

Vnos in prikaz podatkov

1. naloga

```
using namespace std; //vključimo imenski prostor std
#include <iostream> //napoved uporabe knjižnice za vhodni in izhodni tok
//Zgornji dve vrstici sta vedno vključeni v program!
//Rešitev z uporabo ukaza za skok v novo vrsto:
int main()
{
    int x; //spremenljivka, ki jo potrebujemo v ukazu za zaustavitev
    cout << "b" << endl << "e" << endl << "s" << endl << "e" << endl;
    cout << "d" << endl << "i" << endl << "l" << endl << "o";
    cin >> x; //zaustavitev konzolnega okna, da preberemo vsebino
    return 0;
}
```

//Rešitev z uporabo posebne znakovne konstante:

```
int main()
{
    int x;
    cout << "b\ns\ne\nl\ni\nl\no";
    cin >> x;
    return 0;
}
```

2. naloga

```
int main()
{
    double PI = 22 / 7.0; //vsaj en od koeficientov mora biti decimalno
//številko, da bodo v rezultatu decimalke
    cout << "Število PI: " << PI;
    cin >> PI;
    return 0;
}
```

3. naloga

```
int main()
{
    char znak;
    cout << "Vnesi znak: "; //navodilo uporabniku
    cin >> znak; //vnos znaka
    int koda = (int)znak; //pretvorba znaka v celo število
    cout << "ASCII " << koda;
    cin >> koda;
    return 0;
}
```

4. naloga

```
int main()
{
    double a,b;
    cout << "Vnesi stevilo a: ";
    cin >> a;
    cout << "Vnesi stevilo b: ";
    cin >> b;
    double x = (-3 * b + 5 * a * a + b * b) / (2 * a + 4 * b);
    cout << "x = " << x;
    cin >> x;
    return 0;
}
```

Naključne vrednosti in uporaba metod matematične knjižnice

1. naloga

```
int main()
{
    srand(time(NULL)); //inicializacija generatorja naključnih števil
    int prvo = rand() % 90 + 10; //izberemo vrednost med 0 in 90 in prištejemo 10
    int drugo = rand() % 61 - 30;
    double tretje = ((double)rand() / RAND_MAX) * 10;
    cout << "Prvo: " << prvo << endl << "Drugo: " << drugo << endl << "Tretje: "
        << tretje;
    cin >> prvo;
    return 0;
}
//Za metodo time() potrebujemo knjižnico time.h
```

2. naloga

```
int main()
{
    srand(time(NULL));
    rand();
    //ustvarimo decimalno število med 0 in 10 in ga na številski osi
    //za 10 mest premaknemo v desno (prištejemo 10):
    double stevilo = ((double)rand() / RAND_MAX) * 10 + 10;
    stevilo = round(stevilo * 10) / 10; //zaokrožimo na 1 decimalko
    cout << "Stevilo: " << stevilo;
    cin >> stevilo;
    return 0;
}
```

3. naloga

```
int main()
{
    srand(time(NULL)); //inicializacija generatorja naključnih števil
    char crka = rand() % 26 + 65; //ustvarimo naključno veliko črko
    cout << "Crka: " << crka << endl;
    crka = crka + 32;
    cout << "Crka + 32: " << crka << endl;
    cin >> crka;
    return 0;
}
//Če kodo ASCII pri velikih črkah povečamo za 32, dobimo malo črko.
```

4. naloga

```
#define _USE_MATH_DEFINES //Ti dve vrstici potrebujemo,
#include <math.h> //da lahko uporabimo konstanto M_PI.
int main()
{
    int r = 3, v = 5; //polmer osnovne ploskve in višina
    //POVRŠINA VALJA:
    double P = 2 * (M_PI * pow(r, 2)) + 2 * M_PI * r * v;
    cout << "Povrsina valja: " << P << endl;
    //PROSTORNINA VALJA:
    double V = M_PI * pow(r, 2) * v;
    cout << "Prostornina valja: " << V;
    cin >> r;
    return 0;
}
```

5. naloga

```
int main()
{
    int a, b;
    cout << "Vnesi stevilo a: ";
    cin >> a;
    cout << "Vnesi stevilo b: ";
    cin >> b;
    double rez = sqrt(a + pow(a - b, 2));
    cout << "Rezultat: " << round(rez * 1000) / 1000;
    cin >> a;
    return 0;
}
//Rezultat smo zaokrožili kar v stavku za izpis.
```

Pogojni stavek

1. naloga

```
int main()
{
    int stevilo;
    cout << "Vnesi stevilo: ";
    cin >> stevilo;
    if (stevilo >= 100 && stevilo < 200)
    {
        cout << "Stevilo ustreza.";
    }
    else
    {
        cout << "Stevilo ne ustreza.";
    }
    cin >> stevilo;
    return 0;
}
```

