастатура) во кои најзначајната цифра е помала од најмалку значајната цифра.

```
#include <stdio.h>
int main ()
{
    int dg,gg,i,a,suma,cifra,k=0;
    printf("Vnesuvanje na intervalot:\n");
    printf("Dolna granica: ");    scanf ("%d",&dg);
    printf("Gorna granica: ");    scanf ("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        a=i;
        while (a)
        {
```

```

        cifra=a%10;
        a/=10;
    }
    if (cifra < i%10) k++;
}
printf("Postojat vkupno %d takvi cifri",k);
return 0;
}

```

9. Да се напише програма во која од тастатура се внесува еден број. За внесениот број да се ипечатат сите негови делители.

```

#include <stdio.h>
int main ()
{
    int broj,i,k;
    printf("Vnesi eden broj: ");
    scanf("%d",&broj);
    k=0;
    for (i=1;i<=broj;i++)
    {
        if (!(broj%i))
        {
            k++;
            printf("%d\n",i);
        }
    }
    if (!k) printf ("Brojot nema deliteli, t.e. toj e prost!\n");
    else printf ("Postojat %d deliteli\n",k);
    return 0;
}

```

10. Да се напише програма во која од тастатура се внесува еден број. Ако внесениот број е парен да се испечати првиот број поголем од него делив со 7, а ако бројот е непарен да се испечати првиот број помал од него делив со 9.

```

#include <stdio.h>
int main ()
{
    int n,i;
    printf("Vnesi broj n= ");
    scanf("%d",&n);
    if (n%2==0)
    {
        for (i=n+1;i>0;i++)
        {
            if (i%7==0)
            {
                printf("%d",i);
                break;
            }
        }
    }
}

```

```

else
{
    for (i=n-1;i<n;i--)
    {
        if (i%9==0)
        if (n<9)
        {
            printf("Ne postoi takov broj\n");
            break;
        }
        else
        {
            printf("%d",i);
            break;
        }
    }
}
return 0;
}

```

11. Да се напише програма која ќе ги испечати сите броеви од даден интервал (почетокот и крајот на интервалот се внесуваат од тастатура) чија сума на непарните цифри е непарен број.

```

#include <stdio.h>
int main ()
{
    int dg,gg,i,a,suma,cifra,k=0;
    printf("Vnesuvanje na intervalot:\n");
    printf("Dolna granica: ");
    scanf ("%d",&dg);
    printf("Gorna granica: ");
    scanf("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        a=i;
        suma=0;
        while (a)
        {
            cifra=a%10;
            a/=10;
            if (cifra%2) suma+=cifra;
        }
        if (suma%2)
        {
            printf("%d\n",i);
            k++;
        }
    }
    printf("Postojat %d takvi broevi vo dadeniot interval.\n",k);
    return 0;
}

```


12. Да се напише програма која од непознат број на знаци кои се внесуваат преку тастатура ќе ги испечати знаците со најголем и најмал ASCII код.

Пример: *a n r s A B M F b c. Ќе испечати A како знак со најмал ASCII код и s како знак со најголем ASCII код*

```
#include <stdio.h>
int main ()
{
    char c, max, min;
    printf("Za kraj da se vnese izvicnik (!)\n");
    scanf("%c",&c);
    min=c;
    max=c;
    while (scanf("%c",&c))
    {
        if (c=='!') break;
        if (c<min) min=c;
        if (c>max) max=c;
    }
    printf("%c e znak so najgolem ASCII kod, a ",max);
    printf("%c e znak so najmal ASCII kod.\n",min);
    return 0;
}
```

13. Да се напише програма која ќе ги испечати сите цели броеви од зададен опсег (почетокот и крајот на опсегот се внесуваат од тастатура) на кои збирот на непарните цифри е поголем од збирот на парните цифри.

Пример: *за бројот 2534, $5+3 > 2+4$. Броевите кои содржат само парни или непарни цифри нема да се печатат*

```
#include <stdio.h>
int main ()
{
    int dg, gg, i, a, cifra, x=0;
    int suma_neparni, suma_parni, parni, neparni;
    printf("Vnesuvanje na intervalot\n");
    printf("Dolna granica: ");
    scanf("%d",&dg);
    printf("Gorna granica: ");
    scanf("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        a=i;
        suma_parni=0;
        suma_neparni=0;
        parni=0;
        neparni=0;
        while (a)
        {
            cifra=a%10;
            a/=10;
            if (!(cifra%2))
            {
                if (!cifra) break;
                else
```

```

        {
            suma_parni+=cifra;
            parni++;
        }
    }
    if (cifra%2)
    {
        suma_neparni+=cifra;
        neparni++;
    }
}
if ((parni) && (neparni) && (suma_neparni>suma_parni))
{
    printf("%d\n",i);
    x++;
}
}
printf("Ima %d broevi koi go ispolnuvat uslovot.\n",x);
return 0;
}

```

14. Да се напише програма во која од тастатура се внесуваат вредности за страните на триаголник. Програмата треба да провери дали триаголникот е правоаголен. Внесените вредности не се подредени по големина.

```

#include <stdio.h>
int main ()
{
    int a,b,c,a2,b2,c2;
    printf("Vnesi gi tranite:\n");
    printf("a= "); scanf("%d",&a);
    printf("b= "); scanf("%d",&b);
    printf("c= "); scanf("%d",&c);
    a2=a*a;
    b2=b*b;
    c2=c*c;
    if ((a+b<c) || (a+c<b) || (b+c<a))
        printf("Ne moze da se konstuiara triagolnik!\n");
    else
    {
        if(a2==(b2+c2) || b2==(a2+c2) || c2==(a2+b2))
            printf ("Triagolnikot e pravoagolen.\n");
        else printf("Triagolnikot ne e pravoagolen.\n");
    }
    return 0;
}

```

15. Да се напише програма која ќе ги испечати сите трицифрени броеви и нивните делители.

Излез: 100: 1, 2, 4, 5, 10, 20, 25, 50, 100
 101: 1, 101
 102: 1, 2, 3, 6, 17, 34, 51, 102

 999: 1, 3, 9, 27, 37, 111, 333, 999

```
#include <stdio.h>
int main ()
{
    int i,j;
    for (i=100; i<=999; i++)
    {
        printf("%d: ",i);
        for (j=1; j<=i;j++)
        {
            if (!(i%j))
                printf("%d, ",j);
        }
        printf("\n");
    }
    return 0;
}
```

16. Да се напише програма која ќе ги испечати сите двоцифрени броеви и нивните делители.

Излез: 10: 1, 2, 5, 10
 11: 1, 11
 12: 1, 2, 3, 4, 6, 12

 99: 1, 3, 9, 11, 33, 99

```
#include <stdio.h>
int main ()
{
    int i, j;
    for (i=10; i<=99; i++)
    {
        printf("%d: ",i);
        for (j=1; j<=i;j++)
        {
            if (!(i%j))
                printf("%d, ",j);
        }
        printf("\n");
    }
    return 0;
}
```

17. Да се напише програма која ќе ги испечати сите цели броеви од зададен опсег (почетокот и крајот на опсегот се внесуваат од тастатура) кои имаат ист збир на непарните и парните цифри.

Пример: за бројот 2514, $2+4 = 5+1$. Броевите кои содржат само парни или непарни цифри нема да се печатат

```
#include <stdio.h>
int main ()
{
    int dg, gg, i, a, cifra, x=0;
    int suma_neparni, suma_parni, parni, neparni;
    printf("Vnesuvanje na intervalot\n");
    printf("Dolna granica: ");
    scanf("%d",&dg);
    printf("Gorna granica: ");
    scanf("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        a=i;
        suma_parni=0;
        suma_neparni=0;
        parni=0;
        neparni=0;
        while (a)
        {
            cifra=a%10;
            a/=10;
            if (!(cifra%2))
            {
                if (!cifra) break;
                else
                {
                    suma_parni+=cifra;
                    parni++;
                }
            }
            if (cifra%2)
            {
                suma_neparni+=cifra;
                neparni++;
            }
        }
        if ((parni) && (neparni) && (suma_neparni == suma_parni))
        {
            printf("%d\n",i);
            x++;
        }
    }
    printf("Ima %d broevi koi go ispolnuvat uslovot.\n",x);
    return 0;
}
```

18. Од тастатура се внесуваат непознат број цели броеви. Да се напише програма што ќе ги изброи и прикаже сите броеви кои се деливи со производот на нивната најзначајна цифра и најмалку значајна цифра.

```
#include <stdio.h>
int main ()
{
    int broj, a, cifra, najZ, najmalkuZ, x=0, q;
    printf("Za kraj vnesi znak!\n");
    while(scanf("%d",&broj))
    {
        a=broj;
        while (a)
        {
            cifra=a%10;
            a/=10;
        }
        najZ=cifra;
        najmalkuZ=broj%10;
        q=najZ*najmalkuZ;
        if((broj%q==0) && (q))
        {
            printf("%d e takov broj\n",broj);
            x++;
        }
    }
    if (x==1) printf ("1 broj go zadovoluva uslovot.\n");
    else printf("%d broevi go zadovoluvaat uslovot.\n",x);
    return 0;
}
```

19. Од тастатура се внесуваат непознат број цели броеви. Да се напише програма што ќе ги изброи и прикаже сите броеви кои се деливи со разликата на нивната најзначајна цифра и најмалку значајна цифра.

```
#include <stdio.h>
int main ()
{
    int broj,a,cifra,najZ,najmalkuZ,x=0,q;
    printf("Za kaj vnesi znak!\n");
    while(scanf("%d",&broj))
    {
        a=broj;
        while (a)
        {
            cifra=a%10;
            a/=10;
        }
        najZ=cifra;
        najmalkuZ=broj%10;
        q=najZ-najmalkuZ;
        if((q) && (broj%q==0))
        {
```

```

        printf("%d e takov broj\n",broj);
        x++;
    }
}
if (x==1) printf ("1 broj go zadovoluva uslovot.\n");
else printf("%d broevi go zadovoluvaat uslovot.\n",x);
return 0;
}

```

20. Да се напише програма што ќе ја претвори температурата од Фаренхајтови степени во Целзиусови и резултатите ќе ги прикаже табеларно. Температурата се движи од 0° до 300° Фаренхајтови со чекор од 20° степени. Формулата за претварање е следната: $^{\circ}\text{C} = 5/9 (^{\circ}\text{F} - 32)$. Потоа Целзиусовите степени да се прикажат со прецизност до 1 децимала. Задачата да се реши со употреба на for структурата.

```

0      -17.8
20     -6.7
40      4.4
60     15.6
80     26.7
100    37.8
итн.....

```

```

#include <stdio.h>
int main ()
{
    int Farenhajt, i;
    float Celzius;
    for (Farenhajt=0;Farenhajt<=300;Farenhajt+=20)
    {
        printf("%d\t",Farenhajt);
        Celzius=5/9.0*(Farenhajt-32);
        printf("%.1f\n", Celzius);
    }
    return 0;
}

```

21. Да се напише програма која ќе ги отпечати апсолутните вредности само на парните броеви од зададен опсег на цели броеви (опсегот се внесува од тастатура). Задачата да се реши само со помош на библиотеката stdio.h, односно забрането е користење на функциите abs() или fabs().

```

#include <stdio.h>
int main ()
{
    int dg, gg, i, a, cifra, parni;
    printf("Vnesuvanje na intervalot\n");
    printf("Dolna granica: ");
    scanf("%d",&dg);
    printf("Gorna granica: ");
    scanf("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        a=i;
        cifra=a%10;
    }
}

```

```

        if((i) && !(cifra%2))
        {
            if(i<0) printf("%d\n",-i);
            else printf("%d\n",i);
        }
    }
    return 0;
}

```

22. Да се напише програма што ќе ја претвара температурата од степени Целзиусови во Фаренхајтови и резултатите ќе ги прикаже табеларно. Температурата се движи во границите од -100° до 200° Целзиусови со чекор од 20° степени. Формулата за претворање е следна: $^{\circ}\text{F} = 32 + 9/5 \cdot ^{\circ}\text{C}$. Притоа Фаренхајтовите степени да се прикажат со прецизност од 1 децимала. Задачата да се реши со употреба на for структура.

```

-100    -148.0
-80     -112.0
-60     -76.0
-40     -40.0
-20     -4.0
0       32.0
итн.....

```

```

#include <stdio.h>
int main ()
{
    int Celzsius, i;
    float Farenhajt;
    for (Celzsius=-100;Celzsius<=200;Celzsius+=20)
    {
        printf("%d\t",Celzsius);
        Farenhajt=32+9.0/5*Celzsius;
        printf("%.1f\n", Farenhajt);
    }
    return 0;
}

```

23. Да се напише програма која ќе ги отпечати апсолутните вредности само на непарните броеви од зададен опсег на цели броеви (опсегот се внесува од тастатура). Задачата да се реши само со помош на библиотеката stdio.h, односно забрането е користење на функциите abs() или fabs().

```

#include <stdio.h>
int main ()
{
    int dg, gg, i, a, cifra, parni;
    printf("Vnesuvanje na intervalot\n");
    printf("Dolna granica: ");
    scanf("%d",&dg);
    printf("Gorna granica: ");
    scanf("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        a=i;
        cifra=a%10;
    }
}

```

```

        if((i) && (cifra%2))
        {
            if(i<0) printf("%d\n",-i);
            else printf("%d\n",i);
        }
    }
    return 0;
}

```

24. Да се напише програма во која од тастатура се внесуваат вредности за страните на два триаголника и да се провери дали триаголниците се складни, или различни. Внесените вредности се подредени по големина.

Пример: $a=4, b=5, c=6$

```

#include <stdio.h>
int main ()
{
    int a, b, c, a_1, b_1, c_1;
    printf("Vrednosti za I triagolnik:\n");
    printf("a= "); scanf("%d",&a);
    printf("b= "); scanf("%d",&b);
    printf("c= "); scanf("%d",&c);

    printf("Vrednosti za II triagolnik:\n");
    printf("a_1= "); scanf("%d",&a_1);
    printf("b_2= "); scanf("%d",&b_1);
    printf("c_3= "); scanf("%d",&c_1);

    if (((a+b)<c || (a+c)<b || (b+c)<a) && ((a_1+b_1)<c_1 || (a_1+c_1)<b_1 || (b_1+c_1)<a_1))
        printf("Ne moze da se konstruiirat triagolnici.\n");
    else if ((a==a_1) && (b==b_1) && (c==c_1))
        printf("Trigolnicite SE skladni.\n");
    else
        printf ("Triagolnicite NE SE skladni.\n");
    return 0;
}

```

25. Од тастатура се внесуваат координатите на две кружници и нивните радиуси. Да се напише програма во која ќе се провери дали кружниците меѓусебно се сечат.

координатите на точките кои ги внесуваме ќе бидат координти на центарот на кружниците, па од нив потоа ќе бараме централно растојание меѓу кружниците кое е еднакво на растојанието меѓу двете точки. Условот кој се

бара е: $r_1 - r_2 < \overline{O_1O_2} < r_1 + r_2$ и $|r_1| > [r_2]$.

```

#include <stdio.h>
#include <math.h>
int main ()
{
    int x1,x2,y1,y2,r1,r2;
    float O1O2;
    printf ("Dimenzii na prvata kruznica:\n");
    printf("x1= "); scanf("%d",&x1);
    printf("y1= "); scanf("%d",&y1);
    printf("r1= "); scanf("%d",&r1);
    printf ("Dimenzii na vtorata kruznica:\n");

```



```
printf("x2= "); scanf("%d",&x2);
printf("y2= "); scanf("%d",&y2);
printf("r2= "); scanf("%d",&r2);

/*Centralno rastojanie megju kruznicite*/
O1O2=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));

if ((r1-r2) < O1O2 && O1O2 < (r1+r2) && abs(r1) > abs(r2)) printf("Kruznicite se secat.\n");
else printf("Kruznicite ne se secat.\n");
return 0;
}
```

26. Од тастатура се внесуваат координатите на две кружници и нивните радиуси. Да се напише програма во која ќе се провери дали кружниците меѓусебно се допират.

Условот кој е бара е: $\overline{O_1O_2} = r_1 + r_2$ или $\overline{O_1O_2} = r_1 - r_2$.

```
#include <stdio.h>
#include <math.h>
int main ()
{
    int x1,x2,y1,y2,r1,r2;
    float O1O2;
    printf ("Dimenzii na prvata kruznicata>\n");
    printf("x1= "); scanf("%d",&x1);
    printf("y1= "); scanf("%d",&y1);
    printf("r1= "); scanf("%d",&r1);
    printf ("Dimenzii na vtorata kruznicata>\n");
    printf("x2= "); scanf("%d",&x2);
    printf("y2= "); scanf("%d",&y2);
    printf("r2= "); scanf("%d",&r2);

    /*Centralno rastojanie megju kruznicite*/
    O1O2=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));

    if ((O1O2) && (O1O2==(r1-r2) || O1O2==(r1+r2))) printf ("Kruznicite se dopiraat.\n");
    else printf ("Kruznicite ne se dopiraat.\n");
    return 0;
}
```

27. Од тастатура се внесуваат координатите на две кружници и нивните радиуси. Да се напише програма во која ќе се провери дали едната кружница ја опфаќа другата и ако ја опфаќа дали тие се концентрични.

Условот кој се бара е: $\overline{O_1O_2} < r_1 - r_2$, а потоа $O_1(x_1, y_1) = O_2(x_2, y_2)$

```
#include <stdio.h>
#include <math.h>
int main ()
{
    int x1,x2,y1,y2,r1,r2;
    float O1O2;
    printf ("Dimenzii na prvata kruznicata>\n");
    printf("x1= "); scanf("%d",&x1);
    printf("y1= "); scanf("%d",&y1);
```

```
printf("r1= "); scanf("%d",&r1);
printf ("Dimenzii na vtorata kruznicica>\n");
printf("x2= "); scanf("%d",&x2);
printf("y2= "); scanf("%d",&y2);
printf("r2= "); scanf("%d",&r2);

/* Centralno rastojanie megju kruznicite */
O1O2=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));

if (O1O2 < (r1-r2))
{
    printf ("Kruznicite se postaveni edna vo druga");
    if ((x1==x2) && (y1==y2)) printf (" i se koncentricni.\n");
    else printf (" , no ne se koncentricni.\n");
}
else printf ("Kruznicite ne se postaveni edna vo druga.\n");
return 0;
}
```

28. Да се напише програма кој ќе ги отпечати сите четирицифрени броеви за кои важи следното равенство:

$$\overline{abcd} = a^2 + b^2 + c^2 + d^2$$

На крај програмата треба да прикаже колку четирицифрени броеви го задоволуваат равенството. Во програмата, дозволема е употреба само на for структура како механизам за повторување на низа наредби.

Напомена: Програмата ќе врати **Nema takvi broevi vo intervalot.**

```
#include <stdio.h>
int main ()
{
    int abcd, a, b, c, d, zbir, x=0;
    for (abcd=1000;abcd<=9999;abcd++)
    {
        d=abcd%10;
        c=(abcd/10)%10;
        b=(abcd/100)%10;
        a=(abcd/1000);
        zbir=a*a+b*b+c*c+d*d;
        if (abcd==zbir)
        {
            printf("%d\n",abcd);
            x++;
        }
    }
    if (!x) printf ("Nema takvi broevi vo intervalot.\n");
    return 0;
}
```

29. Да се напише програма што ќе го пресмета следниот производ:

$$proizvod = \prod_{i=1}^n \frac{i!}{m}$$

Вредностите на m и n се внесуваат преку тастатура.

```
#include <stdio.h>
int main ()
{
    int m, n, faktoriel=1, i;
    float proizvod=1.0;
    printf("n= "); scanf("%d",&n);
    printf("m= "); scanf("%d",&m);
    for (i=1; i<=n; i++)
    {
        faktoriel*=i;
        proizvod*=(1.0)*faktoriel/m;
    }
    printf("Proizvod= %.3f.\n", proizvod);
    return 0;
}
```

30. Да се напише програма во кој од тастатура се внесуваат вредностите за страните на еден триаголник и да се провери дали триаголникот е рамностран, рамнокрак или разностран. Внесените вредности не се подредени по голмина.

```
#include <stdio.h>
int main ()
{
    int a,b,c;
    printf("Vnesi gi stranite: \n");
    printf("a= "); scanf("%d",&a);
    printf("b= "); scanf("%d",&b);
    printf("c= "); scanf("%d",&c);

    if ((a+b)<c || (a+c)<b || (b+c)<a)
        printf("Ne moze da se konstruira triagolnik.\n");
    else
    {
        if ((a==b) && (b==c) && (c==a))    printf ("Triagolnikot e ramnostran.\n");
        else if ((a==b) || (b==c) || (c==a))  printf("Triagolnikot e ramnokrak.\n");
        else    printf("Triagolnikot e raznostran.\n");
    }
    return 0;
}
```

31. Од тастатура се внесуваат непознат број цели броеви. Да се напише програма што ќе ги изброи и прикаже сите броеви чија најзначајна цифра е парна.

```
#include <stdio.h>
int main ()
{
    int a, broj, cifra, x=0;
    while (scanf("%d",&broj))
    {
        a=broj;
        while (a)
        {
            cifra=a%10;
            a/=10;
        }
        if (!(cifra%2))
        {
            printf("%d\n",broj);
            x++;
        }
    }
    printf("%d broevi go zadovoluvaaat uslovot.\n",x);
    return 0;
}
```

32. Да се напише програма што ќе ја пресмета следната сума:

$$Suma = \sum_{i=1}^n i^m$$

Вредностите за m и n се внесуваат од тастатура.

```
#include <stdio.h>
int main ()
{
    int i, j, m, n, eksponent, suma=0;
    printf("n= "); scanf("%d",&n);
    printf("m= "); scanf("%d",&m);
    for (i=1;i<=n;i++)
    {
        eksponent=1;
        for (j=1;j<=m;j++)
            eksponent*=j;
        suma+=eksponent;
    }
    printf("Suma= %d.\n",suma);
    return 0;
}
```

33. Од тастатура се внесуват непознат број на цели броеви. Да се напише програма што ќе ги изброи и прикаже сите броеви чија најзначајна цифра е 3.

```
#include <stdio.h>
int main ()
{
    int broj, pom, cifra, x=0;
    while (scanf("%d",&broj))
    {
        pom=broj;
        while (pom)
        {
            cifra=pom%10;
            pom/=10;
        }
        if (cifra==3)
        {
            printf("%d\n",broj);
            x++;
        }
    }
    printf("%d brovi go zadovoluvaat uslovot.\n",x);
    return 0;
}
```

34. Напиши програма што на комјутерски екран ќе ја прикаже таблицата на множење за сите тројки броеви x, y и z, каде броевите x, y и z се помеѓу 1 и 15.

```
1 * 1 * 1 = 1
1 * 1 * 2 = 2
.....
1 * 1 * 15 = 15
1 * 2 * 1 = 2
.....
1 * 15 * 15 = 225
2 * 1 * 1 = 2
.....
```

```
#include <stdio.h>
int main ()
{
    int x,y,z;
    for (x=1;x<=5;x++)
    for (y=1;y<=5;y++)
    for (z=1;z<=5;z++)
    printf("%d * %d * %d = %d\n",x,y,z,x*y*z);
    return 0;
}
```

35. Од тастатура се внесуваат N броеви. Да се напише програма што ќе ги изброи и прикаже сите броеви што се деливи со 3. Бројот N се внесува од тастатура.

```
#include <stdio.h>
int main ()
{
    int N, broj, i, x=0;
    printf("N= "); scanf("%d",&N);
    printf("Vnesuvaj broevi...\n");
    for (i=1; i<=N; i++)
    {
        scanf("%d",&broj);
        if (!(broj%3))
        {
            printf("%d e deliv so 3\n", broj);
            x++;
        }
    }
    printf("%d broevi go zadovoluvaa uslovot.\n", x);
    return 0;
}
```

36. Да се напише програма која за дадено n ја пресметува и печати вредноста на изразот:

$$\sum_{i=1}^n i!$$

```
#include <stdio.h>
int main ()
{
    int n, i, suma=0, faktoriel=1;
    printf("n= "); scanf("%d",&n);
    for (i=1; i<=n; i++)
    {
        faktoriel*=i;
        suma+=faktoriel;
    }
    printf("Sumata e %d", suma);
    return 0;
}
```

37. Од тастатура треба да се внесат вредностите за страните на еден триаголник, потоа да се провери дали триаголникот може да биде правоаголен и да се отпечати хипотенузата. Внесените вредности се подредени по големина.

```
#include <stdio.h>
int main ()
{
    int a, b, c;
    printf("Vnesi gi stranite: \n");
    printf("a= "); scanf("%d",&a);
    printf("b= "); scanf("%d",&b);
    printf("c= "); scanf("%d",&c);
}
```

```

if (((a) && (b) && (c)) && ((a+b)>c || (a+c)>b || (b+c)>a))
printf("Trigolnikot moze da bide pravoagolen so hipotenuza %d.",c);
else printf("Ne moze da se konstruira triagolnik.");
return 0;
}

```

38. Да се напише програма која на екран ќе испечати шаховска табла со ширина на полињата од четири места , при тоа црните полиња ќе останат непотполнети додека белите полиња ќе се потполнат со нивните соодветни индекси како што е прикажано на сликата. Задачата да се реши со користење на for структура.

A8	C8	E8	G8
B7	D7	F7	H7
A6	C6	E6	G6
B5	D5	F5	H5
A4	C4	E4	G4
B3	D3	F3	H3
A2	C2	E2	G2
B1	D1	F1	H1

```

#include <stdio.h>
int main ()
{
    int i;
    char j;
    for (i=8;i>=1;i--)
    {
        if (!(i%2))
        {
            for (j=65; j<72; j+=2)
                printf(" %c%d ", j, i);
        }
        else
        {
            for (j=66; j<=72; j+=2)
                printf(" %c%d ", j, i);
        }
        printf("\n");
        printf("\n");
    }
    return 0;
}

```

39. Да се определи максимум и минимум меѓу три броја внесени преку тастатура.

```
#include <stdio.h>
int main ()
{
    int a,b,c,max,min;
    printf ("a= "); scanf ("%d",&a);
    printf ("b= "); scanf ("%d",&b);
    printf ("c= "); scanf ("%d",&c);

    max=a;
    if (b>max) max=b;
    if (c>max) max=c;
    printf("Maksimum e %d\n",max);

    min=a;
    if (b<min) min=b;
    if (c<min) min=c;
    printf("Minimum e %d\n",min);
    return 0;
}
```

40. Да се напише програма која за даден цел број внесен од тастатура ќе изврши ротација за едно место надесно. Бројот добиен по ротацијата да се испечати на екран.

Пример: за бројот 2514, ќе го испечати бројот 4251

```
#include <stdio.h>
int main ()
{
    int a,broj, mnozitel=1,cifra;
    scanf ("%d",&broj);
    a=broj;
    while (a)
    {
        cifra=a%10;
        a/=10;
        mnozitel*=10;
    }
    printf ("%d", (broj%10)*(mnozitel/10) + broj/10);
    return 0;
}
```

41. Од тастатура се внесуваат x и y координати на непознат број на точки. Да се напише програма која ќе го испечати вкупниот број на точки кои лежат во вториот квадрант.

```
#include <stdio.h>
int main ()
{
    int x, y, k=0;
    printf ("Za kaj vnesi izvicnik (!)\n");
    printf ("Vnesuvaj koordinati... \n");
    while (scanf ("%d",&x) && scanf ("%d",&y))
    {
```



```

        if ((x<0) && (y>0) && ((x) || (y)))
        {
            printf("(%d,%d)\n",x,y);
            k++;
        }
    }
    printf("%d tocki go zadovoluvaat uslovot.\n",k);
    return 0;
}

```

42. Напиши програма што на комјутерски екран ќе ја прикаже таблицата на собирање за сите тројки броеви x, y и z каде броевите x, y и z се меѓу -10 и 10. Изгледот на екранот е:

```

-10 + -10 + -10 = -30
-10 + -10 + -9 = -29
.....
-10 + -10 + 10 = -10
10 + -9 + -10 = -29
.....
-10 + 10 + 10 = 10
-9 + -10 + -10 = -29
.....

```

```

#include <stdio.h>
int main ()
{
    int x, y, z;
    for (x=-10; x<=10; x++)
    {
        for (y=-10; y<=10; y++)
        {
            for (z=-10; z<=10; z++)
                printf("%d + %d + %d = %d\n",x,y,z,x+y+z);
        }
    }
    return 0;
}

```

43. Од тастатура се внесуваат непознат број цели броеви. Да се напише програма што ќе ги изброи и прикаже сите броеви чија најзначајна цифра е парна.

```

#include <stdio.h>
int main ()
{
    int x=0, cifra, a, broj;
    while (scanf("%d",&broj))
    {
        a=broj;
        while (a)
        {
            cifra=a%10;
            a/=10;
        }
        if (!(cifra%2))
        {

```

```

        printf("%d e takov broj\n",broj);
        x++;
    }
}
printf("%d broevi go ispolnuvaat uslovot.\n",x);
return 0;
}

```

44. Да се напише програма која ќе ги испечати сите парни броеви помали од N (N се внесува од тастатура) кои се истовремено деливи со збирот на нивните непарни цифри и со производот на нивните парни цифри. На крај да се испечати колку ткви брови се пронајдени.

```

#include <stdio.h>
int main ()
{
    int N, i, cifra, a, z, p, x=0;
    printf("N= "); scanf("%d",&N);
    for (i=2; i<N; i+=2)
    {
        a=i;
        p=1;
        z=0;
        while (a)
        {
            cifra=a%10;
            a/=10;
            if (!(cifra%2)) p*=cifra;
            else z+=cifra;
        }
        if ((z) && (p) && (!(i%z)) && (!(i%p)))
        {
            printf("%d\n",i);
            x++;
        }
    }
    printf("%d broevi go zadovoluvaat uslovot.\n",x);
    return 0;
}

```

45. Од тастатура се внесуваат непознат број цели брови. Да се напише програма што ќе го пронајде и испечати најголемиот број како и колку пати се појавува.

```

#include <stdio.h>
int main ()
{
    int poz,max,broj,x,k=1;
    printf("Za kraj da se vnese znak!\n");
    scanf("%d\n",&broj);
    poz=1;
    max=broj;
    while (scanf("%d\n",&broj))
    {
        k++;
    }
}

```

```

        if (broj>max)
        {
            x=0;
            max=broj;
            poz=k;
        }
        if (broj==max)
            x++;
    }
    printf("Maksimum e %d na pozicija %d, a se povtoruva %d pati\n",max,poz,x);
    return 0;
}

```

46. Од тастатура се внесуваат непознат број цели брови. Да се напише програма што ќе го пронајде и испечати најголемиот број и најмалиот број како и колку пати се појавуваат.

```

#include <stdio.h>
int main ()
{
    int pmax,pmin,max,min,broj,x,k=1,d;
    printf("Za kraj da se vnese znak!\n");
    scanf("%d\n",&broj);
    pmax=1;    max=broj;
    pmin=1;    min=broj;
    while (scanf("%d\n",&broj))
    {
        k++;
        if (broj>max)
        {
            x=0;
            max=broj;
            pmax=k;
        }
        if (broj<min)
        {
            d=0;
            min=broj;
            pmin=k;
        }
        if (broj==max) x++;
        if (broj==min) d++;
    }
    printf("Maksimum e %d na pozicija %d, a se povtoruva %d pati\n",max,pmax,x);
    printf("Minimum e %d na pozicija %d, a se povtoruva %d pati\n",min,pmin,d);
    return 0;
}

```

47. Да се напише програма која од непознат број на знаци кои се внесуваат преку тастатура ќе ги табелира знаците, нивните ASCII кодови, хексадецималната и окталната претстава од нивните ASCII кодови.

```
#include <stdio.h>
int main ()
{
    char znak;
    printf("Za kraj vnesi izvicnik (!)\n");
    while (scanf("%c",&znak))
    {
        if (znak=='!') break;
        printf ("Znak: %c\tASCII: %d\tXo: %x\ttO: %o\n",znak,znak,znak,znak);
    }
    return 0;
}
```

48. Да се напише програма која ќе ги отпечати сите броеви од даден опсег (почетокот и крајот на опсегот се внесуваат од тастатура) кои имаат иста прва и последна цифра. Потоа да се испечати и бројот на броевите кои го исполнуваат условот.

Пример: 5,... 22,... 565,... 9869,...

```
#include <stdio.h>
int main ()
{
    int i,dg,gg,a,cifra,x=0;
    printf ("Vnesi go opsegot: \n");
    printf ("Dolna granica: "); scanf("%d",&dg);
    printf ("Gorna granica: "); scanf("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        a=i;
        while (a)
        {
            cifra=a%10;
            a/=10;
        }
        if (cifra==(i%10))
        {
            printf("%d\n",i);
            x++;
        }
    }
    printf("%d broevi go zadovoluvaaat uslovot.\n");
    return 0;
}
```

49. Да се напише програма која помага во правењето на распределба на еден студент по просторија за полагање на колоквиум. На почетокот за студентот преку тастатура се внесува бројот на индекс (само редниот број, а годината не се внесува, на пример 234/2009, се внесува 234). Ако редниот број на индексот е поголем од 900, се печати дека студентот полага во просториите на ФЛАОП. Останатите студенти се распоредуваат во простории според остатокот на делење на нивниот реден број со 5. Студентите со остаток 0 при делење на нивниот реден број со 5 полагаат во Амфитеатар, остаток 1 во просторијата 223, остаток 2 во просторијата 211, остаток 3 во просторијата 110, остаток 4 во Барака 3. За внесениот индекс од студент, програмата треба да отпечати:

student so indeks XXX polaga vo prostorijata ZZZZ.

```
#include <stdio.h>
int main ()
{
    int index;
    printf ("Vnesi go redniot broj na indeks: ");
    scanf ("%d",&index);
    if (index > 900)
        printf ("Studentot so indeks %d polaga vo prostoriite na FLAOP.\n",index);
    else
    {
        if (!(index%5))      printf ("Studentot so indeks %d polaga vo Amfiteatar.\n",index);
        else if (index%5==1) printf ("Studentot so indeks %d polaga vo prostorija 223.\n",index);
        else if (index%5==2) printf ("Studentot so indeks %d polaga vo prostorija 211.\n",index);
        else if (index%5==3) printf ("Studentot so indeks %d polaga vo prostorija 110.\n",index);
        else                 printf ("Studentot so indeks %d polaga vo Barak 3.\n",index);
    }
    return 0;
}
```

50. За да се избегне гужва при вадењето на биометриски пасош потребно е да се закаже термин за сликање. Да се напише програма која помага во правењето на распределба на еден граѓанин по просторија за сликање. Како податок, за граѓанинот преку тастатура се внесуваат последните шест цифри од матичниот број (од xx xx xxx XXXXXX, се внесува XXXXXX). Ако внесениот број почнува со 45 тогаш да се продолжи со распределбата, во спротивно на екран да прикаже Nema potreba od raspredelba. Ако распределбата продолжи, граѓаните се распределуваат во простории според остатокот на делење на нивниот шестцифрен број со 4. Граѓаните со остаток 0 при делење на нивниот шестцифрен број со 4 ќе се сликаат во просторија 'Cetiri', остаток 1 во просторијата 'Eden', остаток 2 во просторијата 'Dva', остаток 3 во просторијата 'Tri'. За внесениот шестцифрен број од матичниот број на еден граѓанин, програмата треба да отпечати:

Gragjaninot so posledni cifri od maticen broj XXXXXX se slika vo prostorij YYYY.

I начин:

```
#include <stdio.h>
int main ()
{
    int maticen;
    printf ("Vnesi pi poslednite 6 cifri od maticniot broj: ");
    scanf ("%d",&maticen);
    if (maticen/10000!=45)
        printf ("Nema potreba od raspredelba.\n");
}
```

```

else
{
    if (!(maticen%4))
    {
        printf("Gragjaninot so posledni cifri od maticniot broj");
        printf(" %d se slika vo prostorija cetiri.\n",maticen);
    }
    else if (maticen%4==1)
    {
        printf("Gragjaninot so posledni cifri od maticniot broj");
        printf(" %d se slika vo prostorija eden.\n",maticen);
    }
    else if (maticen%4==2)
    {
        printf("Gragjaninot so posledni cifri od maticniot broj");
        printf(" %d se slika vo prostorija dva.\n",maticen);
    }
    else
    {
        printf("Gragjaninot so posledni cifri od maticniot broj");
        printf(" %d se slika vo prostorija tri.\n",maticen);
    }
}
return 0;
}

```

II начин:

```

#include <stdio.h>
int main ()
{
    int maticen;
    printf("Vnesi pi poslednite 6 cifri od maticniot broj: ");
    scanf ("%d",&maticen);
    if (maticen/10000!=45)
        printf("Nema potreba od raspredelba.\n");
    else
    {
        switch (maticen%4)
        {
            case 1:
            {
                printf ("Gragjaninot so posledni cifri od maticniot broj");
                printf(" %d se slika vo prostorija eden.\n",maticen);
                break;
            }
            case 2:
            {
                printf ("Gragjaninot so posledni cifri od maticniot broj");
                printf(" %d se slika vo prostorija dva.\n",maticen);
                break;
            }
            case 3:
            {
                printf ("Gragjaninot so posledni cifri od maticniot broj");

```

```

        printf(" %d se slika vo prostorija tri.\n",maticen);
        break;
    }
    case 0:
    {
        printf ("Gragjaninot so posledni cifri od maticniot broj");
        printf(" %d se slika vo prostorija cetiri.\n",maticen);
        break;
    }
}
return 0;
}

```

51. Да се напише програма која за два внесени цели броја ќе провери дали сумата на нивните цифри се еднаква.

```

#include <stdio.h>
int main ()
{
    int broj1, broj2, cifra1, cifra2;
    int suma1=0, suma2=0, a, b;
    printf("Vnesi gi dvata broja: \n");
    printf("a= "); scanf("%d",&broj1);
    printf("b= "); scanf("%d",&broj2);

    a=broj1;
    while (a)
    {
        cifra1=a%10;
        a/=10;
        suma1+=cifra1;
    }

    b=broj2;
    while (b)
    {
        cifra2=b%10;
        b/=10;
        suma2+=cifra2;
    }

    if (suma1==suma2)
        printf("Broevite %d i %d go zadovoluvat uslovot.\n",broj1,broj2);
    else
        printf("Broevite %d i %d ne go zadovoluvat uslovot.\n",broj1,broj2);
    return 0;
}

```

52. Да се напише програма која ги печати простите броеви од даден интервал (границите на интервалот се внесуваат преку тастатура) и нивниот број.

```
#include <stdio.h>
void main ()
{
    int dg, gg, i, j, x, k=0;
    printf ("Vnesuvanje na granicite na intervalot> \n");
    printf ("Dolna granica: "); scanf ("%d",&dg);
    printf ("Gorna granica: "); scanf ("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        x=0;
        for (j=2;j<=i;j++)
            if (!(i%j))
                x++;
        if (x<=2)
        {
            printf("%d\n",i);
            k++;
        }
    }
    printf("%d broevi se prosti.\n",k);
}
```

53. Да се напише програма која како влезен аргумент прима цел број а потоа проверува дали производот на цифрите на тој број е прост број.

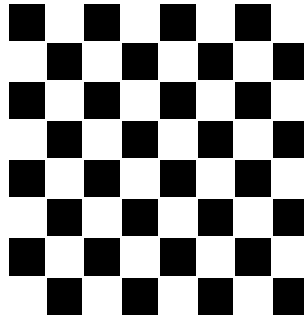
```
#include <stdio.h>
void main ()
{
    int broj, cifra, p=1, i, x=0, pom;
    printf ("Vnesi cel broj: ");
    scanf ("%d",&broj);
    pom=broj;
    while (pom)
    {
        cifra=pom%10;
        pom/=10;
        if (!cifra)
            break;
        else
            p*=cifra;
    }
    for (i=1; i<=p; i++)
    {
        if (!(p%2))
            x++;
    }
    if (x<=2)
        printf("%d go ispolnuv uslovot.\n",broj);
    else
        printf("%d ne go ispolnuv uslovot.\n",broj);
}
```


54. Да се напише програма која ќе ги печати сите прости броеви од даден интервал (почетокот и крајот на интервалот се внесуваат од тастатура), кај кои производот на последните 2 цифри е исто така прост број (за едноцифрени броеви се смета само една цифра).

```
#include <stdio.h>
int main ()
{
    int dg, gg, i, q, j, k, z, r, p, dvecifri;
    printf ("Opreduvanje na intervalot:\n");
    printf ("Dolna granica: "); scanf ("%d",&dg);
    printf ("Gorna granica: "); scanf ("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        k=0;
        r=0;
        p=0;
        if (i<10)
        {
            for (j=1;j<=i;j++)
            {
                if (!(i%j)) k++;
            }
            if (k<=2) printf("%d\n",i);
        }

        if (i>=10)
        {
            for (q=1;q<=i;q++)
            {
                if (!(i%q)) p++;
            }
            if (p<=2)
            {
                dvecifri=i%100;
                for (z=1; z<=dvecifri; z++)
                {
                    if (!(dvecifri%z)) r++;
                }
                if (r<=2) printf("%d\n",i);
            }
        }
    }
    return 0;
}
```

55. Да се напише програма која ќе печати шаховска табла на која ќе бидат обележани само црните полиња како на сликата:



```
#include <stdio.h>
void main ()
{
    int i, j;
    for (i=1;i<=8;i++)
    {
        if (i%2)
        {
            for (j=1;j<=4;j++) printf("[ " );
        }
        else
        {
            for (j=1;j<=4;j++) printf(" []");
        }
        printf("\n");
    }
}
```

56. Да се напише програма која на компјутерскиот екран ја испечати следнава слика:



```
#include <stdio.h>
void main ()
{
    int i, j;
    for (i=1;i<=4;i++)
    {
        for (j=1; j<=i; j++) printf("* ");
        printf("\n");
    }
    for (i=1;i<=3;i++)
    {
        for (j=3; j>=i; j--) printf("* ");
        printf("\n");
    }
}
```

57. Да се напише програма која на компјутерскиот екран ќе ја испечати следнава слика:

```
*
 *
  *
   *
```

```
#include <stdio.h>
void main ()
{
    int i,j;
    for (i=1;i<=4;i++)
    {
        for (j=1;j<=4;j++)
        {
            if (i==j) printf("*");
            else      printf(" ");
        }
        printf("\n");
        printf("\n");
    }
}
```

58. Да се напише програма која ќе печати квадратна матрица од произволен ред и општи членови.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

```
#include <stdio.h>
void main ()
{
    int i, j, red;
    printf("Red na matrica: "); scanf("%d",&red);
    printf("%dX%d MATRICA: \n",red,red);
    for (i=1;i<=red;i++)
    {
        for (j=1;j<=red; j++)
        printf("a%d%d ",i,j);
        printf("\n");
    }
}
```

59. Од тастатура се внесуваат непознат број на координати на точки во рамнина. Да се напише програма која ќе ги испечати координатите на точката од првиот квадрант што е најблиску до координатниот почеток.

```
#include <stdio.h>
int main ()
{
    int x,y,minx,miny,z=1;
    printf("Za kaj vnesi izvicnik (!)\n");
    printf("Vnesuvaj koordinati... \n");
    scanf("%d",&x); scanf("%d",&y);
```

```

if (!(x>0) && (y>0))
printf("Tockata ne e od prv kvadrant.\n");
else
{
    minx=x;
    miny=y;
    while (scanf("%d",&x) && scanf("%d",&y))
    {
        if ((x>0) && (y>0))
        {
            if ((x<minx) && (y<miny))
            {
                minx=x;
                miny=y;
            }
        }
        else
        printf("Tockata ne e od prv kvadrant.\n");
    }
    printf("(%d,%d)\n",minx,miny);
}
return 0;
}

```

60. Да се напише програма која ќе ги изброи отпечатените броеви од даден опсег (опсегот се внесува преку тастатура) кои имаат парна почетна и крајна цифра.

```

#include <stdio.h>
void main ()
{
    int dg, gg, i, pom, cifra, k=0;
    printf ("Opredeluvnje na intervalot:\n");
    printf ("Dolna granica: "); scanf ("%d",&dg);
    printf ("Gorna granica: "); scanf ("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        pom=i;
        while (pom)
        {
            cifra=pom%10;
            pom/=10;
        }
        if (!(cifra%2) && !((i%10)%2) && (i%10))
        {
            printf("%d\n",i);
            k++;
        }
    }
    printf("%d brovi go zadovoluvat uslovot.\n",k);
}

```

61. Да се напише програма која ќе прочита од тастатура непознат број на координатни точки и ќе испечати колку од точките се наоѓаат на растојание поголемо или еднакво на 5 од координатниот почеток.

```
#include <stdio.h>
#include <math.h>
int main ()
{
    int x, y, i;
    float d;
    printf("Za kraj da se vnese znak!\n");
    printf("Vnesi koordinati (x,y)...\n");
    while (scanf("%d %d",&x, &y))
    {
        d=sqrt(x*x+y*y);
        if (d>=5) printf("(%d,%d)\n",x,y);
        else      printf("Uslovot ne e ispolnet.\n");
    }
    return 0;
}
```

62. Да се најде НЗД на два броја.

```
#include <stdio.h>
int main()
{
    int n1,n2;
    printf("Vnesi dva broja:"); scanf("%d %d",&n1,&n2);
    while(n1!=n2)
    {
        if(n1>n2) n1=n1-n2;
        else      n2=n2-n1;
    }
    printf("NZD=%d",n1);
    return 0;
}
```

63. Да се најде НЗС на два броја.

```
#include <stdio.h>
int main()
{
    int a, b;
    printf("Vnesi dva broja:"); scanf("%d %d",&a,&b);
    while(a!=b)
    {
        if(a>b) a=a-b;
        else    b=b-a;
    }
    printf("NZS=%d",(n1*n2)/a);
    return 0;
}
```

64. Од тастатура се внесуваат неопределен број на цели броеви. Да се напише програма која ќе ги испечати броевите чија сума на цифри е прост број.

```
#include <stdio.h>
int main ()
{
    int broj, suma, a, cifra, i, k;
    while (scanf("%d",&broj))
    {
        suma=0;
        k=0;
        a=broj;
        while (a)
        {
            cifra=a%10;
            a/=10;
            suma+=cifra;
        }
        for (i=1;i<=suma;i++)
        {
            if (!(suma%i))
                k++;
        }
        if (k<=2)
            printf("%d e takov broj\n",broj);
    }
    return 0;
}
```

65. Да се напише програма која за дадена вредност на целиот број x (вредноста на x се задава преку тастатура) ќе ја пресмета вредноста на:

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

со собирање на првите 10 членови.

```
#include <stdio.h>
int main ()
{
    int x, broi, znak, i;
    float clen, konecno;
    printf("Vnesi vrednost za x= ");
    scanf("%d",&x);
    konecno=1; clen=1; znak=1;
    for (i=1;i<=10;i++)
    {
        broi=2*i;
        znak=(-1)*znak;
        clen=clen*x*x/(broi*(broi-1));
        konecno=konecno+znak*clen;
    }
    printf("Vrednosta na izrazot e %.5f.\n",konecno);
    return 0;
}
```

66. Од тастатура се внесуваат координатите на N отсечки. Да се напише програма која на компјутерскиот екран ќе ја прикаже должината на најдолгата отсечка.

```
#include <stdio.h>
#include <math.h>
int main ()
{
    int N, x1, x2, y1, y2, i;
    double d, dmax=-5.0;
    printf("N= "); scanf("%d",&N);
    for (i=1;i<=N;i++)
    {
        printf("x1%d= ",i); scanf("%d",&x1);
        printf("x2%d= ",i); scanf("%d",&x2);
        printf("y1%d= ",i); scanf("%d",&y1);
        printf("y2%d= ",i); scanf("%d",&y2);

        /* Opredeluvanje na rastojanieto */
        d=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));

        if (d>dmax) dmax=d;
        printf("\n");
    }
    printf("%.3f\n",dmax);
    return 0;
}
```

67. Да се напише програма која ќе ги испечати сите цели броеви од зададен опсег и нивните трансформирани претстави. Почетокот и крајот на опсегот се внесуваат од тастатура. Со трансформацијата се добива број на кој цифрите му се превртени и помеѓу секоја од цифрите се вметнува цифрата 7.

Пример: за бројот 20, ќе го испечати бројот 20 и 72
 за бројот 100, ќе го испечати бројот 100 и 7071
 за бројот 236 ќе го испечати бројот 236 и 67372

```
#include <stdio.h>
int main()
{
    int a, b, pom, obraten, i, cifra;
    printf ("Opredeluvnje na intervalot:\n");
    printf ("Dolna granica: "); scanf ("%d",&a);
    printf ("Gorna granica: "); scanf ("%d",&b);
    for (i=a;i<=b;i++)
    {
        pom=i;
        obraten=0;
        while (pom)
        {
            cifra=pom%10;
            obraten=obraten*10+cifra;
            if (pom>9)
            {
                obraten=obraten*10+7;
            }
        }
    }
}
```

```

    }
    pom/=10;
}
printf("%d --> %d\n",i,obraten);
}
return 0;
}

```

68. Да се испечати обратниот број на број кој се внесува од тастатура.

```

#include <stdio.h>
int main()
{
    int broj, prevrten, cifra, pom;
    printf("Vnesi broj a= ");
    scanf("%d",&broj);
    pom=broj;
    prevrten=0;
    while (pom)
    {
        cifra=pom%10;
        pom/=10;
        prevrten=prevrten*10+cifra;
    }
    printf("%d",prevrten);
    return 0;
}

```

69. Да се напише програма која во даден опсег од цели броеви ќе ги испечати обратните броеви на дадените и тоа составени само од непарните цифри со вметната цифра 5 меѓу секоја од цифрите.

Пример: 123456 ---> 55351

```

#include <stdio.h>
int main()
{
    int broj, prevrten, cifra, pom;
    printf("Vnesi broj a= ");
    scanf("%d",&broj);
    pom=broj;
    prevrten=0;
    while (pom)
    {
        cifra=pom%10;
        if (cifra%2)
        {
            prevrten=prevrten*10+cifra;
            if (pom>9) prevrten=prevrten*10+5;
        }
        pom/=10;
    }
    printf("%d",prevrten);
    return 0;
}

```


70. Да се напише програма која за даден број внесен од тастатура ќе го најде неговиот обратен број и тоа составен само од цифрите кои се на парните позиции во дадениот број со цифра 5 меѓу нив. Позициите се бројат од 1.

Пример: 987643 ---> 45759

```
#include <stdio.h>
int main()
{
    int broj, prevrten, cifra, pom,i=0;
    printf("Vnesi broj a= ");
    scanf("%d",&broj);
    pom=broj;
    prevrten=0;
    while (pom)
    {
        cifra=pom%10;
        i++;
        if (!(i%2))
        {
            prevrten=prevrten*10+cifra;
            if (pom>9)
            {
                prevrten=prevrten*10+5;
            }
        }
        pom/=10;
    }
    printf("%d",prevrten);
    return 0;
}
```

71. Да се напише програма која од целите броеви на даден интервал ќе ги испечати целите броеви сотовени само од непарните цифри на дадените броеви.

```
#include <stdio.h>
int main ()
{
    int dg, gg, i, p, pom, cifra, nbr;
    printf ("Vnesi gi granicite na intervalot:\n");
    printf ("Dolna granica: "); scanf ("%d",&dg);
    printf ("Gorna granica: "); scanf ("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        pom=i; nbr=0; p=1;
        while (pom)
        {
            cifra=pom%10;
            pom/=10;
            if ((cifra%2) && (cifra))
            {
                nbr=nbr+p*cifra;
                p*=10;
            }
        }
    }
}
```

```
        printf("%d\n",nbr);
    }
    return 0;
}
```

72. Да се напише програма која за даден број ќе го испечати истиот тој број.

```
#include <stdio.h>
int main ()
{
    int broj, cifra, broj1, proizvod, a;
    printf("a= "); scanf("%d",&broj);
    a=broj; broj1=0; proizvod=1;
    while (a)
    {
        cifra=a%10;
        a/=10;
        broj1=broj1+proizvod*cifra;
        proizvod*=10;
    }
    printf("%d i %d se isti",broj, broj1);
    return 0;
}
```

73. Да се напише програма која од даден интервал (границите на интервалот се внесуваат преку тастатура) ќе ги испечати само броевите кои се читаат исто од лево на десно и од десно на лево.

```
#include <stdio.h>
int main ()
{
    int dg, gg, i, x=0, a, cifra;
    printf ("Vnesi gi granicite na intervalot:\n");
    printf ("Dolna granica: "); scanf ("%d",&dg);
    printf ("Gorna granica: "); scanf ("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        a=i;
        int nbr=0;
        while (a)
        {
            cifra=a%10;
            a/=10;
            nbr=nbr*10+cifra;
        }
        if ((nbr==i) && (i>9))
        {
            printf("%d = %d\n",i,nbr);
            x++;
        }
    }
    printf("%d broevi go zadovoluvaat uslovot.\n",x);
    return 0;
}
```

74. Да се напише програма во која од тастатура се внесуваат непознат број на броеви. Од нив да се прикажат на екран сите петтоцифрени броеви кои го исполнуват условот: производот на цифрите од непарните позиции да е број делив со 4. Позициите се бројат од десно во лево и најдесната цифра е на позиција 0. Да е испечати колку броеви го задоволуваат условот.

Пример: 26327 $\rightarrow 2 \cdot 6 = 12$, 12 е делив со 4, 26327 го исполнува условот.
 27334 $\rightarrow 3 \cdot 7 = 21$, 21 не е делив со 4, 27334 не го исполнува условот.

I начин:

```
#include <stdio.h>
int main ()
{
    int broj, poz, p, pom, a, x=0;
    while (scanf("%d",&broj))
    {
        pom=broj;
        poz=0;
        p=1;
        while (pom)
        {
            if ((poz%2) && (poz))  p*=pom%10;
            {
                poz++;
                pom/=10;
            }
            if ((poz==5) && !(p%4))
            {
                printf("%d",broj);
                x++;
            }
        }
    }
    printf("%d broevi go adovoluvat uslovot.\n",x);
    return 0;
}
```

II начин:

```
#include <stdio.h>
int main ()
{
    int broj, pom, a, b, c, d, e,x=0;
    while (scanf("%d",&broj))
    {
        if (broj < 10000)
        {
            printf("Obidete se povtorno!\n");
            continue;
        }
        else
        {
            pom=broj;
            a=pom%10; pom/=10;
            b=pom%10; pom/=10;
```

```

        c=pom%10; pom/=10;
        d=pom%10; pom/=10;
        e=pom%10;
        if (!((b*d)%4) && (broj<99999))
        {
            printf("%d\n",broj);
            x++;
        }
        else
        {
            printf("Obidete e povtorno!\n");
            continue;
        }
    }
    printf("%d broevi go zadovoluvaat uslovot.\n",x);
    return 0;
}

```

75. Да се напише програма во која од тастатура се внесува непознат број на броеви. Од нив да се прикажат на екран сите петтоцифрени броеви кои го исполнуват условот: збирот на цифрите од непарните позиции да е број делив со 3. Позициите се бројат од лево на десно и најлевата цифра е на позиција 0.

Пример: 26327 → 6+2=8, 8 не е делив со 3, 26327 не го исполнува условот.

27324 → 7+2=9, 9 е делив со 3, 27324 го исполнува условот.

I начин:

```

#include <stdio.h>
int main ()
{
    int broj, poz, p, pom, a, x=0;
    while (scanf("%d",&broj))
    {
        pom=broj;
        poz=4;
        p=0;
        while (pom)
        {
            if ((poz%2) && (poz)) p+=pom%10;
            {
                poz--;
                pom/=10;
            }
            if ((poz==0) && !(p%3))
            {
                printf("%d",broj);
                x++;
            }
        }
    }
    printf("%d broevi go zadovoluvat uslovot.\n",x);
    return 0;
}

```

}

II начин:

```

#include <stdio.h>
int main ()
{
    int broj, pom, a, b, c, d, e, x=0;
    while (scanf("%d",&broj))
    {
        if (broj < 10000)
        {
            printf("Obidete se povtorno!\n");
            continue;
        }
        else
        {
            pom=broj;
            a=pom%10; pom/=10;
            b=pom%10; pom/=10;
            c=pom%10; pom/=10;
            d=pom%10; pom/=10;
            e=pom%10;
            if (!((b+d)%3) && (broj<99999))
            {
                printf("%d\n",broj);
                x++;
            }
            else
            {
                printf("Obidete e povtorno!\n");
                continue;
            }
        }
    }
    printf("%d broevi go zadovoluvaaat uslovot.\n",x);
    return 0;
}

```

76. Некој текст се внесува од тастатура знак по знак се додека не се внесе нов ред. Да се напише програма која на екран ќе ја прикаже позицијата на првата појава на буквата V, како и вкупниот број на појави на буквата V во текстот.

```

#include <stdio.h>
void main ()
{
    int poz_V=-100, vk_V=0, br_znaci=0;
    char znak;
    while (scanf("%c",&znak) && znak!='\n')
    {
        br_znaci++;
        if (znak=='V' && poz_V==-100) /*poz_V za da se najde prvata pojava na V */
            poz_V=br_znaci;
        if (znak=='V')

```

```

        vk_V++;
    }
    if (!(vk_V)) printf("V ne se javuva vo tekstot.\n");
    else        printf("Prvata pojav na V e na pozicija %d, ");
    printf(" a se pojavuva vkupno %d pati",poz_V,vk_V);
}

```

77. Некој текст се внесува од тастатура знак по знак се додека не се внесе нов ред. Да се напише програма која на екран ќе ја прикаже позицијата на последната појава на буквата W, како и вкупниот број на букви различни од W во текстот.

```

#include <stdio.h>
void main ()
{
    int br_znaci=0, poz_W=0, raz_W=0;
    char znak;
    while (scanf("%c",&znak) && znak!='\n')
    {
        br_znaci++;
        if (znak=='W')
            poz_W=br_znaci;
        if (znak!='W')
            raz_W++;
    }
    if (!(poz_W)) printf("W ne se pojavuva vo tekstot.\n");
    else          printf("Poslednata pozicija na W e %d, a",poz_W);
    printf(" se внесени vkupno %d znaci razlicni od W",raz_W);
}

```

78. Некој текст се внесува од тастатура знак по знак се додека не се внесе нов ред. Да се напише програма која ќе печати колку реченици биле внесени. Можноста за постоење на '??', '...', '!!!', '!' се занемарува.

```

#include <stdio.h>
void main ()
{
    int rec=0;
    char znak;
    while(scanf("%c",&znak) && znak!='\n')
    {
        if ((znak=='.' ) || (znak=='?') || (znak=='!'))
            rec++;
    }
    if (rec==0) printf("Ne stavivte znak za kraj na recenica.\n");
    else        printf ("Vnesovte %d recenici.\n",rec);
}

```

79. Некој текст се внесува од тастатура знак по знак се додека не се внесе нов ред. Да се напише програма која ќе печати колку согласки се внесени.

```
#include <stdio.h>
void main ()
{
    int rec=0;
    char znak;
    while(scanf("%c",&znak) && znak!='\n')
    {
        if (!((znak=='.') || (znak=='?') || (znak=='!') || (znak==' ') || (znak==',')))
        {
            if (!((znak=='1') || (znak=='2') || (znak=='3') || (znak=='4') || (znak=='5')))
            {
                if (!((znak=='6') || (znak=='7') || (znak=='8') || (znak=='9')))
                {
                    if (!((znak=='a') || (znak=='e') || (znak=='i') || (znak=='o') || (znak=='u')))
                        rec++;
                }
            }
        }
    }
    if (rec==0)
        printf("Ne se внесени soglaski.\n");
    else
        printf ("Vnesovte %d soglaski.\n",rec);
}
```

80. Да се напише програма која ќе ги испечати сите парни броеви помали од n (n се внесува преку тастатура) кои се истовремено деливи со збирот на нивните цифри и производот на нивните непарни цифри. На крај да го испечати бројот на пронајдени такви броеви.

Пример: 10, 12, 18, ..., 36, ..., 108, ..., 6312, ...

```
#include <stdio.h>
int main ()
{
    int n, z, p, a, cifra, i, k=0, q;
    printf("n= "); scanf("%d",&n);
    for (i=1;i<=n;i++)
    {
        if (!(i%2))
        {
            a=i;
            z=0;
            p=1;
            q=0;
            while (a)
            {
                cifra=a%10;
                a/=10;
                z+=cifra;
                if ((cifra%2) && (cifra))
                {

```

```

                p*=cifra;
                q++;
            }
        }
        if ((i%z==0) && (i%p==0) && (q) && (z))
        {
            printf("%d\n",i);
            k++;
        }
    }
}
if (!(k)) printf("Ne postoi takov broj vo intervalot.\n");
else     printf ("%d broevi go zadovoluvaat intervalot.\n",k);
}

```

81. Напишете програма која ќе ги отпечати сите броеви во даден опсег кои имаат иста прва и последна цифра. На крајот да се испечати бројот на броевите кои го исполнуват условот.

```

#include <stdio.h>
int main ()
{
    int dg, gg, i, pom, cifra, x=0;
    printf("Granici na intervalot:\n");
    printf("Dolna granica: "); scanf("%d",&dg);
    printf("Gorna granica: "); scanf("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        pom=i;
        while (pom)
        {
            cifra=pom%10;
            pom/=10;
        }
        if (cifra==i%10)
        {
            printf("%d\n",i);
            x++;
        }
    }
    if (!(x)) printf("Nema takvi broevi vo intervalot.\n");
    else     printf("%d broevi go zadovoluvaat uslovot.\n",x);
    return 0;
}

```


82. Од тастатура се внесуваат непознат број на цели броеви. Да се напише програма кој за секој внесен број ќе формира нов, така што секој цифра X во бројот ќе се замени со цифра Y, а цифрата Z ќе се испушти. Цифрите X, Y и Z се внесуваат од тастатура и се исти а сите броеви. Програмата на кај печати колку броеви биле внесени.

Пример: X=1 Y=3 Z=9
 21941 → 2343

```
#include <stdio.h>
int main ()
{
    int x, y, z, p, cifra, pom, broj, pr, vk=0;
    printf("x= "); scanf("%d",&x);
    printf("y= "); scanf("%d",&y);
    printf("z= "); scanf("%d",&z);
    while (scanf("%d",&broj))
    {
        vk++;
        pom=broj;
        p=0;
        pr=1;
        while (pom)
        {
            cifra=pom%10;
            pom/=10;
            if (cifra!=z)
            {
                if (cifra==x) cifra=y;
                p=p+pr*cifra;
                pr*=10;
            }
        }
        printf("%d\n",p);
    }
    printf("Vneseni se %d broevi.\n",x);
    return 0;
}
```

83. Да се напише програма за калкулатор што пресметува аритметички израз, на пример: $5-3+5*2/3$. Изразот се внесува преку тастатура, броевите се цели броеви помеѓу 0 и 9, операциите се множење (*), целобројно делење(/), собирање (+), одземање(-). Не постои преоритет на операциите, што значи горенаведениот израз ќе се евалуира како: $5-3=2+5=7*2=14/3=4$, при што резултатот ќе го испечати на екран.

```
#include <stdio.h>
int main ()
{
    int broj, resenie=0;
    char znak='+';
    do
    {
        scanf("%d",&broj);
        switch (znak)
        {
```

```

        case '+':
        {
            resenie+=broj;
            break;
        }
        case '-':
        {
            resenie-=broj;
            break;
        }
        case '*':
        {
            resenie*=broj;
            break;
        }
        case '/':
        {
            resenie/=broj;
            break;
        }
    }
}
while (scanf("%c", &znak) && znak!='=');
printf("Resenieto e %d\n",resenie);
return 0;
}

```

84. Да се напише програма која за внесен број од тастатура ќе ги испечати сите негови делители.

```

#include <stdio.h>
void main ()
{
    int broj, x=0, i;
    printf("a= "); scanf("%d",&broj);
    printf("%d:\t",broj);
    for (i=1;i<=broj;i++)
    {
        if (!(broj%i))
        {
            printf("%d, ",i);
            x++;
        }
    }
    printf("\n%d ima %d deliteli.\n",broj, x);
}

```

85. Да се напише програма која ќе ги испечати сите броеви од даден интервал чија сума на парните цифрите е делива со 3.

```
#include <stdio.h>
int main ()
{
    int x=0, dg, gg, pom, cifra, suma, i;
    printf("Granici na intervalot:\n");
    printf("Dolna granica: "); scanf("%d",&dg);
    printf("Gorna granica: "); scanf("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        pom=i; suma=0;
        while (pom)
        {
            cifra=pom%10;
            pom/=10;
            if (!(cifra%2)) suma+=cifra;
        }
        if ((suma) && !(suma%3))
        {
            printf("%d\n",i);
            x++;
        }
    }
    printf("%d broja go adovoluvaat uslovot.\n",x);
    return 0;
}
```

86. Да се напише програма која ќе ја печати аритметичката средина на првите 20 прости броеви.

```
#include <stdio.h>
int main ()
{
    int i, j, k, prost=0, suma=0;
    for (i=2;i>0;i++)
    {
        k=0;
        for (j=1;j<=i;j++)
        {
            if (!(i%j)) k++;
        }
        if (k<=2)
        {
            prost++;
            printf("%d\n",i);
            suma+=i;
        }
        if (prost==20) break;
    }
    printf("Art. sredina e %.3f\n",(float)suma/20);
    return 0;
}
```

87. Да се напише програма што ќе го најде најмалиот позитивен број што е истовремено делив со сите цифри од 1 до 9. Да се испечати бројот и неговиот спротивен број.

```
#include <stdio.h>
int main()
{
    int i, j, k, rot, mnozitel, cifra, pom;
    for (i=1; i>0;i++)
    {
        k=0;
        for (j=1;j<=9;j++)
        {
            if (!(i%j)) k++;
        }
        if (k==9) break;
    }
    pom=i; rot=0; mnozitel=1;
    while (pom)
    {
        cifra=pom%10;
        pom/=10;
        rot=rot*10+cifra;
    }
    printf("%d ---> %d", i, rot);
    return 0;
}
```

88. Телекомуникациската компанија FEEIT – Phones располага со продажни центри во кои се нудат различни услуги на корисниците. На термините кои се наоѓаат на влез на секој продажен центар, корисниците треба да изберат меѓу услугите: „Плаќање сметки“ со внесување на буквата p, или „Дополнителни услуги“ – со внесување на буквата d. Дополнително, треба да се внесе буквата l ако станува збор за физичко лице или буквата f ако станува збор за фирма. Да се напише програма во која прво се внесува буква за избор на услуга (Плаќање сметки или дополнителни услуги). Потоа се притиска ентер. На крај се внесува буква за тоа кој ја бара услугата (физичко лице или фирма), по што повторно се притиска ентер. Програмата проверува каков вид на услуга се бара и кој ја бара услугата и во зависност од тоа се печати следната порака:

- „Salter 1“ – ако се бара плаќање на сметки од страна на физичко лице.
- „Salter 2“ – ако се бара плаќање на сметки од страна на фирма
- „Salter 3“ – ако се бараат дополнителни услуги од страна на физичко лице
- „Salter 4“ – ако се бараат дополнителни услуги од страна на фирма

Ако не се внесе валидна буква се печати порака „Невалиден избор“.

```
#include <stdio.h>
int main ()
{
    char c1, c2, pr1, pr2;
    printf("Dobredojdovte na terminalot na FEEIT-Phones.\n");
    printf("Uslugite koi Vi gi nudime se slednite> \n");
    printf("1. Plakjanje smetki ----- pritisnete <p>\n");
    printf("2. Drugi ulugi ----- pritisnete <d>\n");
    scanf("%c",&c1);
    scanf("%c",&pr1);
```

```

if(pr1=='\n')
{
if (c1=='p' || c1=='d')
{
printf("Molime da go vnesete podnesitelot na branjeto.\n");
printf("Za <Fizicko lice> pritisnete <l>.\n");
printf("Za <Firma> pritisnete <f>.\n");
scanf("%c",&c2);
scanf("%c",&pr2);
if(pr2=='\n')
{
if (c2=='l' || c2=='f')
{
if (c1=='p' && c2=='l') printf("Shalter 1.\n");
else
{
if (c1=='p' && c2=='f') printf("Shalter 2.\n");
else
{
if (c1=='d' && c2=='l') printf("Shalter 3.\n");
else printf("Shalter 4.\n");
}
}
}
}
else printf("Nevaliden izbor.\n");
}
}
else printf("Nevaliden izbor.\n");
}
return 0;
}

```

89. Да се напише програма кој за секој број од даден опсег (почетокот и крајот н опсегот се внесуваат од тастатура) проверува дали бројот напишан со истите цифри, но во обратен редослед е еднаков на збирот на кубовите на цифрите. Ако условот е исполнет, бројот се печати. На крај треба да се испечати и вкупниот број на брови во опсегот кои го задоволуваат условот.

Пример: *Оригинален број:* 173 \rightarrow 371 = $3^3 + 7^3 + 1^3$
 Оригинален број: 351 \rightarrow 153 = $1^3 + 5^3 + 3^3$

```

#include <stdio.h>
void main ()
{
    int x=0, i, dg, gg, suma, prevrten, pom, cifra;
    printf("Vnesi granica: \n");
    printf("Dolna granica: "); scanf("%d",&dg);
    printf("Gorna granica: "); scanf("%d",&gg);
    for (i=dg;i<=gg;i++)
    {
        pom=i;
        prevrten=0;
        suma=0;
        while (pom)
        {
            cifra=pom%10;

```

```

        pom/=10;
        suma+=cifra*cifra*cifra;
        prevrten=prevrten*10+cifra;
    }
    if (suma==prevrten)
    {
        printf("%d\n",i);
        x++;
    }
}
printf("%d brovi go zadovoluvat uslovot.\n",x);
}

```

90. Да се напише програма која го пресметува следниот израз:

$$p = \prod_{i=1}^n \left(\sum_{j=1}^i \frac{i}{2j+1} \right)$$

каде вредноста на n се внесува преку тастатура.

```

#include <stdio.h>
int main ()
{
    int i, j, n;
    float clen, zbir=0.0, proizvod=1.0;
    printf("n= "); scanf("%d",&n);
    for (i=1;i<=n;i++)
    {
        clen=0.0;
        for (j=1;j<=i;j++)
        {
            clen=(float)i/(2*j+1);
            zbir+=clen;
        }
        proizvod*=zbir;
    }
    printf("p=%f\n",proizvod);
    return 0;
}

```

91. Од тастатура се внесуваат непознат број на броеви. Да се напише програма која проверува дали сумата на секој тековно внесен број и предходниот број е парен број. Ако условот е исполнет да се испечатат двата броја (тековниот и предходниот). На крај програмат да го испечати вкупниот број на парови кои го исполнуваат условот како и најголемата сума на два последователни броја што била најдена и го задоволува бараниот услов.

```

#include <stdio.h>
void main ()
{
    int x=0, k=0, suma, broj, predhoden;
    while (scanf("%d",&broj))
    {
        if (x)

```

```
{
    suma=predhoden+broj;
    if (!(suma%2))
    {
        printf("%d -- %d\n", predhoden, broj);
        k++;
    }
    predhoden=broj;
    x++;
}
printf("%d parovi go zadovoluvat uslovot.\n",k);
}
```

II

ДВОДИМЕНЗИОНАЛНИ ПОЛИЊА [МАТРИЦИ] во С

1. Да се напише програма која дадена матрица ја трансформира во следниов облик:

$$\begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & a_{00} & a_{01} & \dots & a_{0n} & 0 \\ 0 & a_{10} & a_{11} & \dots & a_{1n} & 0 \\ 0 & \vdots & \vdots & \dots & \vdots & 0 \\ 0 & a_{n0} & a_{n1} & \ddots & a_{nn} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
#include <stdio.h>
#define MAX 100

int main ()
{
    int i, j, red, kol;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=1; i<=red; i++)
    {
        for (j=1; j<=kol; j++)
        {
            printf("a[%d,%d]=",i-1,j-1);
            scanf("%d", &a[i][j]);
        }
    }

    for (i=0; i<red+2; i++)
    {
        for (j=0; j<kol+2; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```


2. Да се состави програма која ќе изврши замена на секој елемент на дадената матрица со сумата на неговите непосредни соседи по колони и редови. Почетната матрица се внесува преку тастатура. Резултатот се сместува во нова матрица, која потоа се печати. Максималните димензии на матриците се 10 x 10.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 7 & 8 & 9 & 10 & 9 \\ 8 & 7 & 6 & 5 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 6 & 9 & 12 & 9 \\ 9 & 13 & 17 & 21 & 19 \\ 15 & 24 & 28 & 32 & 23 \\ 22 & 30 & 32 & 32 & 24 \\ 14 & 22 & 21 & 20 & 14 \end{bmatrix} \begin{matrix} x_1 & d_1 & x_2 \\ d_2 & 0 & d_3 \\ x_3 & d_4 & x_3 \end{matrix} \quad \Sigma = d_1 + d_2 + d_3 + d_4$$

НАЧИН:

```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, red, kol;
    int a[MAX][MAX], b[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=1; i<red+1; i++)
    {
        for (j=1; j<kol+1; j++)
        {
            printf("a[%d,%d]= ", i-1, j-1);
            scanf("%d", &a[i][j]);
        }
    }

    /*Зголемување на рангот на матрицата за 2 нулти колони и редови*/
    for (i=0; i<red+2; i++)
    {
        for (j=0; j<kol+2; j++)
        {
            b[i][j]=a[i][j];
        }
    }

    /*Определување на соодветната сума*/
    for (i=1; i<red+1; i++)
    {
        for (j=1; j<kol+1; j++)
        {
            b[i-1][j-1]=b[i][j+1]+b[i][j-1]+b[i+1][j]+b[i-1][j];
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
```

```

        {
            printf("%d\t",b[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

II НАЧИН:

```

#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, red, kol, suma;
    int a[MAX][MAX], b[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d,%d]=",i,j);
            scanf("%d",&a[i][j]);
        }
    }

    for (i=0;i<red;i++)
    {
        for (j=0;j<kol;j++)
        {
            suma=0;
            if (i>0)      suma+=a[i-1][j];
            if (i<red-1)  suma+=a[i+1][j];
            if (j>0)      suma+=a[i][j-1];
            if (j<kol-1)  suma+=a[i][j+1];
            /* Ако i=j=0 или i=red или j=kol прескокни! */
            b[i][j]=suma;
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",b[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

3. Да се состави програма која ќе ги промени местата на редот со минимална сума на елементи и редот со максимална сума на елементи во дадена целобројна матрица $A_{m \times n}$. Елементите на почетната матрица се внесуваат од тастатура, а елементите на променетата матрица да се испечатат на екран. Максималните димензии на матрицата се 10 x 10.

$$\begin{bmatrix} \textcolor{red}{1} & \textcolor{red}{2} & \textcolor{red}{1} & \textcolor{red}{0} \\ \textcolor{green}{5} & \textcolor{green}{3} & \textcolor{green}{4} & \textcolor{green}{9} \\ 4 & 4 & 8 & 5 \\ 0 & 2 & 10 & 6 \\ 2 & 6 & 2 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} \textcolor{green}{5} & \textcolor{green}{3} & \textcolor{green}{4} & \textcolor{green}{9} \\ \textcolor{red}{1} & \textcolor{red}{2} & \textcolor{red}{1} & \textcolor{red}{0} \\ 4 & 4 & 8 & 5 \\ 0 & 2 & 10 & 6 \\ 2 & 6 & 2 & 3 \end{bmatrix}$$

І НАЧИН:

```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i,j, red, kol, min=0, max=0;
    int zmax, zmin, zbir, z=0;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for(j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j);
            scanf("%d",&a[i][j]);
        }
    }

    for(j=0; j<kol; j++) z+=a[0][j];
    zmin=z;
    zmax=z;

    for (i=0; i<red; i++)
    {
        zbir=0;
        for(j=0; j<kol; j++) zbir+=a[i][j];
        if (zbir>zmax)
        {
            zmax=zbir;
            max=i;
        }
        if (zbir<zmin)
        {
            zmin=zbir;
            min=i;
        }
    }

    printf("Max: %d   Min: %d\n", max, min);
```

```

    for (i=0; i<red; i++)
    {
        for(j=0; j<kol; j++)
        {
            if (i==max) printf("%d\t",a[min][j]);
            else
            {
                if (i==min) printf("%d\t",a[max][j]);
                else printf("%d\t",a[i][j]);
            }
        }
        printf("\n");
    }
    return 0;
}

```

II НАЧИН:

```

#include <stdio.h>
#define MAX 10

int main ()
{
    int i,j, red, kol, min=0, max=0, temp;
    int zmax, zmin, zbir, z=0;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for(j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j);
            scanf("%d",&a[i][j]);
        }
    }

    for(j=0; j<kol; j++) z+=a[0][j];
    zmin=z;
    zmax=z;

    for (i=0; i<red; i++)
    {
        zbir=0;
        for(j=0; j<kol; j++) zbir+=a[i][j];
        if (zbir>zmax)
        {
            zmax=zbir;
            max=i;
        }
        if (zbir<zmin)
        {

```

```

        zmin=zbir;
        min=i;
    }
}

printf("Max: %d   Min: %d\n", max, min);

for (j=0; j<kol; j++)
{
    temp=a[max][j];
    a[max][j]=a[min][j];
    a[min][j]=temp;
}

for (i=0; i<red; i++)
{
    for(j=0; j<kol; j++)
    {
        printf("%d\t",a[i][j]);
    }
    printf("\n");
}
return 0;
}

```

4. Да се напише програма која за дадена матрица ќе ја пронајде редицата (нејзиниот индекс) која има максимален производ на елементите и истата ќе ја испечати.

$$\begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 3 & 7 & 4 \\ 1 & 1 & 5 & 7 \\ -1 & 6 & 5 & 10 \end{bmatrix} \rightarrow 2 \quad 3 \quad 7 \quad 4$$

```

#include <stdio.h>
#define MAX 10

int main ()
{
    int i,j, max_proizvod=1, proizvod=1, maxred=0, red, kol;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (j=0; j<kol; j++) max_proizvod*=a[0][j];

    for (i=0;i<red;i++)

```

```

{
    proizvod=1;
    for (j=0; j<kol; j++) proizvod*=a[i][j];
    if (proizvod>max_proizvod)
    {
        max_proizvod=proizvod;
        maxred=i;
    }
}

printf("Max. proizvod ima redot br. %d\n",maxred);
for (j=0; j<kol; j++) printf("%d\t",a[maxred][j]);
return 0;
}

```

5. Да се напише програма која дадена матрица ќе ја измени на начин што на крај матрицата ќе личи на шаховска табла, со тоа што елементите во „црните“ полиња ќе добијат нулта вредност а останатите ќе ја задржат старата вредност.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 7 & 3 & 9 & 2 & 8 \\ 6 & 7 & 8 & 9 & 10 \\ 7 & 8 & 9 & 10 & 9 \\ 8 & 7 & 6 & 5 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 3 & 0 & 5 \\ 0 & 3 & 0 & 2 & 0 \\ 6 & 0 & 8 & 0 & 10 \\ 0 & 8 & 0 & 10 & 0 \\ 8 & 0 & 6 & 0 & 4 \end{bmatrix}$$

1 НАЧИН:

```

#include <stdio.h>
#define MAX 10

int main ()
{
    int i,j, red, kol;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        if (i%2)
        {
            for (j=1; j<kol; j+=2) a[i][j]=0;
        }
        else
        {
            for(j=0; j<kol; j+=2) a[i][j]=0;
        }
    }
}

```

```

    }
}

for (i=0; i<red; i++)
{
    for (j=0; j<kol; j++)
    {
        printf("%d\t",a[i][j]);
    }
    printf("\n");
}
return 0;
}

```

II НАЧИН:

```

#include <stdio.h>
#define MAX 10

int main ()
{
    int i,j, red, kol;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

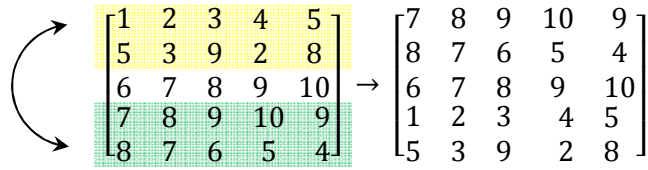
    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=1-i%2; j<kol; j+=2) a[i][j]=0;
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

6. Да се напише програма која дадена матрица ќе ја измени на начин што елементите од горната половина на матрицата ќе си ги сменат местата со елементите од долната половина.



```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i,j, red, kol, temp;
    int a[MAX][MAX];

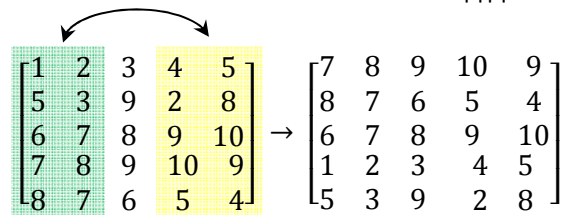
    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red/2; i++)
    {
        for (j=0; j<kol; j++)
        {
            temp=a[i][j];
            a[i][j]=a[i+(red+1)/2][j];
            a[i+(red+1)/2][j]=temp;
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```


7. Да се напише програма која дадена матрица ќе ја измени на начин што елементите од левата половина на матрицата ќе си ги сменат местата со елементите од десната половина.



```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i,j, red, kol, temp;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol/2; j++)
        {
            temp=a[i][j];
            a[i][j]=a[i][j+(kol+1)/2];
            a[i][j+(kol+1)/2]=temp;
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

8. Да се напише програма која ќе споредува дали збирот на парните брови од левата половина на една матрица е еднков со збирот на непарните брови од десната половина на матрицата.

$$\begin{bmatrix} 1 & 7 & 3 & 4 & 5 \\ 6 & 3 & 9 & 2 & 8 \\ 6 & 7 & 8 & 9 & 10 \\ 7 & 8 & 9 & 10 & 9 \\ 8 & 7 & 6 & 5 & 4 \end{bmatrix}$$

$$6 + 6 + 8 + 8 = 5 + 9 + 9 + 5$$

```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, red, kol;
    int lev_zbir=0, desen_zbir=0;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol/2; j++)
        {
            if (!(a[i][j]%2)) lev_zbir+=a[i][j];
        }
    }

    for (j=kol-1; j>=(kol+1)/2;j--)
    {
        for (i=0; i<red;i++)
        {
            if (a[i][j]%2) desen_zbir+=a[i][j];
        }
    }

    if (lev_zbir==desen_zbir) printf ("Matricata go ispolnuva baraniot uslov.\n");
    else printf("Matricat ne go ispolnuva baraniot uslov.\n");
    return 0;
}
```

9. Да се напише програма која ќе споредува дали збирот на простите броеви од горната половина на дадена матрица е еднаков со збирот на простите броеви од долната половина на матрицата.

$$\begin{bmatrix} 1 & 2 & 1 & 1 & 7 \\ 3 & 6 & 4 & 5 & 4 \\ 10 & 4 & 8 & 0 & 7 \\ 5 & 7 & 6 & 0 & 3 \\ 9 & 8 & 9 & 0 & 2 \end{bmatrix}$$

$$2 + 7 + 3 + 5 = 5 + 7 + 3 + 2$$

```
#include <stdio.h>
#define MAX 10

int prost (int br)
{
    int i,x=0;
    if (br==1) return 0;
    for (i=1;i<=br;i++)
    {
        if (!(br%i)) x++;
    }
    if (x<=2) return br;
    else return 0;
}

int main ()
{
    int i, j, red, kol, br;
    int goren_zbir=0, dolen_zbir=0;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red/2; i++)
    {
        for (j=0; j<kol; j++)
        {
            br=a[i][j];
            if (prost(br)) goren_zbir+=br;
        }
    }

    for (j=0; j<kol; j++)
    {
        for (i=red-1; i>=(red+1)/2; i--)
```

```

        {
            br=a[i][j];
            if (prost(br))  dolen_zbir+=br;
        }
    }

    if (dolen_zbir==goren_zbir) printf("Matricata go ispolnuva uslovot.\n");
    else printf("Matricata ne go ispolnuva uslovot.\n");
    return 0;
}

```

10. Да се напише програма што ќе прочита, а потоа ќе трансформира една квадратна матрица од цели броеви, така што во новодобиената матрица сите елементи од главната дијагонала кои се парни ќе бидат зголемени за 2. На крај матрицата треба да се прикаже на екран.

$$\begin{bmatrix} 8 & 3 & 4 & 5 & 7 \\ 5 & 5 & 6 & 5 & 5 \\ 0 & 6 & 4 & 4 & 3 \\ 2 & 2 & 5 & 7 & 0 \\ 3 & 4 & 6 & 7 & 10 \end{bmatrix} \rightarrow \begin{bmatrix} 10 & 3 & 4 & 5 & 7 \\ 5 & 5 & 6 & 5 & 5 \\ 0 & 6 & 6 & 4 & 3 \\ 2 & 2 & 5 & 7 & 0 \\ 3 & 4 & 6 & 7 & 12 \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 10

int main ()
{
    int i,j, rang;
    int a[MAX][MAX];

    printf("Vnesi rang na mtricata: "); scanf("%d",&rang);

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++)
    {
        if (a[i][i]%2) a[i][i]+=2;
    }

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

11. Да се напише програма која дадена квадратна матрица ќе ја измени на начин, со тоа што елементите од главната дијгонала ќе имаат вредност еднаква на сумата на „жолтите“ полиња (обелжани на сликот со круг). Останатите полиња ќе останат непроменети. На крај да се отпечати новодобиената матрица. Максималната димензија на матрицата е 10 x 10.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 7 & 8 & 9 & 10 & 9 \\ 8 & 7 & 6 & 5 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 68 & 2 & 3 & 4 & 5 \\ 1 & 68 & 3 & 4 & 5 \\ 6 & 7 & 68 & 9 & 10 \\ 7 & 8 & 9 & 68 & 9 \\ 8 & 7 & 6 & 5 & 68 \end{bmatrix}$$

```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, rang, zbir=0;
    int a[MAX][MAX];

    printf("Vnesi rang na matricata: "); scanf("%d",&rang);

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a(%d%d)= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++)
    for (j=1-i%2; j<rang; j+=2) zbir+=a[i][j];

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            if (i==j) printf("%d\t",zbir);
            else printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

12. Да се напише програма која дадена матрица ќе ја измени на начин, со тоа што елементите од споредната дијагонала ќе имаат вредност еднаква на сумата на „жолтите“ полиња (обелжани на сликата со правоаголник). Останатите полиња ќе останат непроменети. На крај да се отпечати новодобиената матрица. Максималната димензија на матрицата е 10 x 10.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 7 & 8 & 9 & 10 & 9 \\ 8 & 7 & 6 & 5 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 3 & 4 & 75 \\ 1 & 2 & 3 & 75 & 5 \\ 6 & 7 & 75 & 9 & 10 \\ 7 & 75 & 9 & 10 & 9 \\ 75 & 7 & 6 & 5 & 4 \end{bmatrix}$$

```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, rang, zbir=0;
    int a[MAX][MAX];

    printf("Vnesi rang na matricata: "); scanf("%d",&rang);

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a(%d%d)= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++)
    for (j=i%2; j<rang; j+=2) zbir+=a[i][j];

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            if ((i+j)==rang-1) printf("%d\t",zbir);
            else printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

13. Да се напише програма која ќе изврши замена на секој елемент на дадена матрица со сумата на неговите дијаметрални соседи. Почетната матрица се внесува преку тастатура. Резултатот се сместува во нова матрица, која потоа се печати. Максималните димензии на матриците се 100 x 100.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 7 & 8 & 9 & 10 & 9 \\ 8 & 7 & 6 & 5 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 4 & 6 & 8 & 4 \\ 9 & 18 & 22 & 26 & 13 \\ 10 & 20 & 24 & 26 & 14 \\ 14 & 28 & 28 & 28 & 14 \\ 8 & 16 & 18 & 18 & 10 \end{bmatrix} \quad \begin{matrix} d_1 & x_2 & d_2 \\ x_1 & 0 & x_3 \\ d_4 & x_3 & d_3 \end{matrix} \quad \Sigma = d_1 + d_2 + d_3 + d_4$$

1 НАЧИН:

```

#include <stdio.h>
#define MAX 100

int main ()
{
    int i, j, red, kol;
    int suma=0;
    int a[MAX][MAX], b[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            b[i][j]=0;
            if (i>0) b[i][j]+=a[i-1][j-1];
            if (i<red-1) b[i][j]+=a[i+1][j+1];
            if (j>0) b[i][j]+=a[i+1][j-1];
            if (j<kol-1) b[i][j]+=a[i-1][j+1];
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",b[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

II НАЧИН:

```

#include <stdio.h>
#define MAX 100

int main ()
{
    int i, j, red, kol;
    int suma=0;
    int a[MAX][MAX], b[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a(%d%d)=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            b[i][j]=0;
            b[i][j]+=a[i-1][j-1]+a[i+1][j+1]+a[i+1][j-1]+a[i-1][j+1];
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",b[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

III НАЧИН:

```

#include <stdio.h>
#define MAX 100

int main ()
{
    int i, j, red, kol;
    int suma=0;
    int a[MAX][MAX], b[MAX][MAX];

```



```

printf("Vnesi br. redovi: "); scanf("%d",&red);
printf("Vnesi br. koloni: "); scanf("%d",&kol);

for (i=0; i<red; i++)
{
    for (j=0; j<kol; j++)
    {
        printf("a(%d%d)=",i,j);  scanf("%d",&a[i][j]);
    }
}

for (i=0; i<red; i++)
{
    for (j=0; j<kol; j++)
    {
        int r, k;
        b[i][j];
        r=i-1;
        if (r>=0 && r<red)
        {
            k=j-1;
            if (k>=0 && k<kol) b[i][j]+=a[r][k];
            k=j+1;
            if (k>=0 && k<kol) b[i][j]+=a[r][k];
        }
        r=i+1;
        if (r>=0 && r<red)
        {
            k=j-1;
            if (k>=0 && k<kol) b[i][j]+=a[r][k];
            k=j+1;
            if (k>=0 && k<kol) b[i][j]+=a[r][k];
        }
    }
}

for (i=0; i<red; i++)
{
    for (j=0; j<kol; j++)
    {
        printf("%d\t",b[i][j]);
    }
    printf("\n");
}

return 0;
}

```

14. Да се напише програма која за дадена квадратна матрица ќе ги испечати елементите над главната и споредната дијагонала и елементите под главната и споредната дијагонала. Печатењето да биде во форма на матрица.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 8 \\ 5 & 6 & 7 & 2 & 6 \\ 0 & 2 & 4 & 5 & 8 \\ 4 & 6 & 7 & 0 & 3 \\ 1 & 1 & 5 & 6 & 2 \end{bmatrix} \rightarrow \begin{matrix} & & & & 1 & & & & 8 \\ & & & & & 6 & & & 2 \\ & & & & & & 4 & & \\ & & & & 6 & & & 0 & \\ 1 & & & & & & & & 2 \end{matrix}$$

```
#include <stdio.h>
#define MAX 100

int main ()
{
    int i, j, rang;
    int suma=0;
    int a[MAX][MAX], b[MAX][MAX];

    printf("Vnesi rang na matricata: "); scanf("%d",&rang);

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            if ((i==j) || (i+j==rang-1)) printf ("%d\t",a[i][j]);
            else printf("\t");
        }
        printf("\n\n");
    }
    return 0;
}
```

15. Да се напише програма која ќе им ги промени местата на колоната со минимална сума на елементи и колоната со максимална сума на елементи во дадена целобројна матрица $A_{m \times n}$. Елементите на матрицата се внесуваат од тастатура, а елементите на променетата матрица да се вметнат во нова матрица $B_{m \times n}$ и да се испечатат на екран. Максималните димензии на матриците се 10 x 10.

$$\begin{bmatrix} \textcolor{red}{1} & 2 & \textcolor{blue}{1} & 0 \\ \textcolor{red}{5} & 3 & \textcolor{blue}{4} & 9 \\ \textcolor{red}{4} & 4 & \textcolor{blue}{8} & 5 \\ \textcolor{red}{0} & 2 & \textcolor{blue}{10} & 6 \\ \textcolor{red}{2} & 6 & \textcolor{blue}{2} & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 1 & 0 \\ 4 & 3 & 5 & 9 \\ 8 & 4 & 4 & 5 \\ 10 & 2 & 0 & 6 \\ 2 & 6 & 2 & 3 \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, red, kol, zbir, temp;
    int zbir_min=0, poz_min=0;
    int zbir_max=0, poz_max=0;
    int a[MAX][MAX], b[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d]= "); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++) zbir_min+=a[i][0];
    zbir_min=zbir_max;

    for (j=0; j<kol; j++)
    {
        zbir=0;
        for (i=0; i<red; i++)
        {
            zbir+=a[i][j];
            if (zbir>zbir_max)
            {
                zbir_max=zbir;
                poz_max=j;
            }
            if (zbir<zbir_min)
            {
                zbir_min=zbir;
                poz_min=j;
            }
        }
    }

    for (i=0; i<red; i++)

```

```

    {
        temp=a[i][poz_min];
        a[i][poz_min]=a[i][poz_max];
        a[i][poz_max]=temp;
    }

    for (i=0;i<red;i++)
    {
        for (j=0;j<kol;j++)
        {
            b[i][j]=a[i][j];
        }
    }

    for (i=0;i<red;i++)
    {
        for (j=0;j<kol;j++)
        {
            printf("%d\t",b[i][j]);
        }
        printf("\n\n");
    }
    return 0;
}

```

16. Да се напише програма која ќе заменува редица со колона на дадена квадратна матрица.

```

#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, red, kol,temp;
    int red_zamena, kol_zamena;
    int a[MAX][MAX], b[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    printf("Vnesi redica za zamena: "); scanf("%d",&red_zamena);
    printf("Vnesi kolona za zamena: "); scanf("%d",&kol_zamena);

    i=0;
    j=0;
    while (i<red && j<kol)
    {

```

```

        temp=a[i][kol_zamena-1];
        a[i][kol_zamena-1]=a[red_zamena-1][j];
        a[red_zamena-1][j]=temp;
        i++;
        j++;
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

17. Да се напише програма која ќе го зголеми рангот на матрицата со копирање на зададена редица веднаш до оригиналот.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 100

int main ()
{
    int i, j, red, kol;
    int red_copy;
    int a[MAX][MAX], b[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    printf("Vnesi redica za kopiranje: "); scanf("%d",&red_copy);

    for (i=0; i<red_copy; i++)
    {
        for (j=0; j<kol; j++)
        {
            b[i][j]=a[i][j];
        }
    }
}

```

```

for (i=red_copy; i<red; i++)
{
    for (j=0; j<kol; j++)
    {
        b[i+1][j]=a[i][j];
    }
}

for (j=0; j<kol; j++) b[red_copy][j]=a[red_copy-1][j];

for (i=0; i<=red; i++)
{
    for (j=0; j<kol; j++)
    {
        printf("%d\t",b[i][j]);
    }
    printf("\n\n");
}
return 0;
}

```

18. Да се напише програма која ќе пресмета средна вредност на елементите кои се наоѓаат истовремено под главната и под споредната дијгонала на една квадратна матрица.

1	2	3	4	5
4	7	9	2	1
2	4	0	8	6
1	2	3	4	5
2	2	4	5	2

$$(2 + 3 + 4 + 5)/4 = 3.5$$

```

#include <stdio.h>
#define MAX 100

int main ()
{
    int i, j, rang, zbir=0, vk=0;
    int a[MAX][MAX];

    printf("Vnesi rang na matricata: "); scanf("%d",&rang);

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            if ((i+j)>(rang-1) && (i>j))
            {

```

```

        zbir+=a[i][j];
        vk++;
    }
}
printf("%d / %d = %.3f\n", zbir, vk, (float)zbir/vk);
return 0;
}

```

19. Да се напише програма во која од тастатура се внесува матрица (димензиите на матрицата се внесуваат од тастатура, не поголеми од 50 x 50). Да се испечати бројот кој се наоѓа на позиција на пресек на редицата и колоната што имаат најголем производ на елементите.

1	2	3	4
8	2	0	9
6	5	7	1
5	9	2	3

→ 5

```

#include <stdio.h>
#define MAX 50

int main ()
{
    int i, j, red, kol;
    int max_kol=1, max_red=1, proizvod, poz_red=0, poz_kol=0;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d]= ", i, j); scanf("%d",&a[i][j]);
        }
    }

    for (j=0; j<kol; j++) max_kol*=a[0][j];
    for (i=0; i<red; i++) max_red*=a[i][0];

    /*Производ на ред*/
    for (i=0; i<red; i++)
    {
        proizvod=1;
        for (j=0; j<kol; j++)
        {
            proizvod*=a[i][j];
            if (proizvod>max_red)
            {
                max_red=proizvod;
                poz_red=i;
            }
        }
    }
}

```

```

/*Производ на колона*/
for (j=0; j<kol; j++)
{
    proizvod=1;
    for (i=0; i<red; i++)
    {
        proizvod*=a[i][j];
        if (proizvod>max_kol)
        {
            max_kol=proizvod;
            poz_kol=j;
        }
    }
}

printf("\na[%d%d] = %d", poz_red, poz_kol, a[poz_red][poz_kol]);
return 0;
}

```

20. Да се напише програма во која од тастатура се внесува матрица (димензиите на матрицата се внесуваат од тастатура, не поголеми од 50 x 50). Да се најде најголемиот елемент на матрицата и да се испечати на која позиција се наоѓа.

```

#include <stdio.h>
#define MAX 50

int main ()
{
    int i, j, red, kol;
    int a[MAX][MAX];
    int max=a[0][0], i_max=0, j_max=0;

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            if (a[i][j]>max)
            {
                max=a[i][j];
                i_max=i;
                j_max=j;
            }
        }
    }
}

```



```

    }
}
printf("a[%d%d] = %d",i_max, j_max, max);
return 0;
}

```

21. Да се напише програма во која од тасттура се внесува квадратна матрица со димензии $n \times n$ (n се внесува од тастатура). Програмата треба да ја пресмета сумата на сите елементи од матрицата. Потоа треба да ја измени матрицата во долно триаголна, на тој начин што сите елементи под главната дијгонала ќе добијат вредност еднаква на пресметаната сума. На крај треба да се испечати добиената матрица во формат на матрица. Максималните димензии на матрицата се 10×10 .

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 5 & 6 & 7 & 8 \\ 2 & 7 & 8 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} & & & 72 \\ & & 72 & 72 \\ & 72 & 72 & 72 \\ 72 & 72 & 72 & \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 100
int main ()
{
    int i, j, rang, zbir=0;
    int a[MAX][MAX];

    printf("Vnesi rang na matricata: "); scanf("%d",&rang);

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            zbir+=a[i][j];
        }
    }

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            if ((i+j)>(rang-1)) printf("%d\t",zbir);
            else printf("\t");
        }
        printf("\n\n");
    }
    return 0;
}

```

22. Да се состави програма која ќе изврши замена на секој елемент на дадена матрица со разликата на сумата на елементите од редот и сумата на елементите од колоната на кои им припаѓа ((сума на ред) – (сума на колона)). Притоа вредноста на елементот кој го заменуваме не влегува во сумите. Почетната матрица се внесува преку тастатура. Резултатот се сместува во нова матрица која потоа се печати. Максималните димнзии на матриците е 100 x 100.

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 2 & 3 & 4 & 6 \\ 1 & 6 & 8 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 8 & 1 & -4 & -2 \\ 11 & 4 & -1 & 1 \\ 14 & 7 & 2 & 4 \end{bmatrix}$$

```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, k, red, kol, rang;
    int a[MAX][MAX], b[MAX][MAX];
    int sumared, sumakol;

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        /* Сума на ред */
        sumared=0;
        for (k=0; k<kol; k++) sumared+=a[i][k];
        for (j=0; j<kol; j++)
        {
            /* Сума на колона */
            sumakol=0;
            for (k=0; k<red; k++) sumakol+=a[k][j];
            b[i][j]=sumared-sumakol;
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",b[i][j]);
        }
        printf("\n\n");
    }
    return 0;
}
```

23. Да се состави програма во која преку тастатура се внесува димензија на матрица како и вредностите на матрицата (со димензија не поголема од 50 x 50). Потоа во програмата од елементите на матрицата над споредната дијагонала треба да се најде елементот со најмала вредност, а од елементите на матрицата под споредната дијагонала да се најде елементот со најголема вредност. Елементите од споредната дијагонала да се заменат со сумата на двете најдени вредности.

$$\begin{bmatrix} 9 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 11 \\ 7 & 8 & 9 & 10 & 9 \\ 8 & 7 & 6 & 5 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 9 & 2 & 3 & 4 & 12 \\ 1 & 2 & 3 & 12 & 5 \\ 6 & 7 & 12 & 9 & 1 \\ 7 & 12 & 9 & 10 & 9 \\ 12 & 7 & 6 & 5 & 4 \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 50

int main ()
{
    int i, j, rang;
    int A[MAX][MAX];

    printf("Vnesi rang na mtrica: "); scanf("%d",&rang);

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

    int min=A[0][0], max=A[rang-1][rang-1];

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            if (i+j>rang-1)
            {
                if (A[i][j]>max) max=A[i][j];
            }
            if (i+j<rang-1)
            {
                if (A[i][j]<min) min=A[i][j];
            }
        }
    }

    for (i=0; i<rang; i++) A[i][rang-1-i]=max+min;

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("%d\t",A[i][j]);
        }
    }
}

```

```

    }
    printf("\n");
}
return 0;
}

```

24. Да се состави програма во која преку тастатура се внесуваат елементите на две матрици A и B со ист број на редици и колони, не повеќе од 50 x 50. Потоа се генерира трета матрица C со ист број редици и колони како A и B. Секој елемент од матрицата C[i,j] има вредност 0 или 1 во зависност од тоа дали елементот A[i,j] е квадрат на B[i,j]. Резултатната матрица C се печати на екран.

$$A = \begin{bmatrix} 2 & 4 & 6 & 16 & 4 \\ 9 & 18 & 22 & 26 & 13 \\ 36 & 20 & 24 & 26 & 14 \\ 49 & 28 & 28 & 28 & 81 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 0 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 7 & 8 & 0 & 10 & 9 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 50

int main ()
{
    int i, j, red, kol;
    int A[MAX][MAX], B[MAX][MAX], C[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    printf("Matrica A: \n");
    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("A[%d%d] = ",i,j); scanf("%d",&A[i][j]);
        }
    }

    printf("Matrica B: \n");
    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("B[%d%d] = ",i,j); scanf("%d",&B[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            if (A[i][j]==B[i][j]*B[i][j]) C[i][j]=1;
            else C[i][j]=0;
        }
    }

    printf("Matrica C: \n");

```

```

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",C[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

25. Да се напише програма која ќе прочита квадратна матрица со максимални димензии 100 x 100 и ќе провери дали матрицата е магичен квадрат. Магичен квадрат е матрица чии што суми на елементи по колони и редици имаат иста вредност.

$$\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

Во матрицата од примерот сумата на елементите по сите колони и сите редови е 15. Дополнително програмата треба да провери дали матрицата е специјален магичен квадрат односно сумата на елементите од главната и споредната дијагонала е еднаков на сумата на елементите по колони и редици.

```

#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, rang, zbir=0, x=0, k=0, sumagd=0, sumasd=0;
    int a[MAX][MAX], b[MAX][MAX];
    int sumared, sumakol;

    printf("Vnesi rang na matricata: "); scanf("%d",&rang);

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++) zbir+=a[i][0];

    for (i=0; i<rang; i++)
    {
        sumared=0;
        for (j=0; j<rang; j++) sumared+=a[i][j];
        if (sumared!=zbir) x++;
    }

    for (j=0; j<rang; j++)
    {
        sumakol=0;

```

```

        for (i=0; i<rang; i++) sumakol+=a[i][j];
        if (sumakol!=zbir) x++;
    }

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            if (i==j) sumagd+=a[i][j];
            if ((i+j)==(rang-1)) sumasd+=a[i][j];
        }
    }
    if ((sumagd!=zbir) || (sumasd!=zbir)) k++;

    if (!x) printf("Matricat e magicen kvadrat.\n");
    else printf("Matricata ne e magicen kvadrat.\n");
    if (!k) printf("Matricata e specijalen magicen kvadrat.\n");
    return 0;
}

```

26. Да се напише програма која ќе прочита квадратна матрица со максимални димензии 100 x 100 и ќе провери дали збирот на цифрите на сите елементи од главната дијагонала е еднаков на збирот на цифрите на сите елементи од споредната дијагонала. Пресметката на збирот на цифри за елементите да се реализира со посебна **рекурзивна функција**.

$$\begin{bmatrix} 12 & 56 & 4 & 0 \\ 23 & 34 & 25 & 3 \\ 45 & 123 & 58 & 9 \\ 67 & 3 & 5 & 3 \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 100

int zbir_cifri(int n)
{
    if (n<10) return n;
    else return (n%10+zbir_cifri(n/10));
}

int main ()
{
    int i, j, rang, zbirgd=0, zbirsd=0, gd, sd;
    int a[MAX][MAX];

    printf("Vnesi rang na matrica: "); scanf("%d",&rang);

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a[%d%d] = ",i,j);    scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++)

```

```

{
    gd=a[i][i];
    sd=a[i][rang-1-i];
    zbirgd+=zbir_cifri(gd);
    zbirsd+=zbir_cifri(sd);
}

if (zbirgd==zbirsd) printf("Matricata go ispolnuva uslovot.\n");
else printf("Matricata ne go ispolnuva uslovot.\n");
return 0;
}

```

27. Да се напише програма во која од тастатура се внесуваат елементите (целобројни вредности) на матрица со димензии $m \times n$ (не поголеми од 100). Потребно е да се направи промена на матрицата така што максималниот елемент ќе ја добие вредноста на сумата на останатите елементи во таа редица.

$$\begin{bmatrix} 3 & -2 & 5 & 1 \\ -7 & 1 & 0 & 6 \\ 1 & 7 & 2 & -8 \\ 4 & -3 & 1 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & -2 & 5 & 1 \\ -7 & 1 & 0 & 6 \\ 1 & -5 & 2 & -8 \\ 4 & -3 & 1 & 2 \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 100

int main ()
{
    int i, j, red, kol;
    int a[MAX][MAX];
    int max, poz_i=0, poz_j=0, zbir=0;

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d] = ",i,j); scanf("%d",&a[i][j]);
        }
    }

    max=a[0][0];
    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            if (a[i][j]>max)
            {
                max=a[i][j];
                poz_i=i;
                poz_j=j;
            }
        }
    }
}

```

```

    for (j=0; j<kol; j++) zbir+=a[poz_i][j];
    printf("%d\n", zbir);
    a[poz_i][poz_j]=zbir-max;

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t", a[i][j]);

        }
        printf("\n\n");
    }
    return 0;
}

```

28. Да се напише програма во која се внесува квадратна матрица од цели броеви со максимална големина од 10 x 10. Матрицата да се измени на таков начин што елементите од главната и споредната дијагонала ќе си ги заменат своите места. На крај да се отпечати променетата матрица.

$$\begin{bmatrix} \color{red}{3} & 4 & 5 & 6 & \color{red}{7} \\ 1 & \color{red}{2} & 3 & \color{red}{6} & 4 \\ 4 & 2 & \color{red}{7} & 9 & 1 \\ 1 & \color{red}{9} & 0 & \color{red}{3} & 5 \\ \color{red}{4} & 6 & 2 & 8 & \color{red}{1} \end{bmatrix} \rightarrow \begin{bmatrix} \color{red}{7} & 4 & 5 & 6 & \color{red}{3} \\ 1 & \color{red}{6} & 3 & \color{red}{2} & 4 \\ 4 & 2 & \color{red}{7} & 9 & 1 \\ 1 & \color{red}{3} & 0 & \color{red}{9} & 5 \\ \color{red}{1} & 6 & 2 & 8 & \color{red}{4} \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 50

int main ()
{
    int i, j, rang, temp;
    int a[MAX][MAX];

    printf("Vnesi rang na matricata: "); scanf("%d",&rang);

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a[%d%d]= ", i, j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++)
    {
        temp=a[i][i];
        a[i][i]=a[i][rang-1-i];
        a[i][rang-1-i]=temp;
    }

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)

```



```

        {
            printf("%d\t",a[i][j]);
        }
        printf("\n\n");
    }
    return 0;
}

```

29. Да се напише програма која ќе определи и ќе испише на екран дали збирот на парните елементи кои лежат над главната дијагонала е еднаков со збирот на непарните елементи кои лежат под споредната дијагонала на една квадратна матрица. Квадратната матрица како и нејзината димензија ја внесува корисникот преку тастатура.

1	2	3	4	5
4	5	2	7	6
1	8	9	3	2
2	3	4	1	4
5	7	4	8	9

```

#include <stdio.h>
#define MAX 50

int main ()
{
    int i, j, rang, zbir1=0, zbir2=0;
    int a[MAX][MAX];

    printf("Vnesi rang na matrica: "); scanf("%d",&rang);
    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            printf("a[%d%d] = ",i,j);  scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<rang; i++)
    {
        for (j=0; j<rang; j++)
        {
            if ((i<j) && !(a[i][j]%2)) zbir1+=a[i][j];
            if ((i+j>rang-1) && (a[i][j]%2)) zbir2+=a[i][j];
        }
    }

    if (zbir1==zbir2) printf("Matricata go ispolnuva uslovot.\n");
    else printf("Matricata ne go ispolnuva uslovot.\n");
    return 0;
}

```

30. Да се напише програма која дадена матрица ќе ја промени на следниов начин:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, red, kol;
    int a[MAX][MAX], b[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            b[i][j]=a[red-1-i][kol-1-j];
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",b[i][j]);
        }
        printf("\n\n");
    }
    return 0;
}
```

31. Да се напише програма со која една матрица $A_{m \times n}$ ќе се замени со еднодимензионална низа $P_{m \cdot n}$. Секој елемент на низата е соодветен елемент од матрицата.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$$

```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, red, kol, k=0;
    int a[MAX][MAX], p[MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d]= ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            p[k]=a[i][j];
            k++;
        }
    }

    for (k=0; k<red*kol;k++)
    {
        printf("%d\t",p[k]);
    }
    return 0;
}
```

32. Да се напише програма која во дадена матрица броевите кои се повторуваат ќе ги замени со празно место.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 10 & 7 & 8 & 4 & -1 \\ 0 & 85 & 6 & 0 & 3 \\ 1 & 54 & 22 & 13 & 1 \\ 35 & 11 & 3 & 0 & 23 \\ 12 & 19 & 3 & -3 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} & 2 & & & 5 \\ 10 & 7 & 8 & & -1 \\ & 85 & 6 & & \\ & 54 & 22 & 13 & \\ 35 & 11 & & & 23 \\ 12 & 19 & & -3 & \end{bmatrix}$$

```
#include <stdio.h>
#define MAX 10

int main ()
{
    int i, j, red, kol, k, q, x;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d]=",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red;i++)
    {
        for (j=0; j<kol; j++)
        {
            x=0;
            for (k=0; k<red; k++)
            {
                for (q=0; q<kol; q++)
                {
                    if (a[i][j]==a[k][q]) x++;
                }
            }
            if (x>1) printf(" \t");
            else printf("%d\t",a[i][j]);
        }
        printf("\n\n");
    }
    return 0;
}
```

33. Да се напише програма која од две внесени матрици $A_{m \times n}$ и $B_{n \times m}$ ќе ја испачати матрицата $C_{m \times m}$ која претставува производ на двете матрици.

$$\begin{bmatrix} 2 & 5 & 1 \\ 4 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 4 & 6 \\ 5 & 8 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 34 & 53 \\ 22 & 33 \end{bmatrix}$$

```
#include <stdio.h>
#define MAX 100

int main ()
{
    int i, j, z, red, kol;
    int a[MAX][MAX], b[MAX][MAX], c[MAX][MAX];
    int max, poz_i=0, poz_j=0, zbir=0;

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d] = ",i,j); scanf("%d",&a[i][j]);
        }
    }

    printf("A(%dx%d):\n",red,kol);
    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n\n");
    }
    printf("\n\n");

    for (i=0; i<kol; i++)
    {
        for (j=0; j<red; j++)
        {
            printf("a[%d%d] = ",i,j); scanf("%d",&b[i][j]);
        }
    }

    printf("B(%dx%d):\n",kol, red);
    for (i=0; i<kol; i++)
    {
        for (j=0; j<red; j++)
        {
            printf("%d\t",b[i][j]);
        }
        printf("\n\n");
    }
}
```

```

printf("\n\n");

for (i=0; i<red; i++)
{
    for (j=0; j<kol; j++)
    {
        c[i][j]=0; /* Нема потреба */
        for (z=0; z<kol; z++)
        {
            c[i][j]+=a[i][z]*b[z][j];
        }
    }
}

printf("C(%dx%d):\n",red,red);
for (i=0; i<red; i++)
{
    for (j=0; j<red; j++)
    {
        printf("%d\t",c[i][j]);
    }
    printf("\n\n");
}
return 0;
}

```

34. Да се напише процедури којашто влезна матрица $A_{m \times n}$ ќе ја преуреди така што колоните и се во растечки редослед на максималните елементи на колоната.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 1 \\ 8 & 6 & 6 & 3 & 3 \\ 0 & 3 & 9 & 2 & 7 \\ 8 & 6 & 9 & 4 & 7 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 2 & 1 & 1 & 3 \\ 3 & 6 & 3 & 8 & 6 \\ 2 & 3 & 7 & 0 & 9 \\ 4 & 6 & 7 & 8 & 9 \end{bmatrix}$$

```

#include <stdio.h>
#define MAX 100

void main ()
{
    int i, j, red, kol, k, t;
    int a[MAX][MAX], max[MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d] = ",i,j); scanf("%d",&a[i][j]);
        }
    }

    for (j=0; j<kol; j++)

```

```

{
    max[j]=a[0][j];
    for (i=1; i<red; i++)
    {
        if (max[j]<a[i][j]) max[j]=a[i][j];
    }
}

for (i=0; i<red; i++)
{
    for (j=i+1; j<kol; j++)
    {
        if (max[i]>max[j])
        {
            t=max[i];
            max[i]=max[j];
            max[j]=t;
            for (k=0; k<red; k++)
            {
                t=a[k][i];
                a[k][i]=a[k][j];
                a[k][j]=t;
            }
        }
    }
}

for (j=0; j<kol; j++) printf("%d\t",max[j]);

for (i=0; i<red; i++)
{
    printf("\n\n");
    for (j=0; j<kol; j++)    printf("%d\t",a[i][j]);
}
}

```

III

ИТЕРАТИВНИ И РЕКУРЗИВНИ ФУНКЦИИ ВО С

1. Да се напише функцииска програма за наоѓање на факториел на даден природен број n .

Итерација:

```
#include <stdio.h>

int faktoriel(int n)
{
    int i, f=1;
    if (n==0) return 1;
    for (i=1; i<=n; i++) f*=i;
    return f;
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    printf("%d! = %d", n, faktoriel(n));
    return 0;
}
```

Рекурзија:

```
#include <stdio.h>

int faktoriel(int n)
{
    int i, f=1;
    if (n==0 || n==1) return 1;
    else return n*faktoriel(n-1);
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    printf("%d! = %d", n, faktoriel(n));
    return 0;
}
```


2. Да се напише функциска програма која ќе пресметува x -ти степен на даден број a . [a^x]

Итерација:

```
#include <stdio.h>

int stepen (int x, int a)
{
    int i, s=1;
    if (x==0) return 1;
    for (i=1;i<=x;i++) s*=a;
    return s;
}

int main ()
{
    int x, a;
    printf("x = ");    scanf("%d",&x);
    printf("a = ");    scanf("%d",&a);
    printf("%d^%d = %d",a, x, stepen(x,a));
    return 0;
}
```

Рекурзија

```
#include <stdio.h>

int stepen (int x, int a)
{
    if (x==0) return 1;
    return a*stepen(x-1,a);
}

int main ()
{
    int x, a;
    printf("x = ");    scanf("%d",&x);
    printf("a = ");    scanf("%d",&a);
    printf("%d^%d = %d",a, x, stepen(x,a));
    return 0;
}
```

3. Да се напише функциска програма која наоѓа збир на цифрите на еден природен број.

Итерација:

```
#include <stdio.h>

int zbir_cifri (int a)
{
    int zbir=0;
    if (a<10) return a;
    while (a)
    {
        zbir+=a%10;
    }
}
```

```

        a/=10;
    }
    return zbir;
}

int main ()
{
    int a;
    printf("a = ");    scanf("%d",&a);
    printf("Zbirot na cifrite na %d e %d",a, zbir_cifri(a));
    return 0;
}

```

Рекурзија:

```

#include <stdio.h>

int zbir_cifri (int a)
{
    if (a<10) return a;
    else return (a%10+zbir_cifri(a/10));
}

int main ()
{
    int a;
    printf("a = ");    scanf("%d",&a);
    printf("Zbirot na cifrite na %d e %d",a, zbir_cifri(a));
    return 0;
}

```

4. Да се напише рекурзивна функцииска програма која ќе пресметува **најголем заеднички делител** и **најмал заеднички содржател** на два дадени броја.

```

#include <stdio.h>
int NZD (int n, int m)
{
    int nzd;
    if (n==m) return n;
    else
    {
        if (n<m)    return NZD(m-n,n);
        else       return NZD(n-m,m);
    }
}

int main ()
{
    int n, m;
    scanf("%d %d",&n, &m);
    printf("NZD(%d,%d) = %d",n,m,NZD(n,m));
    printf("NZS(%d,%d) = %d",n,m,(m*n)/NZD(n,m));
    return 0;
}

```

5. Да се напише програма која ќе ги отпечати сите четирицифрени броеви кои се деливи со збирот на двата броја составени од првите две цифри и од последните две цифри на четирицифрен број и на крај да отпечати колку вакви броеви се пронајдени.

Пример: 3417 е делив со $34+17$, 5265, 6578, ...

```
#include <stdio.h>

int zbir_cifri(int n)
{
    int zbir;
    zbir=n%100+n/100;
    if (!(n%zbir)) return 1;
    else return 0;
}

int main ()
{
    int l, vk=0;
    for (i=1000;i<10000;i++)
    {
        if (zbir_cifri(i))
        {
            printf("%d\n",i);
            vk++;
        }
    }
    printf("%d broevi go ispolnuvat uslovot.\n",vk);
    return 0;
}
```

6. Да се напише програма која за даден природен број ја пресметува разликата меѓу најблискиот поголем од него прост број и тој број.

```
#include <stdio.h>

int prost (int n)
{
    int i, x=0;
    for (i=1; i<=n; i++)
    {
        if (!(n%i)) x++;
    }
    if (x>2) return 0;
    else return 1;
}

int goren (int n)
{
    int i,x=1;
    for (i=n+1; x!=0; i++)
    {
        if (prost(i)==1) x=0;
    }
}
```

```

        return i-1;
    }

    int main ()
    {
        int broj;
        printf("Vnesi broj, n = "); scanf("%d",&broj);
        printf("%d, %d, %d\n", broj, goren(broj), goren(broj)-broj);
        return 0;
    }

```

7. Да се напише програма што ќе ги отпечати сите прости броеви помали од 1000 чиј што збир на цифри е исто така прост број. На крај да се отпечати колку вакви броеви биле пронајдени.

На пример: 23, 179, 9613, ...

Итерација & Рекурзија:

```

#include <stdio.h>

int zbir_cifri(int n)
{
    if (n<10) return n;
    else return (n%10+zbir_cifri(n/10));
}

int prost(int n)
{
    int i, x=0;
    for (i=1;i<=n;i++)
    {
        if (!(n%i)) x++;
    }
    if (x>2) return 0;
    else return 1;
}

int main ()
{
    int i;
    for (i=2;i<1000;i++)
    {
        if (prost(zbir_cifri(i))) printf("%d\n",i);
    }
    return 0;
}

```

8. Да се напише програма што од зададен интервал ќе ги отпечати сите парови прости броеви што се разликуваат меѓу себе за 2. На крај да се отпечати и нивниот број.

```
#include <stdio.h>

int prost(int n)
{
    int i, x=0;
    for (i=1;i<=n;i++)
    {
        if (!(n%i)) x++;
    }
    if (x>2) return 0;
    return 1;
}

int main ()
{
    int i, dg, gg, vk=0;
    printf("DOLNA GRANICA: "); scanf("%d",&dg);
    printf("GORNA GRANICA: "); scanf("%d",&gg);

    for (i=dg;i<=gg-2;i++)
    {
        if (prost(i) && prost(i+2))
        {
            printf("%d\t%d\n",i,i+2);
            vk++;
        }
    }
    printf("%d parovi go zadovoluvaat uslovot.\n");
    return 0;
}
```

9. Да се напише функција што прима два параметра x и n и враќа:

$$f(x) = \begin{cases} x + \frac{x^n}{n} - \frac{x^{n+2}}{n+2}, & x \geq 0 \\ -\frac{x^{n-1}}{n-1} + \frac{x^{n+1}}{n+1}, & x < 0 \end{cases}$$

Потоа да се состави програма што ќе ја табелира оваа функција за прочитано n во интервал $x \in [-4,4]$, со чекор 0.1.

```
#include <stdio.h>

float stepen(float x, int n)
{
    if (n==0) return 1;
    else return (x*stepen(x,n-1));
}

float funkcija(float x, int n)
{
    float vrednost;
```

```

        if (x<0) vrednost=-stepen(x,n-1)/(n-1)+stepen(x,n+1)/(n+1);
        else   vrednost=x+stepen(x,n)/n-stepen(x,n+2)/(n+2);
        return vrednost;
    }

    int main ()
    {
        float x;
        int n;
        printf("n = ");  scanf("%d",&n);
        for (x=-4;x<=4;x+=0.1)
        {
            if ((x==0) || (x==1) || (x==-1) || (x==-2)) break;
            else
            {
                printf("%.2f\t%.3f\n",x,funkcija(x,n));
            }
        }
        return 0;
    }

```

10. Да се напише рекурзивна функциска програма која за дадено n ќе го испише соодветниот **Фибоначиев број**. Фибоначиевите броеви се формираат на следниов начин:

$$\begin{aligned}
 a_1 &= 1 \\
 a_2 &= 1 \\
 &\dots \\
 a_n &= a_{n-1} + a_{n-2}
 \end{aligned}$$

```

#include <stdio.h>

int fibonaci (int n)
{
    if ((n==1) || (n==2)) return 1;
    else return fibonaci(n-1)+fibonaci(n-2);
}

int main ()
{
    int n;
    printf("Vnesi broj n = ");  scanf("%d",&n);
    printf("%d-tiot fibonaciev broj e %d\n",n, fibonaci(n));
    return 0;
}

```

11. Да се напише програма што ќе ја испишува вредноста на произволен член на низата дефинирана со:

$$\begin{aligned}x_1 &= 1 \\x_2 &= 2 \\&\dots \\x_n &= \frac{n-1}{n}x_{n-1} + \frac{1}{n}x_{n-2}\end{aligned}$$

```
#include <stdio.h>

float izraz (int n)
{
    if (n==1) return 1;
    if (n==2) return 2;
    return (izraz(n-1)*(n-1)/n+izraz(n-2)/n);
}

int main ()
{
    int n;
    printf("Vnesi broj n = "); scanf("%d",&n);
    printf("x_%d = %.3f\n",n,izraz(n));
    return 0;
}
```

12. Да се напише програма која што за даден опсег ќе ги наоѓа и ќе ги изброи броевите кои го задоволуваат условот да се “строго монотono растечки” или “строго монотono опаѓачки” и соодветно ќе печати порака.

Пример: 123 е строго растечки број бидејќи сите цифри се последователни строго растечки броеви, а бројот 321 е строго опаѓачки. Проверката да биде релизирана со посебни функции.

```
#include <stdio.h>

int opagja (int n)
{
    int cifra, p=n%10, vk=0;
    while (n/10)
    {
        n/=10;
        cifra=n%10;
        if (p<cifra)
        {
            p=cifra;
            vk++;
        }
        else return 0;
    }
    return vk;
}

int raste (int n)
{
    int cifra, p=n%10, vk=0;
```

```

while (n/10)
{
    n/=10;
    cifra=n%10;
    if (p>cifra)
    {
        p=cifra;
        vk++;
    }
    else return 0;
}
return vk;
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    if (raste(n)) printf("Brojot strogo monotono raste.\n");
    else
    {
        if (opagja(n)) printf("Brojot strogo monotono opagja.\n");
        else printf("Brojot ne e monotono rastecki ni opagjacki.\n");
    }
    return 0;
}

```

13. Да се напише програма која ќе пресметува синус од даден агол. Аголот се задава во степени, меѓу 0 и 90. Пресметувањето се реализира според подолу дадената формула, при што прво треба да се направи конверзија од степени во радијани. Да се имплементираат посебни функции за конверзија од степени во радијани и за пресметка на синус. Исто така да се имплементираат посебни функции за пресметување на степен и факториел.

$$\sin(x) = \sum_{i=1}^{50} (-1)^{i+1} \frac{x^{2i-1}}{(2i-1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$rad = \frac{stepeni \cdot \pi}{180}$$

Итерација & Рекурзија:

```

#include <stdio.h>

float stepen (float a, int x)
{
    if (x==0) return 1;
    else return a*stepen(a,x-1);
}

int faktoriel (int a)
{
    if (a==0) return 1;
    else return a*faktoriel(a-1);
}

```



```

float konverzija (int a)
{
    return (a*3.14)/180;
}

float izraz (int n)
{
    int i;
    float zbir=0.0;
    for (i=1;i<=15;i++)
    {
        zbir+=stepen(-1.0,i-1)*stepen(konverzija(n),2*i-1)/faktoriel(2*i-1);
    }
    return zbir;
}

int main ()
{
    int agol;
    printf("Vnesi agol vo tepeni (0<a<90) : "); scanf("%d",&agol);
    printf("\nsin(%d)=%.4f",agol, izraz(agol));
    return 0;
}

```

14. Да се напише програма која што ќе пресметува дали внесен цел број претставува полн квадрат на друг цел број. **Пример:** 25 претставува полн квадрат на бројот 5. Проверката дали бројот е полн квадрат да е реализира во посебна функција.

Упатство: Од бројот се одземаат последнователни непарни броеви се додека не се добие вредност ≤ 0 . Бројот на одземања е квадратниот корен.

Пример: $25 - 1 - 3 - 5 - 7 - 9 = 0 \rightarrow 25 = 5^2$.

```

#include <stdio.h>

int proverka(int n)
{
    int i, razlika=n, vk=0;
    for (i=1;i<=n, razlika>0;i+=2)
    {
        razlika-=i;
        vk++;
    }
    if (razlika==0) return vk;
    else return 0;
}

int main ()
{
    int br;
    printf("n = "); scanf("%d",&br);
    if (proverka(br)) printf("\n%d^2 = %d\n",proverka(br),br);
    else printf("Brojot ne e poln kvadrat!\n");
    return 0;
}

```

15. Да се напише логичка функција што како влезен аргумент прима цел број. Функцијата проверува дали производот на цифрите на бројот е прост број. Производот на цифрите на се определи во посебна рекурзивна функција.

```
#include <stdio.h>

int proizvod(int n)
{
    if (n<10) return n;
    else return (n%10*proizvod(n/10));
}

int prost (int n)
{
    int i, x=0;
    for (i=1;i<=n;i++)
    {
        if (n%i==0) x++;
    }
    if (x>2) return 0;
    else return 1;
}

int main ()
{
    int broj;
    printf("n = "); scanf("%d",&broj);
    if (prost(proizvod(broj))) printf ("Proizvodot na cifrite od %d e prost broj.\n",broj);
    else printf("Proizvodot na cifrite od %d ne e prost broj.\n",broj);
    return 0;
}
```

16. Да се напише функцииска програма која како влезни аргументи прима два цели броја. Функцијата проверува дали сумите на цифрите од двата броја се исти. Одредувањето на сумата на цифрите на бројот да се изврши во посебна рекурзивна функција.

```
#include <stdio.h>

int zbir (int n)
{
    if (n<10) return n;
    else return n%10+zbir(n/10);
}

int sporedba (int a, int b)
{
    if (zbir(a)==zbir(b)) return 1;
    return 0;
}

int main ()
{
    int a,b;
```

```

printf("a = "); scanf("%d",&a);
printf("b = "); scanf("%d",&b);
if (sporedba(a,b)) printf("Broevite %d i %d go ispolnuvat uslovot.\n",a,b);
else printf("Broevite %d i %d ne go ispolnuvat uslovot.\n",a,b);
return 0;
}

```

17. Да се напише функциска програма која од непознат број на броеви внесени од тастатура ќе ги изброи и испечати оние кои имаат непарен број на цифри.

Итерација:

```

#include <stdio.h>

int br(int n)
{
    int i;
    if (n<10) return 1;
    for (i=1;n;i++) n/=10;
    return i-1;
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    printf("%d ima %d cifri.\n",n,br(n));
    return 0;
}

```

Рекурзија:

```

#include <stdio.h>

int br(int n)
{
    if (n<10) return 1;
    return 1+br(n/10);
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    printf("%d ima %d cifri.\n",n,br(n));
    return 0;
}

```

18. Да се напише функциска програма која на внесен цел број го додава бројот запишан со истите цифри, но во обратен редослед. Потоа програмата проверува дали новиот број е палиндром и на екран печати соодветна порака.

Пример: 1. Влез: 23 → 23 + 32 = 55 → бројот е палиндром.
2. Влез: 1753 → 1753 + 3571 = 5324 → бројот не е палиндром

Итерација:

```
#include <stdio.h>

int prevrten(int n)
{
    int cifra, broj=0;
    while(n)
    {
        cifra=n%10;
        n/=10;
        broj=10*broj+cifra;
    }
}

int main ()
{
    int n, broj;
    printf("n = "); scanf("%d",&n);
    broj=n+prevrten(n);
    if (broj==prevrten(broj)) printf("Brojot e palindrom.\n");
    else printf("Brojot ne e palindrom.\n");
    return 0;
}
```

Рекурија 1:

```
#include <stdio.h>

int prevrten(int n,int p)
{
    if(n==0) return 0;
    else
    {
        return ((n%10)*p+prevrten(n/10,p/10));
    }
}

int broj(int n)
{
    if (n<10) return 1;
    return (10*broj(n/10));
}

int main ()
{
    int n, a, p;
    printf("n = "); scanf("%d",&n);
    p=broj(n);
```

```

    a=n+prevrten(n,p);
    if (a==prevrten(a,p)) printf("Brojot e palindrom.\n");
    else printf("Brojot ne e palindrom.\n");
    return 0;
}

```

Рекурзија 2:

```

#include <stdio.h>

long int prevrten(long int n)
{
    long int i=1;
    if(n==0) return 0;
    else
    {
        while(i<=n) i=i*10;
        i=i/10;
        return((n%10)*i+prevrten(n/10));
    }
}

int main ()
{
    int n, a;
    printf("n = "); scanf("%d",&n);
    a=n+prevrten(n);
    if (a==prevrten(a)) printf("Brojot e palindrom.\n");
    else printf("Brojot ne e palindrom.\n");
    return 0;
}

```

Рекурзија 3:

```

#include <stdio.h>

int prevrten(int n, int prev)
{
    prev += n%10;
    n /= 10;
    if (n != 0) prev = prevrten (n, prev*10);
    return prev;
}

int main ()
{
    int n, a;
    printf("n = "); scanf("%d",&n);
    a=n+prevrten(n,0);
    if (a==prevrten(a,0)) printf("Brojot e palindrom.\n");
    else printf("Brojot ne e palindrom.\n");
    return 0;
}

```

19. Да се напише програма која од непознат број на броеви внесени од тастатура ќе ги изброи и испечати оние кои имаат парен број на цифри. Броењето на цифрите да се реализира со посебна **рекурзивна функција**.

```
#include <stdio.h>

int broj(int n)
{
    if (n<10) return 1;
    else return 1+broj(n/10);
}

int main ()
{
    int n;
    while(scanf("%d",&n))
    {
        if (!(broj(n)%2)) printf("\t%d\n",n);
    }
    return 0;
}
```

20. Да се напише функција која како влезен аргумент прима една низа од реални броеви и големината на низата. Функцијата е потребно да ја одреди средната вредност од минималниот и максималниот елемент од низата и да ја врати истата како излезен аргумент.

```
#include <stdio.h>
#define MAX 100

float vrednost(int *a, int n) /* *a е исто со a[] */
{
    int i;
    int min=a[0], max=a[0];
    for (i=0;i<n;i++)
    {
        if (a[i]>max) max=a[i];
        if (a[i]<min) min=a[i];
    }
    return (min+max)/2.0;
}

int main ()
{
    int i, n;
    int a[MAX];
    printf("Vnesi broj na elementi: "); scanf("%d",&n);
    for (i=0;i<n;i++)
    {
        printf("a[%d] = ",i); scanf("%d",&a[i]);
    }
    printf("(min+max)/2=%.3f\n",vrednost(a,n));
    return 0;
}
```

21. Да се напише **рекурзивна функција** која ќе го определува максимумот на елементите од една низа. Низата и бројот на елементите претставуват влезни аргументи за функцијата.

```
#include <stdio.h>
#define MAX 100

int maximum(int *a, int n)
{
    int next, max=a[n];
    if (n==1) return a[1];
    else
    {
        next = maximum(a,n-1); /* Нема потреба од декларирање нова променлива */
        if (next>max) return next;
        else return max;
    }
}

int main ()
{
    int i, n;
    int a[MAX];
    printf("n= "); scanf("%d",&n);
    for (i=1;i<=n;i++) scanf("%d",&a[i]);
    printf("Br. e %d",maximum(a,n));
    return 0;
}
```

22. Да се напише **рекурзивна функција** за пресметување на n - тиот член на низата:

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 2 \\ x_3 &= 3 \\ x_n &= \frac{n-1}{n-2}x_{n-3} + \frac{n}{n-2}x_{n-2} + \frac{n}{n-1}x_{n-1} \end{aligned}$$

```
#include <stdio.h>

float x (int n)
{
    if (n == 1) return 1;
    if (n == 2) return 2;
    if (n == 3) return 3;
    return x(n-3)*(n-1)/(n-2)+x(n-2)*n/(n-2)+x(n-1)*n/(n-1);
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    printf("x_%d = %.3f",n,x(n));
    return 0;
}
```

23. Да се напише функциска програма која како параметар прима низа од n реални броеви, а како резултат ја враќа стандардната девијација дефинирана на следниот начин:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

каде x_i се елементите на низата, а \bar{x} е средната вредност на низата.

```
#include <stdio.h>
#include <math.h>
#define MAX 100

float izraz (int *a, int n)
{
    int i, zbir=0;
    float s_v, z=0;
    for (i=0;i<n;i++) zbir+=a[i];
    s_v=(float)zbir/n;
    for (i=0;i<n;i++) z+=(a[i]-s_v)*(a[i]-s_v);
    return sqrt(z/n);
}

int main ()
{
    int i, n;
    int a[MAX];
    printf("n = "); scanf("%d",&n);
    for (i=0;i<n;i++)
    {
        printf("a[%d] = ",i); scanf("%d",&a[i]);
    }
    printf("Standardnata devijacija e %.5f.\n",izraz(a,n));
    return 0;
}
```

24. Да се напише рекурзивна функција која за дадени a и n ќе пресметува:

$$p = \sum_{i=1}^n \frac{a^i}{i!}$$

```
#include <stdio.h>

int faktoriel (int n)
{
    if (n==0) return 1;
    return n*faktoriel(n-1);
}

int stepen (int a, int n)
{
    if (n==0) return 1;
    return a*stepen(a,n-1);
}

int main ()
```



```

{
    int i, n, a;
    float z, suma=0;
    printf("n = "); scanf("%d",&n);
    printf("a = "); scanf("%d",&a);
    for (i=1;i<=n;i++)
    {
        z=0;
        z=(float)stepen(a,i)/faktoriel(i);
        suma+=z;
    }
    printf("S = %.3f",suma);
    return 0;
}

```

25. Од тастатура се внесуваат непознат број цели броеви. Да се напише програма што ќе ги изброи и прикаже сите броеви кои се деливи со разликата на нивната најзначајна и најмалку значајна цифра. Определувањето на најзначајната цифра да се направи во посебна рекурзивна функција.

```

#include <stdio.h>

int prva(int n)
{
    if (n<10) return n;
    return prva(n/10);
}

int main ()
{
    int n;
    while(scanf("%d",&n))
    {
        if (n%(prva(n)-n%10)==0) printf("\t%d\n",n);
    }
    return 0;
}

```

26. Да се напише функциска програма која ја издвојува цифрата која се наоѓа на определен позиција во даден број.

```

#include <stdio.h>

int broj (int x)
{
    if (x==0) return 1;
    return 10*broj(x-1);
}

int main ()
{
    int n, k, poz;
    printf("n = "); scanf("%d",&n);
    printf("Pozicija = "); scanf("%d",&poz);
}

```

```
printf("\n%d", (n%(broj(poz)))/(broj(poz-1)));
return 0;
}
```

27. Да се напише рекурзивна функциска програма што како влезен аргумент прима цел број. Функцијата ја пресметува сумата на цифрите на еден број кои се наоѓаат на непарните позиции гледано оддесно налево.

```
#include <stdio.h>

int zbir (int n)
{
    if (n<10) return n;
    else return (n%10)+zbir(n/100);
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    printf("\n%d",zbir(n));
    return 0;
}
```

28. Да се напише рекурзивна функциска програма што како влезен аргумент прима цел број. Функцијата ја пресметува сумата на цифрите на еден број кои се наоѓаат на парните позиции гледано оддесно налево.

```
#include <stdio.h>

int zbir (int n)
{
    if (n<10) return n;
    else return (n%100)/10+zbir(n/100); /* (n/10)%10+zbir(n/100); */
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    printf("\n%d",zbir(n));
    return 0;
}
```

29. Да се напише функција што како влезни аргументи прима низа од цели броеви и број на елементи на низата. Функцијата враќа 1 ако збирот на елементите од непарните позиции е поголем од збирот на елементите од парните позиции, гледано оддесно налево. Во спротивно враќа 0.

```
#include <stdio.h>
#define MAX 100

int zbir (int *a, int n)
{
```

```

int i, zn=0, zp=0;
for (i=n;i>=1;i--)
{
    if (i%2) zn+=a[i];
    else zp+=a[i];
}
if (zn>zp) return 1;
else return 0;
}

int main ()
{
    int i, n;
    int a[MAX];
    printf("n = "); scanf("%d",&n);
    for (i=1;i<=n;i++)
    {
        printf("a[%d] = ",i); scanf("%d",&a[i]);
    }
    if (zbir(a,n)) printf("Nizata go ispolnuva uslovot.\n");
    else printf("Nizata ne go ispolnuva uslovot.\n");
    return 0;
}

```

30. Да се напише програма која како параметар прима низа од реални броеви, а како резултат го враќа елементот кој е „најблиску“ до средната вредност на елементите од низата.

```

#include <stdio.h>
#include <math.h>
#define MAX 100

float element(float *a, int n)
{
    int i;
    float zbir=0, sv, min, minel;
    for (i=1;i<=n;i++) zbir+=a[i];
    sv=(float)zbir/n;
    min=abs(a[1]-sv);
    min_el=a[1];
    for (i=2;i<=n;i++)
    {
        if ((abs(a[i]-sv))<min)
        {
            min=abs(a[i]-sv);
            min_el=a[i];
        }
    }
    return min_el;
}

int main ()
{
    int i,n;
    float a[MAX];

```

```

printf("n = "); scanf("%d",&n);
for (i=1;i<=n;i++)
{
    printf("a[%d] = ",i); scanf("%f",&a[i]);
}
printf("%.3f",element(a,n));
return 0;
}

```

31. Да се напише функциска програма која како влезен аргумент прима низа од цели броеви и број на елементи во низата. Функцијата треба да изврши трансформација на влезната низа така што елементите од втората половина ќе се ископират на местото на елементите од првата половина како во примерите. Доколку низата има непарен број на елементи, средниот елемент останува на истата позиција.

Пример 1: низата 2 8 7 6 5 3 4 се трансформира во 5 3 4 6 5 3 4

Пример 2: низата 3 5 4 2 8 1 се трансформира во 2 8 1 2 8 1

```

#include <stdio.h>
#include <math.h>
#define MAX 100

void promena(float *a, int n)
{
    int i, d=(n+1)/2;
    for (i=1;i<=n/2;i++)
    {
        a[i]=a[d+i];
    }
}

int main ()
{
    int i,n;
    float a[MAX];
    printf("n = "); scanf("%d",&n);
    for (i=1;i<=n;i++)
    {
        printf("a[%d] = ",i); scanf("%d",&a[i]);
    }
    for(i=1;i<=n;i++) printf("%d",a[i]);

    promena(a,n);

    printf("\n\n");
    for(i=1;i<=n;i++) printf("%d",a[i]);
    return 0;
}

```

32. Да се напише програма која како влезни аргументи прима низа од цели броеви и број на елементи во низата. Функцијата враќа 1 ако низата го дава истиот број читано оддлево надесно и оддесно налево. Во спротивно враќа 0.

```
#include <stdio.h>
#define MAX 100

int proverka (int *a, int n)
{
    int i;
    for (i=0;i<n/2;i++)
        if (a[i]!=a[n-1-i]) return 0;
    return 1;
}

int main ()
{
    int i, n;
    int a[MAX];
    printf("n = "); scanf("%d",&n);
    for (i=0;i<n;i++)
    {
        printf("[%d] = ",i); scanf("%d",&a[i]);
    }
    printf("%d\n",proverka(a,n));
    if (proverka(a,n)) printf("Nizata go zadovoluva uslovot.\n");
    else printf("Nizata ne go zadovoluva uslovot.\n");
    return 0;
}
```

33. Да се напише програма која како резултат на компјутерскиот екран ќе испечати дали низата е строго растечка.

Пример: 5 7 78 82 90

```
#include <stdio.h>
#define MAX 10

int proverka (int *a, int n)
{
    int i;
    for (i=0;i<n;i++)
        if (a[i]>=a[i+1]) return 0;
    return 1;
}

int main ()
{
    int i, n;
    int a[MAX];
    printf("n = "); scanf("%d",&n);
    for (i=0;i<n;i++)
    {
        printf("a[%d] = ",i); scanf("%d",&a[i]);
    }
}
```

```

    }
    if (proverka(a,n)) printf("DA");
    else printf("NE");
    return 0;
}

```

34. Од тастатура се внесуваат непознат број на цели броеви. На екран да се отпечатаат сите броеви кај кои втората цифра гледано одлево надесно е парна.

```

#include <stdio.h>

int broj (int n)
{
    if (n<10) return n;
    broj(n/10);
    if (n<100) return n%10;
}

int main ()
{
    int n;
    while(scanf("%d",&n))
    {
        if (!(broj(n)%2) && (n>9)) printf("\t%d\n",n);
    }
    return 0;
}

```

35. Да се напише функциска програма која дадена матрица ќе ја трансформира на тој начин што елементите ќе бидат подредени по растечки редослед.

```

#include <stdio.h>
#define MAX 100

void podredi (int a[][MAX], int red, int kol)
{
    int i, j, p, q, temp;

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            for (p=0; p<red; p++)
            {
                for (q=0; q<kol; q++)
                {
                    if (a[i][j]<a[p][q])
                    {
                        temp=a[i][j];
                        a[i][j]=a[p][q];
                        a[p][q]=temp;
                    }
                }
            }
        }
    }
}

```

```

    }
}

int main ()
{
    int i, j, red, kol;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("a[%d%d] = ",i,j);    scanf("%d",&a[i][j]);
        }
    }

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n\n");
    }

    printf("\n\n");
    podredi (a,red,kol);

    for (i=0; i<red; i++)
    {
        for (j=0; j<kol; j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n\n");
    }
    return 0;
}

```

36. Да се напише програма во која се внесуваат непознат број броеви. Програмата ги печати броевите кај кои збирот на првата и последната цифра е ист со збирот на втората и претпоследната.

```
#include <stdio.h>

int cifra (int n, int r)
{
    int cifra;
    while (n>r)
    {
        cifra=n%10;
        n/=10;
    }
    return cifra;
}

int main ()
{
    int n;
    while (scanf("%d",&n))
    {
        if ((cifra(n,0)+n%10) == (cifra(n,10)+(n/10)%10)) printf("\t%d\n",n);
    }
    return 0;
}
```

37. Да се напише **рекурзивна функциска програма** која за дадена низа од реални броеви и бројот на елементи n, ќе ја пресмета најголемата разлика меѓу два соседни елементи.

```
#include <stdio.h>
#include <math.h>
#define MAX 100

float max (float *a, int n)
{
    if (n==1) return a[1];
    else
    {
        float max1=fabs(a[n]-a[n-1]);
        float next=max(a,n-1);
        if (fabs(a[n-1]-next) > max1) return fabs(a[n-1]-next);
        return max1;
    }
}

int main ()
{
    float a[MAX];
    int i;
    for (i=1;i<=5;i++) scanf("%f",&a[i]);
    printf("\n\n%.3f",max(a,5));
    return 0;
}
```


38. Да се напише програма која ќе проверува дали една матрица е симетрична.

$$\begin{bmatrix} 0 & 3 & 5 & 1 \\ 3 & 7 & 6 & 9 \\ 5 & 6 & 2 & 4 \\ 1 & 9 & 4 & 2 \end{bmatrix}$$

```
#include <stdio.h>
#define MAX 100

int proverka (int a[][MAX], int red, int kol)
{
    int i,j;
    for (i=0;i<red;i++)
    {
        for (j=0;j<kol;j++)
        {
            if (a[i][j]!=a[j][i]) return 0;
        }
    }
    return 1;
}

int main ()
{
    int i, j, red, kol;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0;i<red;i++)
    {
        for (j=0;j<kol;j++)
        {
            printf("a[%d%d] = ",i,j); scanf("%d",&a[i][j]);
        }
    }

    if (proverka(a,red,kol)) printf("Matricata e simetricna.\n");
    else printf("Matricata ne e simetricna.\n");
    return 0;
}
```

39. Да се напише функциска програма која дадена матрица ќе ја пресликува огледално во однос на хоризонталата.

$$\begin{bmatrix} 2 & 3 \\ 1 & 7 \\ 0 & 5 \\ 4 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 6 \\ 0 & 5 \\ 1 & 7 \\ 2 & 3 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 5 \\ 4 & 3 & 8 \\ 9 & 0 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 9 & 0 & 6 \\ 4 & 3 & 8 \\ 1 & 2 & 5 \end{bmatrix}$$

```
#include <stdio.h>
#include <math.h>
#define MAX 100
```

```

void preslikuvanje (int a[][MAX], int red, int kol)
{
    int i, j, temp;
    int b[MAX][MAX];
    for (i=0;i<red;i++)
    {
        for (j=0;j<kol;j++)
        {
            a[i][j]=a[red-i-1][j];
        }
    }
}

int main ()
{
    int i, j, red, kol;
    int a[MAX][MAX];

    printf("Vnesi br. redovi: "); scanf("%d",&red);
    printf("Vnesi br. koloni: "); scanf("%d",&kol);

    for (i=0;i<red;i++)
    {
        for (j=0;j<kol;j++)
        {
            printf("a[%d%d] = ",i,j); scanf("%d",&a[i][j]);
        }
    }

    preslikuvanje (a,red,kol);

    for (i=0;i<red;i++)
    {
        for (j=0;j<kol;j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

40. Да се напише функција која како влезни параметри прима низа од цели броеви и број на елементи во низата, а како резултат враќа низа составена само од едноцифрени броеви подредени по растечки редослед.

```

#include <stdio.h>
#define MAX 100

void nova_niza (int a[], int b[], int n, int *p)
{
    int i, j, temp;
    for (i=1; i<=n; i++)
    {

```

```

        for (j=1; j<=n; j++)
        {
            if (a[j]>a[i])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }

    j=0;
    for (i=1; i<=n; i++)
    {
        if (!(a[i]>9))
        {
            b[j]=a[i];
            j++;
        }
    }
    *p=j-1;
}

int main ()
{
    int i, n, p;
    int a[MAX], b[MAX];
    printf("n = "); scanf("%d",&n);
    for (i=1; i<=n; i++)
    {
        printf("a[%d] = ",i);  scanf("%d",&a[i]);
    }

    nova_niza(a, b, n, &p);

    for (i=1;i<=p;i++)
    printf("%d\t",b[i]);
    return 0;
}

```

41. Да се напише функција што како влезни аргументи прима низа од цели броеви и број на елементи во низата. Функцијата треба да врати 1 ако низата содржи само парни едноцифрени брови. Во спротивно да врати нула.

Пример 1: за влезна низа 2 8 6 4 6 како резултат се добива 1

Пример 2: за влезна низа 2 5 6 4 6 како резултат се добива 0

Пример 3: за влезна низа 2 8 6 14 6 како резултат се добива 0

```

#include <stdio.h>
#define MAX 100

int proverka (int *a, int n)
{
    int i;

```

```

    for (i=0; i<n; i++)
        if ((a[i]%2)!=0 || a[i]>10) return 0;
    return 1;
}

void main ()
{
    int i, n;
    int a[MAX];
    printf("n = "); scanf("%d",&n);
    for (i=0; i<n; i++)
    {
        printf("a[%d] = ", i);   scanf("%d",&a[i]);
    }
    if (proverka(a,n)) printf("Nizata go ipolnuv uslovot.\n");
    else printf("Nizata ne go ipolnuv uslovot.\n");
}

```

42. Да се напише функциска програма која за даден интервал ќе ги испечати сите броеви кои имаат парна почетна и крајна цифра.

```

#include <stdio.h>

int prva(int n)
{
    if (n<10) return n;
    else return prva(n/10);
}

int main ()
{
    int dg, gg, i;
    printf("DOLNA GRANICA: "); scanf("%d",&dg);
    printf("GORNA GRANICA: "); scanf("%d",&gg);

    for (i=dg;i<=gg;i++)
    {
        if ((i%10) && ((i%10)%2==0) && (prva(i)%2==0)) printf("%d\n",i);
    }
    return 0;
}

```

43. Да се напише функциска програма која ќе ги печати сите прости броеви од даден интервал (почетокот и крајот на интервалот се внесуваат од тастатура), кај кои производот на последните 2 цифри е исто така прост број.

```

#include <stdio.h>

int prost(int n)
{
    int i, x=0;
    for (i=1;i<=n;i++)
    {

```

```

        if (n%i==0) x++;
    }
    if (x>2) return 0;
    else return 1;
}

int main ()
{
    int dg, gg, i, q;
    printf("DOLNA GRANICA: "); scanf("%d",&dg);
    printf("GORNA GRANICA: "); scanf("%d",&gg);

    for (i=dg;i<=gg;i++)
    {
        q=(i%10)*((i/10)%10);
        if (prost(i) && prost(q) && (q)) printf("%d\n",i);
    }
    return 0;
}

```

44. Да се напише функциска програма која ќе определи дали почетните две и крајните две цифри на еден број се еднакви.

```

#include <stdio.h>

int broj (int n)
{
    if (n<10) return n;
    else
    {
        broj(n/10);
        if (n<100) return n;
    }
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    if (n<1000) printf("Nevaliden broj.\n");
    else
    {
        if (broj(n) == (n%100)) printf("DA.\n");
        else printf("NE.\n");
    }
    return 0;
}

```

45. Да се напише програма која ќе ги печати сите прости броеви од даден интервал кои се палиндроми.

Пример: 5, 7, 11, 101, 131, 151, 181, 191, 313, 353, 373, 383, 727, 757, 787, 797, 919, 929, ...

```
#include <stdio.h>

int prost (int n)
{
    int i, x=0;
    for (i=2; i<n; i++)
    {
        if (n%i==0) return 0;
    }
    return 1;
}

int prevrten (int n)
{
    int broj=0;
    while (n)
    {
        broj=broj*10+n%10;
        n/=10;
    }
    return broj;
}

int proverka (int n)
{
    if ((prevrten(n)==n) && prost(n)) return 1;
    else return 0;
}

void main ()
{
    int i, dg, gg;
    printf("Vnesi dolna granica: "); scanf("%d",&dg);
    printf("Vnesi gorna granica: "); scanf("%d",&gg);
    for (i=dg; i<gg; i++)
    {
        if (proverka(i)) printf("%d\t",i);
    }
}
```

46. Да се напише програма која од непознат број на цели броеви кои се внесуваат преку тастатура ќе ги испише броевите кои се деливи со предходно внесениот број.

Пример: 2 42 4 16 48 96 101 202

```
#include <stdio.h>

int proverka (int n, int p)
{
    if (!(n%p)) return 1;
    else return 0;
}

int main ()
{
    int n, p;
    scanf("%d", &n);
    p=n;
    while(scanf("%d",&n))
    {
        if (proverka(n,p)) printf("\t%d\n",n);
        p=n;
    }
    return 0;
}
```

47. Да се напише програма која ќе ја пресметува вредност на изразот зададен со:

$$p = \prod_{i=1}^n \frac{\sum_{j=1}^i x^j + 2^j}{a_i}$$

каде a_i е i - тиот Фибоначиев број. Пресметката на производот, сумата и соодветните Фибоначиеви броеви да се реализира со посебни рекурзивни функции.

```
#include <stdio.h>

int fibonaci(int n)
{
    if (n==1 || n==2) return 1;
    else return fibonaci(n-1)+fibonaci(n-2);
}

int stepen (int a, int n)
{
    if (n==0) return 1;
    else return a*stepen(a, n-1);
}

int zbir(int a, int n)
{
    if (n==1) return (a+2);
    else return stepen(a, n)+(2*n)+zbir(a, n-1);
}

float proizvod(int a, int n)
{
}
```

```

        if (n==1) return (a+2);
        else return (float)(zbir(a, n)/fibonaci(n))*proizvod(a, n-1);
    }

    void main ()
    {
        int n, a;
        printf("x = "); scanf("%d",&a);
        printf("n = "); scanf("%d",&n);
        printf("%.2f",proizvod(a, n));
    }

```

48. Да се напише функциска програма која а дадено n ја пресметува вредноста на изразот

$$\sum_{i=1}^n i!$$

Да се напишат посебни рекурзивни функции за пресметување на факториел и збир.

```

#include <stdio.h>

int faktoriel(int n)
{
    if (n==0) return 1;
    else return n*faktoriel(n-1);
}

int zbir (int n)
{
    if (n==1) return 1;
    else return faktoriel(n)+zbir(n-1);
}

void main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    printf("\n%d",zbir(n));
}

```

49. Да се напише рекурзивна функција која ја пресметува сумата на првите n природни непарни броеви.

```

#include <stdio.h>

int zbir (int n)
{
    if (n==1) return 1;
    else return (2*n-1)+zbir(n-1);
}

void main ()
{
    int n;

```



```
printf("n = "); scanf("%d",&n);
printf("%d",zbir(n));
}
```

50. Да се напише програма која ќе прима еден број за влез и ќе провери дали истиот е совршен број. Совршен број е број чиј збир на делители (освен самиот тој број) е еднаков на бројот.

Пример: $6 = 1 + 2 + 3$

```
#include <stdio.h>

int sovršen (int n)
{
    int i, zbir=0;
    for (i=1; i<n; i++)
    {
        if ((n%i==0) && (i!=n)) zbir+=i;
    }
    if (zbir==n) return 1;
    else return 0;
}

int main ()
{
    int n;
    printf("n = "); scanf("%d",&n);
    if (sovršen(n)) printf("Brojot e sovršen\n");
    else printf("Brojot ne e sovršen\n");
    return 0;
}
```

51. Да се напише програма која ќе ги испечати сите непарни броеви од даден опсег кои се истовремено деливи и со збирот и со производот на нивните непарни цифри. На крај да се испечати колку такви броеви се пронајдени. За пресметување на збирот и производот на цифрите да се користат **рекурзивни функции**.

Пример: 21 е делив со 1, 111 е делив со $1 + 1 + 1$ и $1 \cdot 1$, 135, 315, 735, ...

```
#include <stdio.h>

int zbir (int n)
{
    if (n<10) return n;
    else
    {
        int cifra=n%10;
        if (cifra%2) return cifra+zbir(n/10);
        else return zbir(n/10);
    }
}

int proizvod (int n)
{
    if (n<10) return n;
```

```

    else
    {
        int cifra=n%10;
        if (cifra%2) return cifra*proizvod(n/10);
        else return proizvod(n/10);
    }
}

void main ()
{
    int dg, gg, i;
    printf("Vnesi opseg: "); scanf("%d %d",&dg, &gg);
    for (i=dg; i<=gg; i++)
    {
        if ((i%2) && !(i%zbir(i)) && !(i%proizvod(i))) printf("%d\t",i);
    }
}

```

52. Да се напише програма која ќе прикаже рачно множење на n-цифрен со m-цифрен број. (n,m<=3).

Пример:

$$\begin{array}{r}
 423 \cdot 125 \\
 \hline
 2115 \\
 8460 \\
 + 42300 \\
 \hline
 52875
 \end{array}$$

```

#include <stdio.h>

void mnozenje (int m, int n)
{
    int cifra, zbir=0, p=1;
    printf("%d*%d\n",m,n);
    printf("-----\n");
    while (n)
    {
        cifra=n%10;
        printf("%d\n",cifra*m);
        n/=10;
        zbir+=cifra*m*p;
        p*=10;
    }
    printf("-----\n");
    printf("%d\n",zbir);
}

void main ()
{
    int m, n;
    scanf("%d %d",&m, &n);
    mnozenje(m,n);
}

```

53. Да се напише програма која од даден интервал на броеви (почетокот и крајот на интервалот се задава преку тастатура) ќе го испечати на екран секој број кај кој неговата бинарна форма се состои од непарен број на единици. Пресметувањето на бројот на единици во бинарната форма на бројот да се претстави со посебна рекурзивна функција. Добивањето на бинарна форма на број се остварува со последователно пресметување на мод и количник при делење со 2, се додека не се добие 1 или 0.

Пример: $1234_{10} \rightarrow 10011010010_2$, 5 единици.

```
#include <stdio.h>

int binaren (int n)
{
    if (n==0) return 0;
    else
    {
        int cifra=n%2;
        if (cifra==1) return 1+binaren(n/2);
        else return binaren (n/2);
    }
}

void main ()
{
    int i, dg, gg, vk=0;
    printf("Dolna granica: "); scanf("%d",&dg);
    printf("Gorna granica: "); scanf("%d",&gg);

    for (i=dg; i<=gg; i++)
    {
        if (binaren(i)%2)
        {
            printf("%d\n",i);
            vk++;
        }
    }
    printf("Vkupno %d broevi.\n",vk);
}
```

IV

ТЕКСТУАЛНИ НИЗИ [СТРИНГОВИ] и ПОКАЖУВАЧИ во C

1. Да се напише функција која како параметри прима низа и проверува дали низата е валиден формат на број на мобилен телефон. Како валиден се смета форматот *+триЦифриЗаКодНаЗемја-двеЦифри-шестЦифри*.

Пример:

+389-70-111111	е валиден формат
+389-70-111a,1	не е валиден формат
070-111111	не е валиден формат

I НАЧИН:

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

int proverka (char *a)
{
    int i, x=0;
    for (i=0;i<=strlen(a)-1;i++)
    {
        if ((isdigit(*(a+i))) || (*(a+i)=='-') || (*(a+i)=='+')) x++;
        else return 0;
    }
    if ((x) && (*(a)=='+') && (*(a+4)=='-') && (*(a+7)=='-')) return 1;
    else return 0;
}

int main ()
{
    char niza[100];
    printf("Vnesi go brojot: \n");
    gets(niza);
    if ((strlen(niza))==14 && (proverka(niza)==1)) printf("Validen broj.\n");
    else printf("Nevaliden broj.\n");
    return 0;
}
```

II НАЧИН:

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>

int proverka (char *str)
{
    int i, n;
    char *a=str;
    if (*a=='+')
    {
        *a++;
        n=0;
        while(n<3)
        {
            if (isdigit(*a))
                *a++;
            else return 0;
            n++;
        }
        if (*a=='-')
        {
            n=0;
            *a++;
            while(n<2)
            {
                if (isdigit(*a))
                    *a++;
                else return 0;
                n++;
            }
        }
        else return 0;
        if (*a=='-')
        {
            *a++;
            n=0;
            while(n<6)
            {
                if(isdigit(*a))
                    *a++;
                else return 0;
                n++;
            }
        }
        else
            return 0;
        return 1;
    }
    else return 0;
}

```

```
int main ()
{
    char s[100];
    printf("Vnesi string:\n");
    gets(s);
    printf("Vneseniot string %s validen broj\n",proverka(s)?"e":"ne e");
    return 0;
}
```

2. Да се напише функцииска програма што како аргументи прима низа од знаци. Функцијата треба да изврши трансформација на влезната низа така што елементите од првата половина ќе се ископираат на местото на елементите од втората половина како во примерите. Доколку низата има непарен број елементи, средниот елемент останува на истата позиција. Задачата да се реши без употреба на средни загради „[“ и „]“.

Пример: низата *abcdefghijk* се трансформира во *abcdefabcde*
 низата *2s4fg.fgh.rf!e* се трансформира во *2s4fg.f2s4fg.f*

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void promena(char *a)
{
    int i;
    for (i=0;i<strlen(a)/2;i++)
    {
        (*(a+i+(strlen(a)/2)+strlen(a)%2))=*(a+i);
    }
}

int main ()
{
    char niza[MAX];
    printf("Vnesi niza od znaci:\n");
    gets(niza);
    promena(niza);
    printf("\nNovata niza e:\n");
    puts(niza);
    return 0;
}
```

Наместо **$(\text{strlen}(a)/2)+\text{strlen}(a)\%2$** може да се стави **$(\text{strlen}(a)+1)/2$** бидејќи го има истиот ефект. Ако бројот на елементи е непарен, тогаш количникот ќе биде чист број, а ако е парен се зема целиот дел од количникот.

3. Да се напише функциска програма која како параметри ќе прима низа од знаци и должина на низата. Функцијата треба да ги исфрли дупликатите од низата и во истата променлива од низата да ја врати новодобиениот низа.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void promena(char *a)
{
    int i, j, k, r=strlen(a);
    for (i=0; i<r; i++)
    {
        for (j=i+1; j<r; j++)
        {
            if ((*a+i)==(*a+j))
            {
                for (k=j; k<r; k++)    (*(a+k))=(*(a+k+1));
            }
        }
    }
}

int main ()
{
    char niza[MAX];
    printf("Vnesi niza od znaci:\n");
    gets(niza);
    promena(niza);
    printf("\nNovata niza e:\n");
    puts(niza);
    return 0;
}
```

4. Да се напише функциска програма што како влезен аргумент прима низа од знаци. Функцијата треба да ги замени последователните празни места со само едно празно место.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void promena(char *a)
{
    int i, j=0, r=strlen(a);
    char ch;
    for (i=0; i<r; i++)
    {
        ch=(*a+i);
        if (isspace(ch))
        {
            while (isspace((*a+i+1))) && i<r) i++;
        }
        (*(a+j))=ch;
    }
}
```

```

        j++;
    }
    (*(a+j))='\0';
}

int main ()
{
    char niza[MAX];
    printf("Vnesi niza od znaci:\n");
    gets(niza);
    promena(niza);
    printf("\nNovata niza e:\n");
    puts(niza);
    return 0;
}

```

5. Да се напише функциска програма што како влезен аргумент прима низа од знаци. Функцијата треба да ги отстрани сите броеви кои се појавуваат во низата.

```

#include <stdio.h>
#include <string.h>
#define MAX 100

void promena(char *a)
{
    int i, j=0, r=strlen(a);
    char ch;
    for (i=0; i<r; i++)
    {
        ch=*(a+i);
        if (isdigit(ch)==0)
        {
            (*(a+j))=ch;
            j++;
        }
    }
    (*(a+j))='\0';
}

int main ()
{
    char niza[MAX];
    printf("Vnesi niza od znaci:\n");
    gets(niza);
    promena(niza);
    printf("\nNovata niza e:\n");
    puts(niza);
    return 0;
}

```


6. Да се напише функциска програма што како влезен аргумент прима низа од знаци. Функцијата треба да ги замени сите мали букви со големи, а големите со мали, а останатите елементи ќе ги отстрани.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void promena(char *a, char *b)
{
    int i, j=0, r=strlen(a);
    char ch;
    for (i=0; i<r; i++)
    {
        ch=*(a+i);
        if (isdigit(ch)==0)
        {
            if (isupper(ch)) ch=tolower(ch);
            else if (islower(ch)) ch=toupper(ch);
            *(b+j)=ch;
            j++;
        }
    }
    *(b+j)='\0';
}

int main ()
{
    char niza[MAX], niza1[MAX];
    printf("Vnesi niza od znaci:\n");
    gets(niza);
    promena(niza, niza1);
    printf("\nNovata niza e:\n");
    puts(niza1);
    return 0;
}
```

7. Да се напише функциска програма која како влезни аргументи прима две низи од знаци. Ако низите содржат различен број на знаци, функцијата треба да врати -1. Ако бројот на знаци во двете низи е еднаков, тогаш функцијата треба да го врати бројот на знаци во кои низите се разликуваат. Задачата да се реши со помош на покажувачи.

Пример: *abcdef ghiiklmn opqrstu*
 abccef ghijklmn ppqrstu
 функцијата ќе врати 3.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

int promena(char *a, char *b)
{
    int i, j, r=strlen(a), q=strlen(b), x=0;
    if (r!=q) return -1;
```

```

else
{
    for (i=0; i<r; i++)
    {
        if ((* (a+i))!=(* (b+i))) x++;
    }
}
if (x!=0) return x;
else return 0;
}

int main ()
{
    char niza[MAX], niza1[MAX];
    printf("Vnesi niza od znaci:\n");
    gets(niza);
    printf("Vnesi niza od znaci:\n");
    gets(niza1);

    if (promena(niza, niza1)==-1) printf("Nizite ne se so ista dolzina.\n");
    if (promena(niza, niza1)==0) printf("Nizite se identicni.\n");
    if (promena(niza, niza1) && promena(niza, niza1)!=-1)
        printf("Nizite se razlikuvaat vo %d elementi.\n",promena(niza, niza1));
    return 0;
}

```

8. Да се напише функциска програма која ќе прими како аргумент две низи од знаци. Функцијата треба од првата низа да ги отфрли сите знаци кои се сретнуваат во втората низа. Функцијата исто така треба да го врати бројот на отфрлени знаци.

Пример: прва низа: 1abc412acd, втора низа: 1b211

Излез: прва низа променета: ac4acd, број на исфрлени знаци: 4.

```

#include <stdio.h>
#include <string.h>
#define MAX 100

int promena(char *a, char *b)
{
    int i, j, r=strlen(a), q=strlen(b), k=0, x;
    for (i=0; i<r; i++)
    {
        x=0;
        for (j=0; j<q; j++)
        {
            if ((* (a+i))==(* (b+j))) x++;
        }
        if (x==0)
        {
            (*(a+k))=(* (a+i));
            k++;
        }
    }
    (*(a+k))='\0';
}

```

```

        return r-k;
    }

    int main ()
    {
        char niza[MAX], niza1[MAX];

        printf("Vnesi 1 niza od znaci:\n");
        gets(niza);
        printf("Vnesi 2 niza od znaci:\n");
        gets(niza1);
        printf("\n%d isfreni znaci.\n",promena (niza, niza1));
        printf("Nova niza:\n");
        puts(niza);
        return 0;
    }

```

9. Да се напише програма што ќе ја прочита содржината на една текстуална низа што завршува со точка ('.'). Читањето се врши знак по знак. На компјутерскиот екран да се прикажат само цифрите од прочитаната низа. Задачата да се реши без користење на низа од знаци (string).

```

#include <stdio.h>

int main ()
{
    char c;
    while ((c=getchar())!=EOF)
    {
        if (c=='.') break;
        else if (isdigit(c)) putchar(c);
    }
    putchar('\n');
    return 0;
}

```

10. Да се напише функцииска програма што како влезни аргументи прима низа од знаци. Функцијата враќа 1 ако низата го дава истиот стринг читано одлево надесно и одесно налево, во спротивно враќа 0.

Пример: adv37465.?fr@rf?.56473vda

```

#include <stdio.h>
#include <string.h>
#define MAX 100

int proverka (char *a)
{
    int i, n=strlen(a);
    for (i=0; i<n/2; i++)
    {
        if ((*a+i)!=(*a+n-i-1)) return 0;
    }
    return 1;
}

```

```

void main ()
{
    char niza[MAX];
    printf("Vnesi niza:\n");
    gets(niza);
    if (proverka(niza)) printf("Nizata e palindrom.\n");
    else printf("Nizata ne e palindrom.\n");
}

```

11. Да се напише функциска програма што како влезни аргументи прима низа од броеви и број на елементи во низата. Функцијата враќа 1 ако збирот на елементите од непарните позиции е поголем од збирот на елементите од парните позиции, гледано оддесно налево, а во спротивно враќа 0.

Пример: 2, 4, 8, 7, 5, 7, 8, 4, 2

$$2 + 8 + 5 + 8 + 2 > 4 + 7 + 7 + 4$$

```

#include <stdio.h>
#define MAX 100

int proverka (int *a, int n)
{
    int i, zbir_par=0, zbir_nep=0;
    for (i=0; i<n; i++)
    {
        if (i%2) zbir_par+=*(a+n-1-i);
        else zbir_nep+=*(a+n-1-i);
    }
    return (zbir_nep>zbir_par);
}

int main ()
{
    int i, n, niza[MAX];
    printf("Vnesi br. na elementi: "); scanf("%d",&n);
    for (i=0; i<n; i++)
    {
        printf("a[%d] = ",i);  scanf("%d",&niza[i]);
    }
    if (proverka(niza,n)) printf("DA.\n");
    else printf("NE.\n");
    return 0;
}

```

12. Да се напише функциска програма што како влезен аргумент прима низа од знаци. Функцијата треба да изврши трансформација на влезната низа така што елементите од втората половина ќе се ископираат на местото на елементите од првата половина како во примерите. Доколку низата има непарен број на елементи, средниот елемент останува на истата позиција.


Пример: низата 2 8 7 6 5 3 4 се трансформира во 5 3 4 6 5 3 4
 низата 3 5 4 2 8 1 се трансформира во 2 8 1 2 8 1

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void promena(char *a)
{
    int i, n=strlen(a), r=(n+1)/2;
    for (i=0;i<n/2;i++)
    {
        (*(a+i))=(*(a+i+r));
    }
}

int main ()
{
    char niza[MAX];
    printf("Vnesi niza od znaci:\n");
    gets(niza);
    promena(niza);
    printf("\nNovata niza e:\n");
    puts(niza);
    return 0;
}
```

13. Да се напише функциска програма која прима низа од броеви. Функцијата треба огледално да ги додаде елементите од втората половина на елементите од првата половина.

Пример: 1 2 3 4 5 6 → g g g 4 5 6


Напомена: програмата ја зема ASCII вредноста на елементите.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void promena(char *a)
{
    int i, n=strlen(a);
    for (i=0;i<n/2;i++)
    {
        (*(a+i))+=(*(a+n-i-1));
    }
}

int main ()
{
    int char[MAX];
```

```
printf("Vnesi niza od znaci:\n");
gets(niza);
promena(niza);
printf("\nNovata niza e:\n");
puts(niza);
return 0;
}
```

14. Да се напише програма што ќе изброи колку елементи од дадена низа се разликуваат од 0, се додека не се пронајде првата нула. Низата сигурно содржи 0.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

int search(char *a)
{
    int i, n=strlen(a), x=0;
    for (i=0; i<n; i++)
    {
        if (*(a+i)=='0') return x;
        x++;
    }
    return 0;
}

void main ()
{
    char niza[MAX];
    printf("Vnesi niza: \n");
    gets(niza);
    printf("Do prvata pojava, %d elementi se razlikuvat od 0.\n",search(niza));
}
```

15. Да се напише функција што како влезен аргумент прима низа од знаци. Функцијата треба да ги замени последователните исти цифри со една цифра.

Пример: Низата „Mozni ocenki na ispitot se 6, 77, 888, 9, 10.“

ќе ја трансформира во низа: „Mozni ocenki na ispitot se 6, 7, 8, 9, 10.“

```
#include <stdio.h>
#include <string.h>
#define MAX 100

int search(char *a)
{
    int i, n=strlen(a), j=0;
    char ch;
    for (i=0; i<n; i++)
    {
        ch=*(a+i);
        if (isdigit(ch))
        {
            while (*(a+i+1)==ch && i<n-1) i++;
        }
    }
}
```

```

        *(a+j)=ch;
        j++;
    }
    else
    {
        *(a+j)=ch;
        j++;
    }
}
*(a+j)='\0';
}

```

```

void main ()
{
    char niza[MAX];
    printf("Vnesi niza: \n");
    gets(niza);
    printf("Nova niza:\n");
    search(niza);
    puts(niza);
}

```

16. Да се напише функциска програма која како влезни аргументи прима една низа од симболи и два дополнителни симболи. Функцијата треба да врати бројот на појавувања на двата симболи едно до друго во влезната низа. Задачата да се реши само со покажувачи. Да не се прави разлика меѓу големи и мали букви.

Пример: *func("otkako studentot otide vo citalna, nauci", 'o', 't');* ќе врати резултат 3.

```

#include <stdio.h>
#include <string.h>
#define MAX 100

int x(char *a, char znak1, char znak2)
{
    int i, vozbor=0, n=strlen(a), vk=0;
    for (i=0; i<n; i++)
    {
        if (toupper(*(a+i))== toupper(znak1) && toupper(*(a+i+1))== toupper(znak2)) vk++;
    }
    return vk;
}

int main ()
{
    char niza[MAX], znak1, znak2;
    printf("Vnesi niza: \n");
    gets(niza);
    printf("Znaci za proverka: "); scanf("%c %c",&znak1, &znak2);
    printf("%d pojavi.\n",x(niza, znak1, znak2));
    return 0;
}

```

17. Да се напише функција:

`void zamena (char* niza, char znak1, char znak2)`

Функцијата треба во низата секоја појава на знакот *znak1* да го замени со знакот *znak2*.

Пример: *niza: ova e primer za niza, znak1: a, znak: A*

Излез: ovA e primer zA niza.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void zamena (char *a, char z1, char z2)
{
    int i, n=strlen(a);
    for (i=0; i<n; i++)
    {
        if (*(a+i)==z1) *(a+i)=z2;
        else
        {
            if (*(a+i)==z2) *(a+i)=z1;
        }
    }
}

int main ()
{
    char niza[MAX], znak1, znak2;
    printf("Vnesi niza:\n");
    gets(niza);
    printf("Vnesi znaci za zamena: ");
    scanf("%c %c", &znak1, &znak2);
    zamena(niza, znak1, znak2);
    printf("Nova niza:\n");
    puts(niza);
    return 0;
}
```

18. Да се напише функција која како влезен аргумент прима низа од знаци. Функцијата треба да го врати бројот на зборови во низата кои завршуваат со буквата „i“ или „I“. Зборовите во низата од знаци се составени од алфанумерички знаци и меѓусебно се разделени со едно или повеќе празни места и (или) интерпункциски знаци.

```
#include <stdio.h>
#include <string.h>
#define MAX 100
int x(char *a)
{
    int i, vozbor=0, n=strlen(a), vk=0, posledna;
    for (i=0; i<n; i++)
    {
        if (isalnum(*(a+i)))
        {
            if (!(vozbor)) vozbor=1;
            posledna=*(a+i);
        }
    }
}
```



```

    }
    else if (vozbor)
    {
        if (toupper(posledna)=='l') vk++;
        vozbor=0;
    }
}
return vk;
}

int main ()
{
    char niza[MAX];
    printf("Vnesi niza: \n");
    gets(niza);
    printf("%d pojavi na i ili l.\n",x(niza));
    return 0;
}

```

19. Да се напише функција која како влезен аргумент прима низа од знаци. Функцијата треба да го врати бројот на зборови во низата кои започнуваат со буквата „а“ или „А“. Зборовите во низата од знаци се составени од алфанумерички знаци и меѓусебно се разделени со едно или повеќе празни места и (или) интерпункциски знаци.

```

#include <stdio.h>
#include <string.h>
#define MAX 100
int x(char *a)
{
    int i, vozbor=0, n=strlen(a), vk=0;
    for (i=0; i<n; i++)
    {
        if (isalnum(*(a+i)))
        {
            if (!(vozbor))
            {
                vozbor=1;
                if ((toupper(*(a+i)))=='A') vk++;
            }
        }
        else if (vozbor) vozbor=0;
    }
    return vk;
}

int main ()
{
    char niza[MAX];
    printf("Vnesi niza: \n");
    gets(niza);
    printf("%d pojavi na a ili A.\n",x(niza));
    return 0;
}

```

20. Да се напише функциска програма која за дадена низа од знаци ќе го одреди бројот на зборови. Зборовите во низата од знаци се составени од алфанумерички знаци и меѓусебно се разделени со едно или повеќе празни места и (или) интерпункциски знаци.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

int x(char *a)
{
    int i, vozbor=0, n=strlen(a), vk=0;
    for (i=0; i<n; i++)
    {
        if (isalnum(*(a+i)))
        {
            if (!(vozbor)) vozbor=1;
        }
        else if (vozbor)
        {
            vozbor=0;
            vk++;
        }
    }
    if (vozbor) vk++;
    return vk;
}

int main ()
{
    char niza[MAX];
    printf("Vnesi niza: \n");
    gets(niza);
    printf("%d zborovi.\n",x(niza));
    return 0;
}
```

21. Да се напише функција која прима две низи од знаци и еден цел број. Бројот претставува позиција од првата низа на која се надоврзува втората низа во целина. По надоврзувањето првата низа ги продолжува од прекилот.

Пример: *niza1: Da se uci programski e lesno.*
 niza2: ne_ pozicija: 21
 излезе: Da se uci programski ne e lesno.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void promena(char *a, char *b, char *c, int poz)
{
    int i, j=0, n=strlen(a), m=strlen(b);
    for (i=0; i<poz; i++)
    {
        *(c+j)=*(a+i);
    }
```

```

        j++;
    }
    i=0;
    for (i=0; i<m; i++)
    {
        *(c+j)=*(b+i);
        j++;
    }
    i=0;
    for (i=poz; i<n; i++)
    {
        *(c+j)=*(a+i);
        j++;
    }
    *(c+j)='\0';
}

int main ()
{
    char niza1[MAX], niza2[MAX], niza3[MAX], poz;
    printf("Vnesi niza 1:\n");
    gets(niza1);
    printf("Vnesi niza 2:\n");
    gets(niza2);
    printf("Vnesi pozicija: "); scanf("%d",&poz);
    promena (niza1, niza2, niza3, poz);
    printf("Nova niza:\n");
    puts(niza3);
    return 0;
}

```

22. Да се напише функција која од дадена низа знаци ќе ги исфрли знаците почнувајќи од n-тиот во должина од k знаци.

```

#include <stdio.h>
#include <string.h>
#define MAX 100

int zamena (char *a, char *b, int k, int n)
{
    int i, len=strlen(a), j=0;
    for (i=0; i<len; i++)
    {
        if (i==(k-1))
        {
            while (i<n+k-1) i++;
        }
        *(b+j)=*(a+i);
        j++;
    }
    *(b+j)='\0';
}

void main ()

```

```

{
    char niza[MAX], niza1[MAX];
    int k, n;
    printf("Vnesi niza:\n");
    gets(niza);
    printf("Vnesi pozicii: "); scanf("%d %d", &k, &n);
    zamena (niza, niza1, k, n);
    printf("Nova niza:\n");
    puts(niza1);
}

```

23. Да се напише функциска програма што ќе одредува колку пати еден знак се наоѓа во даден стринг. Знакот за споредување и стрингот се внесуваат од тастатура.

```

#include <stdio.h>
#include <string.h>
#define MAX 100

int search (char *a, char znak)
{
    int i, vk=0, len=strlen(a);
    for (i=0; i<len; i++)
    {
        if (*(a+i)==znak) vk++;
    }
    return vk;
}

void main ()
{
    char niza[MAX], znak;
    printf("Vnesi niza:\n");
    gets(niza);
    printf("Vnesi znak za pebaruvanje: "); scanf("%c",&znak);
    printf("Znakot %c se pojavuva %d pati.\n",znak,search (niza,znak));
}

```

24. Да се напише функциска програма која ќе ги отстранува бланко знаците од кајот на низата.

```

#include <stdio.h>
#include <string.h>
#define MAX 100

void promena(char *a)
{
    int i, j=0, n=strlen(a);
    for (i=n-1; i>0; i--)
    {
        while (isspace(*(a+n-i)) && i>0) i--;
    }
    *(a+i)='\0';
}

```

```

int main ()
{
    char niza1[MAX];
    printf("Vnesi niza:\n");
    gets(niza1);
    promena (niza1);
    printf("Nova niza:\n");
    puts(niza1);
    return 0;
}

```

25. Да се напише функција која ќе враќа поднiza од зададена текстуална низа определена од позицијата и должината што како параметри се внесуват од тастатура. Поднизата започнува од карактерот што се наоѓа на соодветната позиција во текстуалната низа броено од лево.

```

#include <stdio.h>
#include <string.h>
#define MAX 100

int promena (char *a, int k, int n)
{
    int i, j=0, len=strlen(a);
    for (i=k-1; i<n+k-1; i++)
    {
        *(a+j)=*(a+i);
        j++;
    }
    *(a+j)='\0';
}

void main ()
{
    char niza[MAX];
    int k, n;
    printf("Vnesi niza:\n");
    gets(niza);
    printf("Vnesi pozicija: "); scanf("%d",&k);
    printf("Vnesi dolzina: "); scanf("%d",&n);
    promena(niza,k,n);
    printf("Nova niza:\n");
    puts(niza);
}

```

26. Да се напише програма која за дадена низа од знаци ќе провери дали низата е палиндром. Од внесена низа, пред проверката дали е палиндром, да се исфрлат сите празни места и да не се прави разлика помеѓу мала и голема буква.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

int palindrom (char *a)
{
    int i, j=0, len=strlen(a), len1;
    for (i=0; i<len; i++)
    {
        while (isspace(*(a+i)) && i<len) i++;
        *(a+j)=*(a+i);
        j++;
    }
    *(a+j)='\0';
    len1=j;

    for (j=0; j<len1; j++)
    {
        if (*(a+j)!=*(a+len1-1-j)) return 0;
        return 1;
    }
}

void main ()
{
    char niza[MAX];
    printf("Vnesi niza:\n");
    gets(niza);
    if (palindrom(niza)) printf("DA.\n");
    else printf("NE.\n");
}
```

27. Да се напише процедура која влезната низа a_0, a_1, \dots, a_{n-1} ќе ја трансформира во излезна низа b_0, b_1, \dots, b_{n-1} на следниот начин:

$$\begin{aligned} b_0 &= a_0 + a_{n-1} \\ b_1 &= a_1 + a_{n-2} \\ &\dots \\ b_{n-1} &= a_{n-1} + a_0 \end{aligned}$$

```
#include <stdio.h>
#define MAX 100

int promena (int a[], int b[], int n)
{
    int i;
    for (i=0; i<n; i++) *(b+i)=*(a+i)+*(a+n-i-1);
}
```

```

int main ()
{
    int n, i, a[MAX], b[MAX];
    printf("n = "); scanf("%d",&n);
    for (i=0; i<n; i++)
    {
        printf("a[%d] = ",i); scanf("%d",&a[i]);
    }
    promena(a,b,n);
    for (i=0;i<n; i++)
    {
        printf("%d\t",*(b+i));
    }
    return 0;
}

```

28. Да се напише програма која во низа од N цели броеви ќе го пронајде почетокот и должината на најдолгата растечка подниза.

```

#include <stdio.h>
#define MAX 100

void proverka (int *a, int n, int *len, int *pos)
{
    int i, dolz, poc;
    *pos=0; *len=1;
    i=0;
    while(i<n-1)
    {
        poc=i;
        dolz=1;
        while (*(a+i)<*(a+i+1))
        {
            dolz++;
            i++;
        }
        if (dolz>*len)
        {
            *len=dolz;
            *pos=poc;
        }
        i++;
    }
}

int main ()
{
    int i, n, a[MAX], pos, len;
    printf("n = "); scanf("%d",&n);
    for (i=0; i<n; i++)
    {
        printf("a[%d] = ",i);
        scanf("%d",&a[i]);
    }
}

```

```

    proverka(a, n, &len, &pos);
    printf("Pocetok: %d, dolzina: %d.\n", pos, len);
    return 0;
}

```

29. Да се напише функциска програма која од две текстуални низи кои се внесуваат преку тастатура, секоја појава на ист член на иста позиција во низите ќе го замени со *.

Пример:

влез:	<i>Jas sum prva niza</i>
	<i>Zadaca1 po programiranje</i>
излез:	<i>J*s sum**rva niz*</i>
	<i>Z*daca1**o progr*miranje</i>

```

#include <stdio.h>
#include <string.h>
#define MAX 80

void promena (char *a, char *b)
{
    int i;
    for (i=0; i<strlen(a); i++)
    {
        if (*(a+i)==*(b+i))
        {
            *(a+i)='*';
            *(b+i)='*';
        }
    }
}

int main ()
{
    char a[MAX], b[MAX];
    printf("Niza1:\n");
    gets(a);
    printf("Niza2:\n");
    gets(b);
    promena (a,b);
    puts(a);
    puts(b);
    return 0;
}

```


30. Во дадена низа од n природни броеви (n и низата се читаат од тастатура) броевите кои се наоѓаат на позиција $2i$ и $2i+1$ се непарни да си ги заменат местата (позициите). Потоа да се испечати изменетата низа. Замената да се реализира во посебна функција.

Пример: 1, 5, 13, 8, 11, 65, 18, 27, 19, 21, 31, 36, 22, 7, 17, 84
 5, 1, 13, 8, 65, 11, 18, 19, 27, 31, 21, 36, 22, 17, 7, 84

```
#include <stdio.h>
#include <string.h>
#define MAX 100

void promena (int *a, int n)
{
    int i, temp;
    for (i=0; i<(n+1)/2; i++)
    {
        if ((*a+2*i)%2 && *(a+2*i+1)%2)
        {
            temp=*(a+2*i);
            *(a+2*i)=*(a+2*i+1);
            *(a+2*i+1)=temp;
        }
    }
}

int main ()
{
    int i, n, a[MAX];
    printf("n = "); scanf("%d",&n);
    for (i=0; i<n; i++) scanf("%d",&a[i]);
    promena(a,n);
    for (i=0; i<n; i++) printf("%d\t",a[i]);
    return 0;
}
```

31. Да се напише програма која дадена текстуална низа ќе ја поправи граматички во однос на точка и голема буква после неа. Правилото кое треба да се запази е после секоја точка да има точно едно празно место и наредната реченица да започне со голема буква.

Пример: Zadaca so stringovi.ispit po programiranje. mnogu lesno.
 Zadaca so stringovi. Ispit po programiranje. Mnogu lesno.

```
#include <stdio.h>
#include <string.h>
#define MAX 100

int search(char *a, char *b)
{
    int i, n=strlen(a), j=0;
    for (i=0; i<n; i++)
    {
        if (*a=='.')
        {
            *b=*a;
            *a++;
        }
    }
}
```

```
        *b++;
        if (*a==' ')
        {
            *b=*a;
            *a++;
            *b++;
            *b=toupper(*a);
            *b++;
            *a++;
        }
        else if (*a!=' ')
        {
            *b=' ';
            *b++;
            *b=toupper(*a);
            *b++;
            *a++;
        }
    }
    else
    {
        *b=*a;
        *a++;
        *b++;
    }
}
*b='\0';
}

void main ()
{
    char niza[MAX], niza1[MAX];
    printf("Vnesi niza: \n");
    gets(niza);
    printf("Nova niza:\n");
    search(niza, niza1);
    puts(niza1);
}
```

V

ТЕКСТУАЛНИ ДАТОТЕКИ во C

1. Да се напише програма која за дадена текстуална датотека "text.txt" ќе ги изброи зборовите. Зборовите се составени од алфанумерички знаци и се одделени со најмалку едно празно место или знаци за интерпункција.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main ()
{
    char c;
    int br_zborovi, vo_zbor;
    FILE *dat;
    br_zborovi=0, vo_zbor=0;

    if ((dat=fopen("text.txt","r"))==NULL)
    {
        printf("Datotekata text.txt ne moze da se otvori.\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalnum(c))
        {
            if (!vo_zbor) vo_zbor=1;
        }
        else
        {
            if (vo_zbor)
            {
                vo_zbor=0;
                br_zborovi++;
            }
        }
    }
}
```

```

    }
}
if (vo_zbor)
{
    br_zborovi++;
    vo_zbor=0;
}
fclose(dat);
printf("Vкупно %d zborovi.\n",br_zborovi);
return 0;
}

```

2. Да се напише програма која за дадена текстуална датотека чие што име се внесува од командната линија, ќе го одреди односот на самогласки и согласки.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int samoglaska (char c)
{
    c=toupper(c);
    if (c=='A' || c=='E' || c=='I' || c=='O' || c=='U') return 1;
    else return 0;
}

int main (int argc, char* argv[])
{
    char c;
    int br_samoglaski=0, br_soglaski=0;
    float br;
    FILE *dat;

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("Datotekata %d ne moze da se otvori.\n",argv[1]);
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalpha(c))
        {
            if (samoglaska(c)) br_samoglaski++;
            else br_soglaski++;
        }
    }
    br=(float)br_samoglaski/br_soglaski;
    fclose(dat);
    printf("Odnosot e %d/%d = %5.2f\n", br_samoglaski, br_soglaski, br);
    return 0;
}

```

3. Да се напише програма која за дадена текстуална датотека ќе изброи колку зборови подолги од 3 букви почнуваат и завршуваат на иста буква. Да не е прави разлика меѓу голема и мала буква. Зборовите се составени од произволен број на букви, а меѓусебно се одделени со најмалку еден специјален знак, цифра или белина. Името на влезната датотека се задава од командна линија, а доколку не е зададено се чита од стандардниот влез.

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main (int argc, char* argv[])
{
    char c, prva, posledna;
    int vkupno=0, vo_zbor=0, dolzina=0;
    FILE *dat;
    if (argc>2)
    {
        printf("Using: %s name_of_file\n",argv[0]);
        return (-1);
    }
    if (argc==1)  dat=stdin;
    else if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("Datotekata %s ne moze da se otvori.\n",argv[1]);
        return (0);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalpha(c))
        {
            if (!vo_zbor)
            {
                vo_zbor=1;
                prva=c;
            }
            dolzina++;
            posledna=c;
        }
        else
        if (vo_zbor)
        {
            vo_zbor=0;
            if (dolzina>3 && toupper(prva)==toupper(posledna)) vkupno++;
            dolzina=0;
        }
    }
    if (vo_zbor)
    {
        if (dolzina>3 && toupper(prva)==toupper(posledna)) vkupno++;
    }
    printf("%d zborovi go ispolnuvat uslovot.\n",vkupno);
    return 0;
}
```

4. За дадена датотека, чие што име се задава од командна линија да се испечати вкупниот број на зборови во секој ред кои почнуваат и завршуваат со иста буква. Да се провери дали името на датотеката е зададено преку командна линија.

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

int main (int argc, char* argv[])
{
    char c, prva, posledna;
    int vkupno=0, vo_zbor=0, red=1;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Wrong parameter\n");
        printf("Type name of file: ");
        dat=stdin;
    }

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("File cannot be opened.\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalpha(c))
        {
            if (!vo_zbor)
            {
                vo_zbor=1;
                prva=c;
            }
            posledna=c;
        }
        else
        if (vo_zbor)
        {
            vo_zbor=0;
            if (toupper(prva)==toupper(posledna)) vkupno++;
        }
        if (c=='\n')
        {
            printf("red %d\tzborovi %d\n",red, vkupno);
            vkupno=0;
            vo_zbor=0;
            red++;
        }
    }
    if (vo_zbor)
```

```

    {
        vo_zbor=0;
        if (toupper(prva)==toupper(posledna)) vkupno++;
        printf("red %d\tzborovi %d\n",red, vkupno);
    }
    return 0;
}

```

5. Да се напише програма која за дадена датотека чие име се задава во командната линија преку тастатура ќе го испечати вкупниот број на знаци без празните места. Доколку името на датотеката не е зададено преку командна линија, на екран да се испечати информација за грешка.

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>

int main (int argc, char* argv[])
{
    char c;
    int znak=0, red=1;
    FILE *dat;
    if (argc!=2)
    {
        printf("ERROR: invalid parameter.\n");
        return (-1);
    }
    else if (argc==2) printf("Using: %s name_of_file\n",argv[0]);

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file %s cannot be opened!\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (c!='\n')
        {
            if (!isspace(c)) znak++;
        }
        else
        {
            if (c=='\n')
            {
                printf("%d red: %d znaci.\n",red, znak);
                znak=0;
                red++;
            }
        }
    }
    return 0;
}

```

6. Да се напише програма која од датотека чие име се внесува преку командна линија ќе изброи колку зборови има во секој ред. Зборовите се составени од алфанумерички знаци и меѓусебно се одделени со бланко или интерпункциски знак. Ако името на датотеката не е зададено преку командна линија да се прочита од стандардниот влез, а доколку се внесат повеќе параметри да се сигнализира грешка при повикување на датотека.

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

int main (int argc, char* argv[])
{
    char c;
    int vozbor=0, zbor=0, red=1;
    FILE *dat;

    if (argc==1)
    {
        printf("ERROR: Invalid parameter\n");
        printf("Type name of file: ");
        dat=stdin;
    }
    else if (argc==2)
    {
        printf("Using: %s name_of_file\n",argv[1]);
    }
    else if (argc>2)
    {
        printf("ERROT: Too much parameters\n");
        return (-1);
    }

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file %s cannot be opened\n",argv[1]);
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalnum(c))
        {
            if (!vozbor) vozbor=1;
        }
        else if (vozbor)
        {
            vozbor=0;
            zbor++;
        }
        if (c=="\n")
        {
            vozbor=0;
            printf("red %d : zborovi %d\n",red, zbor);
        }
    }
}
```



```

        zbor=0;
        red++;
    }
}
if (vozbor)
{
    zbor++;
    printf("red %d : zborovi %d\n",red, zbor);
}
return 0;
}

```

7. Да се напише програма која во датотека ќе ги отстрани сите празни места, а редовите ќе ги задржи. Името на датотеката се задава во командната линија, а ако не е зададено таму да сигнализира грешка и да се внесе името од стандардниот влез.

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

int main (int argc, char* argv[])
{
    char c;
    int k=1;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Invalid parameter\n");
        printf("Type name of file: ");
        dat=stdin;
    }

    if (argc==2) printf("Using: %s name_of_file\n",argv[1]);

    if ((dat=fopen(argv[1], "r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (!isspace(c)) putchar(c);
        if (c=='\n') putchar('\n');
    }
    return 0;
}

```

8. Да се напише програма која како параметар ќе го прима името на некоја датотека и ќе го отпечати бројот на редови во кои има по најмалку десет зборови. Зборовите се составени само од букви. Во секој ред има најмногу 80 знаци.

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

int main (int argc, char* argv[])
{
    char c;
    int vo_zbor, zborovi=0, vkupno=0, red=1;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Wrong parameters\n");
        printf("Type name of file: ");
        dat=stdin;
    }

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("File cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalpha(c))
        {
            if(!vo_zbor) vo_zbor=1;
        }
        else if (vo_zbor)
        {
            vo_zbor=0;
            zborovi++;
        }
        if (c=='\n')
        {
            if (zborovi>=10)
            {
                vkupno++;
                printf("red %d\t%d zborovi\n",red, zborovi);
            }
            vo_zbor=0;
            zborovi=0;
            red++;
        }
    }
    if (vo_zbor)
    {
        zborovi++;
    }
}
```

```

        if (zborovi>=10)
        {
            vkupno++;
            printf("red %\t%d zborovi\n",red, zborovi);
        }
    }
    printf("Vkupno %d redovi.\n",vkupno);
    return 0;
}

```

9. Да напише програма која за дадена текстуална датотека ќе ги испечати сите зборови во кои се јавуваат повеќе од две исти букви. Да не се прави разлика меѓу големи и мали букви. На кајот да се отпечати колку такви зборови биле пронајдени. Зборовите се составени само од букви, а меѓусебно се одделени со најмалку еден специјален знак, цифра или белина. Се смета дека во текстот нема зборови подолги од 20 знаци. Името на датотеката се задава од командна линија.

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#define MAX 21

int proverka (char *c)
{
    int i, j, vk=0;
    for (i=0; i<strlen(c); i++)
    {
        vk=0;
        for (j=i; j<strlen(c); j++)
        {
            if (toupper(*(c+i))==toupper(*(c+j))) vk++;
        }
        if (vk>2) return 1;
    }
    return 0;
}

int main (int argc, char* argv[])
{
    char c, zbor[MAX];
    int vo_zbor=0, brojac=0, vkupno=0;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Invalid prameter\n");
        printf("Type name of file: ");
        dat=stdin;
    }

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: File cannot opened\n");
    }
}

```

```

        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalpha(c))
        {
            if (!vo_zbor)
            {
                vo_zbor=1;
            }
            zbor[brojac]=c;
            brojac++;
        }
        else
        if (vo_zbor)
        {
            vo_zbor=0;
            zbor[brojac]='\0';
            if (proverka(zbor))
            {
                puts(zbor);
                vkupno++;
            }
            zbor[0]='\0';
            brojac=0;
        }
    }
    if (vo_zbor)
    {
        zbor[brojac]='\0';
        if (proverka(zbor))
        {
            puts(zbor);
            vkupno++;
        }
    }
    printf("Vkupno %d zborovi.\n",vkupno);
    return 0;
}

```

10. Да се напише програма која за дадена текстуална датотека ќе изброи и отпечати зборови подолги од четири букви кои почнуваат и завршуваат на самогласка. Да не се прави разлика меѓу големи и мали букви. Зборовите се составени само од букви, а меѓусебно се одвоени со најмалку еден специјален знак, цифра или празно место. Името на влезната датотека се задава од командна линија, а доколку не е зададено да се отпечати начин на употреба на програмата. Во датотеката нема зборови подолги од 15 букви.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX 16

int proverka (char *c)
{
    int i, vk=0, n=strlen(c), prva=tolower(*c), posledna=tolower(*(c+n-1));
    if (prva=='a' || prva=='e' || prva=='i' || prva=='o' || prva=='u') vk++;
    if (posledna=='a' || posledna=='e' || posledna=='i' || posledna=='o' || posledna=='u') vk++;
    if (vk==2) return 1;
    else return 0;
}

int main (int argc, char* argv[])
{
    char c, zbor[MAX];
    int vo_zbor=0, brojac=0, vkupno=0;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Invalid parameter\n");
        printf("Type name of file: ");
        dat=stdin;
    }

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalpha(c))
        {
            if (!vo_zbor) vo_zbor=1;
            zbor[brojac]=c;
            brojac++;
        }
        else
        if (vo_zbor)
        {
            vo_zbor=0;
            zbor[brojac]='\0';
        }
    }
}
```

```

        if (strlen(zbor)>4 && proverka(zbor))
        {
            puts(zbor);
            vkupno++;
        }
        zbor[0]='\0';
        brojac=0;
    }
}

if (vo_zbor)
{
    zbor[brojac]='\0';
    if (strlen(zbor)>4 && proverka(zbor))
    {
        puts(zbor);
        vkupno++;
    }
}
fclose(dat);
printf("\nVkupno %d zborovi\n",vkupno);
return 0;
}

```

11. Да се напише програма која за дадена текстуална датотека ќе ги изброи и отпечати зборовите подолги од 6 букви кои почнуваат и завршуваат на согласка. Да на се прави разлика меѓу големи и мали букви. Зборовите се составени само од букви, меѓусебно се одвоени со најмалку еден специјален знак, цифра или празно место. Името на влезната датотека се задава од командна линија, а доколку не е зададено да се отпечати начин на употреба на програмата. Во датотеката нема зборови подолги од 10 букви.

```

#include <stdio.h>
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAX 11

int uslov (char *c)
{
    int vk=0, prva=tolower(*c), posledna=tolower(*(c+strlen(c)-1));
    if (prva=='a' || prva=='e' || prva=='i' || prva=='o' || prva=='u') vk++;
    if (posledna=='a' || posledna=='e' || posledna=='i' || posledna=='o' || posledna=='u') vk++;
    if (vk==0) return 1;
    else return 0;
}

int main (int argc, char *argv[])
{
    char c, zbor[MAX];
    int vkupno=0, vo_zbor=0, i=0;
    FILE *dat;

    if (argc!=2)

```

```

{
    printf("ERROR: Invalid prameter\n");
    printf("Type name of file: ");
    dat=stdin;
}

if (argc==2) printf("Using: %s name_of_file\n",argv[1]);

if ((dat=fopen(argv[1],"r"))==NULL)
{
    printf("ERROR: File cannot be opened\n");
    return (-1);
}

while ((c=getc(dat))!=EOF)
{
    if (isalpha(c))
    {
        if (!vo_zbor) vo_zbor=1;
        zbor[i]=c;
        i++;
    }
    else
    if (vo_zbor)
    {
        zbor[i]='\0';
        if (strlen(zbor)>5 && uslov(zbor))
        {
            puts(zbor);
            vkupno++;
        }
        zbor[0]='\0';
        i=0;
    }
}
if (vo_zbor)
{
    zbor[i]='\0';
    if (strlen(zbor)>5 && uslov(zbor))
    {
        puts(zbor);
        vkupno++;
    }
}
printf("\nVkupno %d zborovi.\n",vkupno);
return 0;
}

```

12. Да се напише програма која за дадена текстуална датотека на екран ќе ги испечати редните броеви на редовите и содржината на редовите во датотеката кои имаат повеќе од 20 цифри. Името на датотеката се задава како параметар во командна линија. Во редовите нема повеќе од 80 знаци.

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <String.h>
#define MAX 80

int main (int argc, char* argv[])
{
    char c, red[MAX];
    int i=0, broevi=0, redovi=1, vkupno=0;
    FILE *dat;

    if (argc!=2) printf("ERROR: Invalid parameter\n");

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isdigit(c))
        {
            broevi++;
            vkupno++;
        }
        red[i]=c;
        i++;
        if (c=='\n')
        {
            red[i]='\0';
            if (broevi>20) printf("red %d\t%s\tbroevi %d\n",redovi, red, broevi);
            redovi++;
            red[i]='\0';
            i=0;
            broevi=0;
        }
    }
    if (broevi>20)
    {
        red[i]='\0';
        printf("red %d\t%s\tbroevi %d\n",redovi, red, broevi);
    }
    printf("Vkupno %d broevi.\n",vkupno);
    return 0;
}
```


13. Да се напише програма која за дадена текстуална датотека на екран ќе ги испечати редните броеви и содржината на редовите од датотеката кои имаат повеќе од 5 големи букви. Името на датотеката се задава како параметар од командна линија.

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <String.h>
#define MAX 100

int main (int argc, char* argv[])
{
    char c, red[MAX];
    int i=0, golemi=0, redovi=1, vkupno=0;
    FILE *dat;

    if (argc!=2) printf("ERROR: Invalid parameter\n");

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalpha(c) && isupper(c))
        {
            golemi++;
            vkupno++;
        }
        red[i]=c;
        i++;
        if (c=='\n')
        {
            red[i]='\0';
            if (golemi>5) printf("red %d\t%s\t golemi bukvi %d\n",redovi, red, golemi);
            redovi++;
            red[0]='\0';
            i=0;
            golemi=0;
        }
    }
    if (golemi>5) printf("red %d\t%s\t golemi bukvi %d\n",redovi, red, golemi);
    printf("\n\nVkupno %d golemi bukvi.\n",vkupno);
    return 0;
}
```

14. Една текстуална датотека (името се чита од командна линија) содржи зборови кои се составени од алфа – нумерички знаци. Да се избројат и прикажат зборовите кои се состојат само од букви.

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#define MAX 100

int proverka (char *c)
{
    int i, n=strlen(c);
    for (i=0; i<n; i++)
    {
        if (isdigit(*(c+i))) return 0;
    }
    return 1;
}

int main (int argc, char *argv[])
{
    char c, zbor[MAX];
    int i=0, vo_zbor=0, vkupno=0;
    FILE *dat;

    if (argc!=2) printf("ERROR: Invalid parameter\n");

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalnum(c))
        {
            if (!vo_zbor) vo_zbor=1;
            zbor[i]=c;
            i++;
        }
        else
        if (vo_zbor)
        {
            vo_zbor=0;
            zbor[i]='\0';
            if (proverka(zbor))
            {
                puts(zbor);
                vkupno++;
            }
            zbor[0]='\0';
            i=0;
        }
    }
}
```

```

    if (vo_zbor)
    {
        zbor[i]='\0';
        if (proverka(zbor))
        {
            puts(zbor);
            vkupno++;
        }
    }
    printf("\nVkupno %d zborovi.\n",vkupno);
    return 0;
}

```

15. Да се напише програма која ќе го испечати зборот со најголем број самогласки во текстуална датотека чиешто име се задава како аргумент од командна линија. Еден збор може да содржи најмногу 20 знаци.

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#define MAX 21

int br_samoglaski(char *c)
{
    int i, n=strlen(c), br=0;
    for (i=0; i<n; i++)
    {
        *(c+i)=tolower(*(c+i));
        if (*(c+i)=='a' || *(c+i)=='e' || *(c+i)=='i' || *(c+i)=='o' || *(c+i)=='u') br++;
    }
    return br;
}

int main (int argc, char *argv[])
{
    char c, zbor[MAX], maxzbor[MAX];
    int max_br=0, vo_zbor, i=0, broj;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Invalid parameters\n");
        return (-1);
    }

    if ((dat=fopen(argv[1], "r"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {

```

```

    if (isalpha(c))
    {
        if (!vo_zbor) vo_zbor=1;
        zbor[i]=c;
        i++;
    }
    else
    if (vo_zbor)
    {
        vo_zbor=0;
        zbor[i]='\0';
        broj=br_samoglaski(zbor);
        if (broj>max_br)
        {
            max_br=broj;
            strcpy(maxzbor, zbor);
        }
        broj=0;
        i=0;
    }
}

if (vo_zbor)
{
    zbor[i]='\0';
    broj=br_samoglaski(zbor);
    if (broj>max_br)
    {
        max_br=broj;
        strcpy(maxzbor, zbor);
        zbor[i]='\0';
    }
}

printf("Zborot %s ima %d samoglaski.\n",maxzbor, max_br);
return 0;
}

```

16. Да се напише програма која за дадена текстуална датотека ќе ги изброи редовите во кои има повеќе од 10 самогласки. Да не се прави разлика меѓу големи и мали букви. Името на датотеката се задава од командна линија.

НАЧИН:

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#define MAX 100

int proverka (char *c)
{
    int i, br=0, n=strlen(c);
    for (i=0; i<n; i++)
    {
        *(c+i)=tolower(*(c+i));
        if (*(c+i)=='a' || *(c+i)=='e' || *(c+i)=='i' || *(c+i)=='o' || *(c+i)=='u') br++;
    }
    return br;
}

int main (int argc, char *argv[])
{
    char c, red[MAX];
    int redovi=1, i=0;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Invalid parameters\n");
        return (-1);
    }

    if ((dat=fopen(argv[1], "r"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (c!='\n')
        {
            red[i]=c;
            i++;
        }
        else
        {
            red[i]='\0';
            if (proverka(red)>10) printf("red %d\t%s\n",redovi,red);
            red[i]='\0';
            i=0;
        }
        redovi++;
    }
}
```

```

        redovi++;
    }
}
red[i]='\0';
if (proverka(red)>10) printf("red %d\t%s\n",redovi,red);
return 0;
}

```

II НАЧИН:

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

int proverka (char c)
{
    int br=0;
    if (c=='a' || c=='e' || c=='i' || c=='o' || c=='u') br++;
    return br;
}

int main (int argc, char *argv[])
{
    char red[MAX];
    int redovi=0, i=0, samoglaski=0;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Invalid parameters\n");
        return (-1);
    }

    if ((dat=fopen(argv[1], "r"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    while (fgets(red, MAX, dat)!=NULL)
    {
        redovi++;
        samoglaski=0;
        for (i=0; i<strlen(red); i++)
        {
            if (proverka(red[i])) samoglaski++;
        }
        if (samoglaski>10) printf("red %d: %s\n",redovi, red);
    }
    return 0;
}

```

17. Да се напише програма која за дадена текстуална датотека ќе ги изброи и отпечати зборовите подолги од 5 букви кои имаат непарен број букви, започнуваат на самогласка, а завршуваат на согласка. Да не се прави разлика меѓу големи и мали букви. Зборовите се составени само од букви, а меѓусебно се одвоени со најмалку еден специфичен знак, цифра или празно место. Името на влезната датотека се задава од командна линија, а доколку не е зададено да се испечати начин на употреба на програмата. Во датотеката нема зборови подолги од 15 букви.

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define MAX 16

int samoglaska (char c)
{
    if (c=='a' || c=='e' || c=='i' || c=='o' || c=='u') return 1;
    return 0;
}

int main (int argc, char *argv[])
{
    char c, zbor[MAX];
    int vkupno=0, i=0, vo_zbor=0;
    FILE*dat;

    if (argc!=2)
    {
        printf("Using: %s name_of_file\n",argv[0]);
        return (-1);
    }

    if ((dat=fopen(argv[1], "r"))==NULL)
    {
        printf("ERROR: Cannot open the file\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalpha(c))
        {
            if (!vo_zbor) vo_zbor=1;
            zbor[i]=c;
            i++;
        }
        else
        if (vo_zbor)
        {
            vo_zbor=0;
            zbor[i]='\0';
            if ((i>5) && (i%2) && samoglaska(*zbor) && !samoglaska(*(zbor+i-1)))
            {
                puts(zbor);
                vkupno++;
            }
        }
    }
}
```

```

        zbor[0]='\0';
        i=0;
    }
}
if (vo_zbor)
{
    zbor[i]='\0';
    if ((i>5) && (i%2) && samoglaska(*zbor) && !samoglaska(*(zbor+i-1)))
    {
        puts(zbor);
        vkupno++;
    }
}
printf("\nVkupno %d zborovi go ispolnuvaat uslovot.\n",vkupno);
fclose(dat);
return 0;
}

```

18. Да се напише програма што дадена текстуална датотека “pocetna.txt” ќе ја промени во друга текстуална датотека “krajna.txt”, заменувајќи ги сите ознаки за нов ред со точка.

```

#include <stdio.h>

int main ()
{
    char c;
    FILE *dat1;
    FILE *dat2;

    if ((dat1=fopen("pocetna.txt","r"))==NULL)
    {
        printf("ERROR: file cannot be found.\n");
        return (-1);
    }

    dat2=fopen("krajna.txt","w");

    while ((c=getc(dat1))!=EOF)
    {
        if (c=='\n') putc('.',dat2);
        else putc(c,dat2);
    }
    fclose(dat1);
    fclose(dat2);
    return 0;
}

```


19. Да се напише програма која ќе ја споредува содржината на две датотеки.

```
#include <stdio.h>

int main (int argc, char *argv[])
{
    char c1, c2;
    int l=0, isti=1;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("Using: name_of_file <file1> <file2>\n");
        return (-1);
    }

    if ((dat1=fopen(argv[1], "rb"))==NULL)
    {
        printf("The first file cannot be opened\n");
        return (-1);
    }

    if ((dat2=fopen(argv[2], "rb"))==NULL)
    {
        printf("The second file cannot b opened\n");
        return (-1);
    }

    while (!feof(dat1))
    {
        c1=getc(dat1);
        c2=getc(dat2);
        if (c1!=c2)
        {
            printf("Datotekite se razlikuvat vo bajtot %lu",l);
            isti=0;
            break;
        }
        l++;
    }
    if (isti) printf("Datotekite se ednakvi.\n");
    close(dat1);
    close(dat2);
    return 0;
}
```

20. За некоја текстуална датотека чиешто име се задава како аргумент од командна линија, да се напише програма што ќе ја испечати линијата која го содржи најдолгиот збор во датотеката. Максималниот број на знаци во една линија е 80. На крај да се отпечати редниот број на бараната линија.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX 81

int main (int argc, char *argv[])
{
    char red[81], maxred[81];
    int brrred=0, brojred=0, vo_zbor=0, bukvi=0, maxbukvi=0, i=0;
    FILE *dat;

    dat=fopen(argv[1],"r");

    maxred[0]='\0';
    while (fgets(red, MAX, dat)!=NULL)
    {
        brrred++;
        for (i=0; i<strlen(red); i++)
        {
            if (isalpha(red[i]))
            {
                if (!vo_zbor) vo_zbor=1;
                bukvi++;
            }
            else
            if (vo_zbor)
            {
                vo_zbor=0;
                if (bukvi>maxbukvi)
                {
                    maxbukvi=bukvi;
                    strcpy(maxred,red);
                    brojred=brrred;
                }
                bukvi=0;
            }
        }
    }
    printf("red %d : %s\n",brojred, maxred);
    return 0;
}
```

21. Да се напише програма што за дадена текстуална датотека со име "pocetna.txt" ќе изброи и ќе прикаже на екран колку пати цифрата 5 се појавила во датотеката.

```
#include <stdio.h>

int main ()
{
    char c;
    int brojac=0;
    FILE *dat;

    if ((dat=fopen("pocetna.txt","r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (c=='5') brojac++;
    }
    printf("Brojot 5 se pojavil %d pati.\n",brojac);
    return 0;
}
```

22. Да се напише програма што од даден датотека ќе го изброи вкупниот број на броеви во секој ред и ќе го испечати во нова датотека. Името на влезната и излезната датоека се задава преку командната линија, а доколку не е зададено да се испечати начин на употреба.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAX 100

int main (int argc, char *argv[])
{
    char c, red[MAX];
    int i=0, broevi=0, redovi=1;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("Using: %s name_of_file <file1> <file2>\n", argv[0]);
        return (-1);
    }

    if ((dat1=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file canot be opened\n");
        return (-1);
    }

    if ((dat2=fopen(argv[2],"w"))==NULL)
```

```

{
    printf("ERROR: file canot be opened\n");
    return (-1);
}

while ((c=getc(dat1))!=EOF)
{
    if (isdigit(c))
    {
        broevi++;
        red[i]=c;
        i++;
    }
    else
    {
        red[i]=c;
        i++;
    }
    if (c=="\n")
    {
        red[i]='\0';
        fprintf(dat2,"red %d\t%s . . . %d broevi\n",redovi, red, broevi);
        redovi++;
        i=0;
        red[0]='\0';
        broevi=0;
    }
}
if (isdigit(c))
{
    broevi++;
    red[i]=c;
    i++;
    red[i]='\0';
    fprintf(dat2,"red %d\t%s . . . %d broevi\n",redovi, red, broevi);
}
return 0;
}

```

23. Да се напише програма која за дадена текстуална датотека ќе ги определи редните броеви на редовите во датотеката кои имаат повеќе од 5 големи букви. Податоците да се запишат во излезна датотека. Името на датотеките се задава како параметар од командна линија.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>

int main (int argc, char *argv[])
{
    char c;
    int red=1, golemi=0;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("ERROR: Invalid parameters\n");
        return (-1);
    }

    if ((dat1=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    if ((dat2=fopen(argv[2],"w"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat1))!=EOF)
    {
        if (isupper(c)) golemi++;
        if (c=="\n")
        {
            fprintf(dat2,"%d\t%d\n",red,golemi);
            golemi=0;
            red++;
        }
    }
    if (isupper(c))
    {
        golemi++;
        fprintf(dat2,"%d\t%d\n",red,golemi);
    }
    return 0;
}
```

24. Да се напише програма која за дадена текстуална датотека ќе ги изброи редовите кои содржат најмалку три збора. Овој број да се отпечати во друга датотека. Името на влезната и излезната датотека се задаваат од командна линија, а доколку не се зададени да се испечати начин на употреба на програмата. Зборовите содржат букви и цифри а се одделени со бланко или специјални знаци.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>

int main (int argc, char *argv[])
{
    char c;
    int vo_zbor=0, red=1, zborovi=0;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("Using: %s name_of_file <fil1> <file2>\n",argv[0]);
        return (-1);
    }

    if ((dat1=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    if ((dat2=fopen(argv[2],"w"))==NULL)
    {
        printf("ERROR: File cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat1))!=EOF)
    {
        if (isalnum(c))
        {
            if (!vo_zbor) vo_zbor=1;
        }
        else
        if (vo_zbor)
        {
            vo_zbor=0;
            zborovi++;
        }
        if (c=='\n')
        {
            if (zborovi>2) fprintf(dat2,"%d red\n",red);
            red++;
            zborovi=0;
            vo_zbor=0;
        }
    }
}
```

```

    }
    if (vo_zbor)
    {
        zborovi++;
        if (zborovi>2) fprintf(dat2,"%d red\n",red);
    }
    return 0;
}

```

25. Да се напише програма која за дадена текстуална датотека ќе го испечати зборот што содржи најмногу различни самогласки. Зборовите се составени само од букви, а меѓусебно се одвоени со најмалку еден специјален знак, цифра или бланко код. Името на влезната датотека се задава од командна линија, а ако не е зададено да се испечати начин на употреба на програмата. Во датотеката нема зборови подолги од 15 букви.

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#define MAX 100

int razlicni (char *c)
{
    int j, n=strlen(c), a=0, e=0, i=0, o=0, u=0;
    for (j=0; j<n; j++)
    {
        c[j]=tolower(c[j]);
        if (c[j]=='a') a=1;
        if (c[j]=='e') e=1;
        if (c[j]=='i') i=1;
        if (c[j]=='o') o=1;
        if (c[j]=='u') u=1;
    }
    return (a+e+i+o+u);
}

int main (int argc, char *argv[])
{
    char c, zbor[MAX], maxzbor[MAX];
    int i=0, vo_zbor=0, maxrsam=0, maxsam=0;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Invalid parameter\n");
        return (-1);
    }

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }
}

```

```

while ((c=getc(dat))!=EOF)
{
    if (isalpha(c))
    {
        if (!vo_zbor) vo_zbor=1;
        zbor[i]=c;
        i++;
    }
    else
    if (vo_zbor)
    {
        vo_zbor=0;
        zbor[i]='\0';
        if (razlicni(zbor) > maxrsam)
        {
            maxrsam=razlicni(zbor);
            strcpy(maxzbor, zbor);
        }
        zbor[0]='\0';
        i=0;
    }
}
if (vo_zbor)
{
    zbor[i]='\0';
    if (razlicni(zbor) > maxrsam)
    {
        maxrsam=razlicni(zbor);
        strcpy(maxzbor, zbor);
    }
}
printf("%s\n",maxzbor);
return 0;
}

```


26. Да се напише програма кој за дадена датотека ќе ги изброи и отпечати зборовите палиндроми. Да не се прави разлика меѓу големи и мали букви. Зборовите се составени само од букви, а меѓусебно се одвоени со најмалку еден специјален знак, цифра или празно место. Името на влезната датотека се задава од командна линија, а доколку не е зададено да се испечати порака за грешка. Во датотеката нема зборови подолги од 10 букви.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 10

int palindrom (char *c)
{
    int i, n=strlen(c);
    for (i=0; i<n; i++)
    {
        if (*(c+i)!=*(c-i+n-1)) return 0;
    }
    return 1;
}

int main (int argc, char *argv[])
{
    char c, zbor[MAX];
    int i=0, vo_zbor=0, palindromi=0;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Invalid prameter\n");
        return (-1);
    }

    if ((dat=fopen(argv[1], "r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalpha(c))
        {
            if (!vo_zbor) vo_zbor=1;
            zbor[i]=c;
            i++;
        }
        else
        if (vo_zbor)
        {
            zbor[i]='\0';
            if (palindrom(zbor))
            {

```

```

        printf("%s\n",zbor);
        palindromi++;
    }
    i=0;
    zbor[0]='\0';
}
}
if (vo_zbor)
{
    zbor[i]='\0';
    if (palindrom(zbor))
    {
        printf("%s\n",zbor);
        palindromi++;
    }
}
return 0;
}

```

27. Да се напише програма која во нова текстуална датотека ќе ги ископира само оние редови од дадена датотека кои го содржат симболот '@'. Името на влезната и излезната датотека се задаваат како параметар од командна линија. Во атотеката нема редови подолги од 80 знаци. Да се испечатат на екран редните броеви на редовите од влезната датотека кои се пренесуваат во излезната.

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 100

int et (char *c)
{
    int i, n=strlen(c);
    for (i=0; i<n; i++)
    {
        if (*(c+i)=='@') return 1;
    }
    return 0;
}

int main (int argc, char *argv[])
{
    char c, red[MAX];
    int i=0, redovi=1;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("ERROR: Invalid prameter\n");
        return (-1);
    }
}

```

```

if ((dat1=fopen(argv[1],"r"))==NULL)
{
    printf("ERROR: file cannot be opened\n");
    return (-1);
}

if ((dat2=fopen(argv[2],"w"))==NULL)
{
    printf("ERROR: file cannot be opened\n");
    return (-1);
}

while ((c=getc(dat1))!=EOF)
{
    if (c!='\n')
    {
        red[i]=c;
        i++;
    }
    else
    {
        red[i]='\0';
        if (et(red))
        {
            fprintf(dat2,"%s\n",red);
            printf("red %d\n",redovi);
        }
        redovi++;
        i=0;
        red[0]='\0';
    }
}
if (isalnum(c))
{
    red[i]=c;
    i++;
    red[i]='\0';
    if (et(red))
    {
        fprintf(dat2,"%s\n",red);
        printf("red %d\n",redovi);
    }
}
return 0;
}

```

28. Да се напише програма која ќе прочита датоека со име vlez.txt. Програмата треба да го испечати бројот на реченици во датотеката. Речениците завршуваат на некој од следниве знаци: ".", "!" или "?". Исто така програмата треба да испечати најдолгиот ред и бројот на знаци во најдолгата реченица.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 100

int main ()
{
    char c, red[MAX], redmax[MAX], brmax=0, maxrec=0;
    int kluc=0, i=0, j=0, rec=0;
    FILE *dat;

    if ((dat=fopen("vlez.txt","r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if ((c=='.' || c=='!' || c=='?') && (kluc==0))
        {
            red[i]=c;
            i++;
            j++;
            kluc=1;
            rec++;
            if (j>maxrec) maxrec=j;
            j=0;
        }
        else
        {
            red[i]=c;
            i++;
            j++;
        }
        if (c!='.' && c!='!' && c!='?') kluc=0;
        if (c=='\n')
        {
            red[i]='\0';
            if (i>brmax)
            {
                strcpy(redmax,red);
                brmax=i;
            }
            red[0]='\0';
            i=0;
        }
    }
}
```

```

    }
    printf("%d recenici.\n",rec);
    printf("%d znaci vo najdolgata recenica\n",maxrec);
    printf("%s\n",redmax);
    return 0;
}

```

29. Да се напише програма која за дадена текстуална датотека ќе го најде просечниот број на појавувања на самогласките во речниците и ќе го отпечати во друга датотека. Притоа сите реченици меѓусебно се одвоени со точка. Името на влезната и излезната датотека се задаваат од командна линија, а доколку не се зададени да се испечати начин на употреба на програмата.

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 100

int samoglaska (char c)
{
    c=tolower(c);
    if (c=='a' || c=='e' || c=='i' || c=='o' || c=='u') return 1;
    return 0;
}

int main (int argc, char *argv[])
{
    char c;
    int rec=0, sam=0, vkupno=0, kluc=0;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("ERROR: %s name_of_file <file1><file2>\n");
        return (-1);
    }

    if ((dat1=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    if ((dat2=fopen(argv[2],"w"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat1))!=EOF)
    {
        if (c!='.')
        {
            kluc=0;

```

```

        if (samoglaska(c)) sam++;
    }
    if (c=='.' && kluc==0)
    {
        kluc=1;
        rec++;
        vkupno+=sam;
        sam=0;
    }
}
if (isalnum(c))
{
    if (samoglaska(c)) sam++;
    vkupno+=sam;
    rec++;
}
printf("Odnosot e %d/%d = %3.5f\n",vkupno, rec, (float)vkupno/rec);
return 0;
}

```

30. Да се напише програма која дадена датотека ќе ја промени на тој начин што секоја мала буква ќе ја промени во голема и обратно, додека останатите знаци од датоеката ќе останат непроменети. Изменетата содржина да се отпечти во друга датотека. Името на влезната и излезната датотека се задавата од командна линија, а доколку не се зададени да се испечати начин на употреба на програмата.

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 100

int main (int argc, char *argv[])
{
    char c;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("ERROR: %s name_of_file <file1><file2>\n");
        return (-1);
    }

    if ((dat1=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    if ((dat2=fopen(argv[2],"w"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }
}

```

```

    }

    while ((c=getc(dat1))!=EOF)
    {
        if (islower(c)) c=toupper(c);
        else
            if (isupper(c)) c=tolower(c);
        putc(c,dat2);
    }
    return 0;
}

```

31. Во текстуална датотека vlez.txt запишана е листа на слики со соодветни растојанија во формат:

```

imeSlika1.jpg celDel1.decimalenDel1
ImeSlika2.jpg celDel2.decimalenDel2
...
ImeSlikaN.jpg celDelN.decimalenDelN

```

Имињата на сликите се составени од букви и цифри, а наставката е одделена со точка. Растојанието и името на сликата се одделени со празно место. Растојанието е децимален број со произволен број на цифри во целиот и децималниот дел. Да се напише програма која ќе ја измени влезната датотека така што точките од децималните броеви ќе ги замени со запирки и променетата верзија ќе ја запише во излезна датотека. Името на излезната датотека се задава како аргумент на командна линија, а ако не е зададено да се испечати начин на употреба на програмата. Во последниот ред од излезната датотека треба да се запише бројот на парови слика – растојание. На крај, на екран да се испечати бројот на слики за кои соодветното растојание е помало од 1.0.

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#define MAX 81

int main (int argc, char *argv[])
{
    char celbroj[10], red[MAX];
    int i=0, j=0, parovi=0, pomali=0;
    FILE *dat1, *dat2;

    if (argc!=2)
    {
        printf("Using: %s izlez.txt\n", argv[0]);
        return (-1);
    }

    if ((dat1=fopen("vlez.txt", "r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    if ((dat2=fopen(argv[1], "w"))==NULL)
    {

```

```

        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while (fgets(red, MAX, dat1)!=NULL)
    {
        parovi++;
        for (i=0; i<strlen(red); i++)
        {
            if (red[i]==' ')
            {
                j=0;
                i++;
                while(red[i]!='.')
                {
                    celbroj[j]=red[i];
                    j++;
                    i++;
                }
                red[i]=';';
            }
        }
        if ((j==1) && (celbroj[j-1]=='0')) pomali++;
        fputs(red,dat2);
    }
    fprintf(dat2, "\n\nBr. parovi: %d\n",parovi);
    printf("%d слики имаат растојание < 1.0\n",pomali);
    return 0;
}

```

32. Да се напише програма која ќе прочита датотека со име vlez.txt. Програмата треба да го испечати: бројот на редови, бројот на знаци (без празни места) и бројот на празни места.

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 100

int main ()
{
    char c;
    int redovi=0, brznaci=0, brblanko=0;
    FILE *dat;

    if ((dat=fopen("vlez.txt","r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (c=='\n') redovi++;
    }
}

```



```

        if (c==' ') brblanko++;
        else brznaci++;
    }
    if (c==' ') brblanko++;
    else brznaci++;
    redovi++;
    printf("%d\t%d\t%d\n",redovi, brblanko, brznaci-redovi);
    return 0;
}

```

33. Да се напише програма која содржината од една датотека ќе ја ископира во друга датотека на тој начин што секој последователна појава на празно место ќе ја замени со еден знак за празно место.

НАЧИН:

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 100

int main (int argc, char *argv[])
{
    char c;
    int kluc=0;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("ERROR: %s name_of_file <file1><file2>\n");
        return (-1);
    }

    if ((dat1=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    if ((dat2=fopen(argv[2],"w"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat1))!=EOF)
    {
        if (c==' ' && kluc==0)
        {
            kluc=1;
            putc(c,dat2);
        }
        else if (c!=' ')
        {

```

```

        kluc=0;
        putc(c,dat2);
    }
}
return 0;
}

```

II НАЧИН:

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 100

int main (int argc, char *argv[])
{
    char c;
    int kluc=0;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("ERROR: %s name_of_file <file1><file2>\n");
        return (-1);
    }

    if ((dat1=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    if ((dat2=fopen(argv[2],"w"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat1))!=EOF)
    {
        if (c==' ')
        {
            putc(c,dat2);
            while ((c=getc(dat1))==' ');
            putc(c,dat2);
        }
        else putc(c,dat2);
    }
    return 0;
}

```

34. Да се напише програма која од една датотека ќе ги избрише сите дупликати кои се наоѓаат еден до друг и новата верзија ќе ја пренесе на излезна датотека.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX 81

int main (int argc, char *argv[])
{
    char red[81], a;
    int i;
    FILE *dat1, *dat2;

    dat1=fopen(argv[1],"r");
    dat2=fopen(argv[2],"w");

    while(fgets(red,MAX,dat1)!=NULL)
    {
        for (i=0; i<strlen(red); i++)
        {
            while ((red[i]==red[i+1]) && (i<strlen(red)))    i++;
            putc(red[i],dat2);
        }
    }
    return 0;
}
```

35. Да се напише програма која од дадена датотека ќе го најде најголемиот збор. Тој збор, редот, бројот на редот и бројот на знаци во зборот да се испечатат во излезна датотека чие име се задава во командната линија аедно со името на влезната датотека од која се чита.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX 81

int main (int argc, char *argv[])
{
    char red[MAX], zbor[MAX], maxzbor[MAX], maxred[MAX];
    int i=0, j=0, vo_zbor=0, brred=0, maxbrred=0, maxz=0;
    FILE *dat1, *dat2;

    maxred[0]='\0';
    maxzbor[0]='\0';
    dat1=fopen(argv[1],"r");
    dat2=fopen(argv[2],"w");

    while (fgets(red, MAX, dat1)!=NULL)
    {
        brred++;
    }
```

```

        for (i=0; i<strlen(red); i++)
        {
            if (isalpha(red[i]))
            {
                if (!vo_zbor) vo_zbor=1;
                zbor[j]=red[i];
                j++;
            }
            else
            if (vo_zbor)
            {
                vo_zbor=0;
                zbor[j]='\0';
                if (j>maxz)
                {
                    maxz=j;
                    strcpy(maxzbor, zbor);
                    strcpy(maxred, red);
                    maxbrred=brred;
                }
                j=0;
                zbor[0]='\0';
            }
        }
    }
    fprintf(dat2,"zbor: %s\tred#: %d\tred: %s\n",maxzbor, maxbrred, maxred);
    return 0;
}

```

36. Да се напише програма која ќе изброи колку пати еден збор се сретнува во дадена датотека. Зборот, како и името на датотеката од која се чита се задават како параметри во командната линија.

```

#include <stdio.h>
#include <string.h>
#define MAX 21

int main (int argc, char *argv[])
{
    char c, zbor[MAX];
    int i=0, vkupno=0, vo_zbor=0;
    FILE *dat;

    if (argc!=3)
    {
        printf("ERROR: Invalid parameters\n");
        return (-1);
    }

    if ((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }
}

```

```

while ((c=getc(dat))!=EOF)
{
    if (isalpha(c))
    {
        if (!vo_zbor) vo_zbor=1;
        zbor[i]=c;
        i++;
    }
    else if (vo_zbor)
    {
        vo_zbor=0;
        zbor[i]='\0';
        if (strcmp(zbor, argv[2])==0) vkupno++;
        i=0;
    }
}
if (vo_zbor)
{
    zbor[i]='\0';
    if (strcmp(zbor, argv[2])==0) vkupno++;
}
printf("%s se pojavuva %d pti vo datotekata %s\n",argv[2], vkupno, argv[1]);
return 0;
}

```

37. Да се напише програма која ќе изброи колку редови има една датотека, колку зборови има максималниот ред и кој е тој ред. Добиените податоци да се запишат во нова датотека чие име се задава од командна линија, а доколку не е зададено да се внесе преку стандардниот влез.

НАЧИН:

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 81

int main (int argc, char *argv[])
{
    char redovi[MAX];
    int i=0, vo_zbor=0, maxred=0, red=0, zbor=0, maxzbor=0;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("ERROR: Invalid parameter\n");
        return (-1);
    }

    if ((dat1=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }
}

```

```

    }

    if ((dat2=fopen(argv[2], "w"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while(fgets(redovi, MAX, dat1)!=NULL)
    {
        red++;
        zbor=0;
        for (i=0; i<strlen(redovi); i++)
        {
            if (isalpha(redovi[i]))
            {
                if (!vo_zbor) vo_zbor=1;
            }
            else if (vo_zbor)
            {
                vo_zbor=0;
                zbor++;
            }
        }
        if (zbor>maxzbor)
        {
            maxzbor=zbor;
            maxred=red;
        }
    }
    if (vo_zbor)
    {
        zbor++;
        if (zbor>maxzbor)
        {
            maxzbor=zbor;
            maxred=red;
        }
    }
    fprintf(dat2,"Od %d redovi, maksimalen e %d so %d zborovi\n",red, maxred, maxzbor);
    return 0;
}

```

II НАЧИН:

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 81

int main (int argc, char *argv[])
{
    char c;
    int vo_zbor=0, maxred=0, red=1, zbor=0, maxzbor=0;

```

```

FILE *dat1, *dat2;

if (argc!=3)
{
    printf("ERROR: Invalid parameter\n");
    return (-1);
}

if ((dat1=fopen(argv[1], "r"))==NULL)
{
    printf("ERROR: file cannot be opened\n");
    return (-1);
}
if ((dat2=fopen(argv[2], "w"))==NULL)
{
    printf("ERROR: file cannot be opened\n");
    return (-1);
}

while ((c=getc(dat1))!=EOF)
{
    if (isalpha(c))
    {
        if (!vo_zbor) vo_zbor=1;
    }
    if (vo_zbor)
    {
        vo_zbor=0;
        zbor++;
    }
    if (c=='\n')
    {
        if (zbor>maxzbor)
        {
            maxzbor=zbor;
            maxred=red;
        }
        zbor=0;
        red++;
    }
}
if (vo_zbor)
{
    red++;
    zbor++;
}
if (zbor>maxzbor)
{
    maxzbor=zbor;
    maxred=red;
}
fprintf(dat2, "Od %d redovi, maksimalen e %d so %d zborovi.\n", red, maxred, maxzbor);
return 0;
}

```

38. За даден датотека, чие име се задава од командна линија, да се испечати вкупниот број на зборови кои содржат повеќе самогласки од согласки. Зборовите се состојат од алфанумерички знаци и се одделени со најмалку едно празно место или знаци за интерпункција.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX 21

int samoglaska(char *c)
{
    int i, n=strlen(c), vk=0;
    for (i=0; i<n; i++)
    {
        if (c[i]=='a' || c[i]=='e' || c[i]=='i' || c[i]=='o' || c[i]=='u') vk++;
    }
    return vk;
}

int soglaska (char *c)
{
    int i, n=strlen(c), vk=0;
    for (i=0; i<n; i++)
    {
        if ((isdigit(c[i])==0) && c[i]!='a' && c[i]!='e' && c[i]!='i' && c[i]!='o' && c[i]!='u') vk++;
    }
    return vk;
}

int main (int argc, char *argv[])
{
    char c, zbor[MAX];
    int i, vo_zbor=0, vkupno=0;
    FILE *dat;

    if (argc!=2)
    {
        printf("ERROR: Invalid parameter\n");
        return (-1);
    }

    if ((dat=fopen(argv[1], "r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }

    while ((c=getc(dat))!=EOF)
    {
        if (isalnum(c))
        {
            if (!vo_zbor) vo_zbor=1;
        }
    }
}
```



```

        zbor[i]=c;
        i++;
    }
    else if (vo_zbor)
    {
        vo_zbor=0;
        zbor[i]='\0';
        if (samoglaska(zbor) > soglaska(zbor))
        {
            printf("%s\n",zbor);
            vkupno++;
        }
        i=0;
        zbor[0]='\0';
    }
}
if (vo_zbor)
{
    zbor[i]='\0';
    if (samoglaska(zbor) > soglaska(zbor))
    {
        printf("%s\n",zbor);
        vkupno++;
    }
}
printf("\nVkupno: %d\n",vkupno);
return 0;
}

```

39. Да се напише програма која од дадена датотека ќе ги отстрани редовите кои се празни. Новата содржина ќе ја испечати во излезна датотека чие име се задава од командна линија.

```

#include <stdio.h>
#include <string.h>
#define MAX 21

int main (int argc, char *argv[])
{
    char red[MAX];
    int i, vk=0;
    FILE *dat1, *dat2;

    if (argc!=3)
    {
        printf("ERROR: Invalid parameter\n");
        return (-1);
    }

    if ((dat1=fopen(argv[1],"r"))==NULL)
    {
        printf("ERROR: file cannot be opened\n");
        return (-1);
    }
}

```

```
if ((dat2=fopen(argv[2], "w"))==NULL)
{
    printf("ERROR: file cannot be opened\n");
    return (-1);
}

while (fgets(red, MAX, dat1)!=NULL)
{
    for (i=0; i<strlen(red); i++)
    {
        if ((strlen(red)==1) && red[0]=='\n') break;
        else putc(red[i], dat2);
    }
}
return 0;
}
```

VI

СТРУКТУРИ во С

- Да се напише структура за претставување на стан. За секој стан се чуваат информации за број на стнот, број на соби во станот, кат на кој се наоѓа станот, дали е опремен и површина што ја зафаќа во m^2 . Покрај тоа, треба да се обезбеди структура за претставување на станбена зграда во која се чуваат информации за името на зградата (низа од 20 знаци), адреса (низа од 30 знаци), низа од станови во зградата (максимум 50) и број на станови во зградата. Треба да се обезбди функција за определување на цена на стан, ако цената се пресметува како збир според следните критериуми:
 - секој m^2 има цена од 1000€
 - за катот на кој се наоѓа станот се плаќа дополнително: I кат: 20€, II кат 15€, III стан 10€ IV стан 5€, а за останатите катови не се плаќа дополнително
 - ако станот е опремен се плаќа дополнително 2000€

Да се напише функција за печатење на информациите за зграда во формат:

Зграда: **имеНаЗграда**

Адреса: **адреса**

1. станБр	брСоби	површина	опремен/не е опремен	цена
2. станБр	брСоби	површина	опремен/не е опремен	цена
3. ...				

```
#include <stdio.h>
#define MAX 50

struct stan {
    int stanNo;
    int roomNo;
    int floor;
    char Eq;
    int sqf;
};

struct zgrada {
    char name [20];
```

```

    char adresa [30];
    int apart_No;
    struct stan apart [MAX];
};

int apart_Price (struct stan c)
{
    int res=(c.sqf)*1000;
    switch (c.floor)
    {
        case 1: res+=20; break;
        case 2: res+=15; break;
        case 3: res+=10; break;
        case 4: res+=5; break;
        default: res=res;
    }
    if (c.Eq=='+') res+=2000;
    return res;
}

void p(struct zgrada z)
{
    printf("Zgrada %s\n", z.name);
    printf("Adresa %s\n", z.adresa);
    int i;
    for (i=0; i<z.apart_No; i++)
    {
        printf("%d\t%d\t%d\t%d\t\t", i+1, z.apart[i].stanNo, z.apart[i].roomNo,
            z.apart[i].sqf);
        if (z.apart[i].Eq=='+') printf("opremen\t%d\n", apart_Price(z.apart[i]));
        else printf("ne e opremen\t%d\n", apart_Price(z.apart[i]));
    }
}

int main ()
{
    struct zgrada flat;
    int i;
    printf ("Ime na zgrada: ");   scanf ("%s",&flat.name);
    printf ("Adresa na zgrada: "); scanf ("%s",&flat.adresa);
    printf ("Broj na stanovi: "); scanf ("%d",&flat.apart_No);
    printf ("\n\n");
    for (i=0; i<flat.apart_No; i++)
    {
        printf ("Br. na stan: "); scanf ("%d", &flat.apart[i].stanNo);
        printf ("Br. na sobi: "); scanf ("%d", &flat.apart[i].roomNo);
        printf ("Kat: ");          scanf ("%d", &flat.apart[i].floor);
        printf ("Povrsina: ");      scanf ("%d", &flat.apart[i].sqf);
        printf ("Oprema: ");        scanf ("%c", &flat.apart[i].Eq);
    }
    printf ("\n\n");
    p (flat);
    return 0;
}

```

2. Да се состави програма која ќе работи со структури за студенти, факултети и универзитети. Структурата за студент треба да содржи информации за име, презиме, индекс и покажувач кон структура за факултет. Структурата за факултет треба да содржи име на факултет, адреса и покажувач кон структура за универзитет. Структурата за универзитет треба да содржи име на универзитет и адреса. Да се напише функција која ќе прима листа од студенти и ќе го печати името и презимето на секој студент како и факултетот и универзитетот каде студира.