2. naloga

```
int main()
{
    int stevilo;
    cout << "Vnesi stevilo: ";
    cin >> stevilo;
    if (stevilo % 5 == 0) //preverimo, če je po deljenju s 5 ostanek enak 0
    {
        cout << "Stevilo " << stevilo << " je večkratnik stevila 5.";
    }
    else
    {
        cout << "Stevilo " << stevilo << " ni večkratnik stevila 5.";
    }
    cin >> stevilo;
    return 0;
}
```

3. naloga

```
int main()
{
    int prvo, drugo;
    cout << "Vnesi prvo stevilo: ";
    cin >> prvo;
    cout << "Vnesi drugo stevilo: ";
    cin >> drugo;
    if (prvo >= drugo)    //preverimo, če je prvo večje ali enako kot drugo
    {
        if (prvo != drugo)    //preverimo, če sta števili različni
        {
            cout << "Prvo število je vecje.";
        }
        else
        {
            cout << "Števili sta enaki.";
        }
    }
    else
    {
        cout << "Drugo stevilo je vecje.";
    }
    cin >> prvo;
    return 0;
}
```

4. naloga

```
int main()
{
    cout << "Vnesi znak: ";
    char znak = _getch();
    if (znak >= 'A' && znak <= 'Z')    //preverimo, če se znak nahaja med črkama A in Z
    {
        cout << "\nZnak " << znak << " je velika crka.";
    }
    else if (znak >= 'a' && znak <= 'z')    //če ni, preverimo še za male črke
    {
        cout << "\nZnak " << znak << " je mela crka.";
    }
    else    //če ni ne eno in nedrugo, smo vnesli nek drug simbol
    {
        cout << "\nZnak " << znak << " ni crka, je nek drug simbol ali stevka.";
    }
    cin >> znak;
    return 0;
}
//Metoda getch potrebuje knjižnico conio.h
```

Zanki while in do-while

1. naloga

```
int main()
{
    int veckratnik = 6;    //vrednost prvega večkratnika
    while (veckratnik < 100)    //ponavljaj, dokler je večkratnik manjši od 100
    {
        cout << veckratnik << ", "; //izpis,
        veckratnik = veckratnik + 6;    //nato večkratnik povečamo za 6
    }
    cin >> veckratnik;
    return 0;
}
//Izpis bi bil brez zadnje vejice, če bi zanko zaključili s predzadnjim
//večkratnikom, zadnjega pa bi izpisali posebej brez vejice:
int main()
{
    int veckratnik = 6;
    while (veckratnik < 100 - 6)    //mejo za konec ponavljanja zmanjšamo za en
                                    //večkratnik
    {
        cout << veckratnik << ", ";
        veckratnik = veckratnik + 6;
    }
    cout << veckratnik;    //izpis zadnjega večkratnika brez vejice
    cin >> veckratnik;
    return 0;
}
```

2. naloga

```
int main()
{
    char crka = 'A';           //nastavimo prvo črko
    while (crka <= 'Z')       //zanka se ponavlja do črke Z
    {
        cout << crka << " ";
        crka++;               //gremo na naslednjo črko v tabeli ASCII
    }
    cout << endl;             //gremo v novo vrsto
    //postopek ponovimo še za male črke:
    crka = 'a';
    while (crka <= 'z')
    {
        cout << crka << " ";
        crka++;
    }
    cin >> crka;
    return 0;
}
```

3. naloga

```
int main()
{
    int i = 0, vsota = 0;      //postavimo začetno vrednost števca in vsote
    int stevilo;
    while (i < 7)
    {
        cout << "Vnesi " << i + 1 << ". stevilo: ";
        cin >> stevilo;
        vsota = vsota + stevilo; //obstoječi vsoti prištejemo prebrano število
        i++;                     //povečamo števec za 1
    }
    cout << "Vsota prebranih števil je " << vsota << endl;
    double povprecje = vsota / 7.0;
    cout << "Povprecje prebranih števil je " << round(povprecje * 1000) / 1000;
    cin >> i;
    return 0;
}
```


4. naloga

```
int main()
{
    srand(time(NULL));
    int i = 20;           //števec za števila
    int soda = 0;       //števec za soda števila
    while (i > 0)       //ponavljaj, dokler je števec večji od 0
    {
        int stevilo = rand() % 90 + 10; //naključno dvomestno število
        if (stevilo % 2 == 0)           //preveri, če je število sodo
        {
            soda++;                     //števec sodih števil zvečamo za 1
        }
        i--;                            //števec števil zmanjšamo za 1
    }
    cout << "Sodih števil je bilo: " << soda;
    cin >> i;
    return 0;
}
```

5. naloga

```
//RAZLIČICA S POGOJNIM STAVKOM:
```

```
int main()
{
    int i = 1; //začnemo z 1, ker je 0 deljiva z 12 in bi šli takoj v novo vrsto
    while (i <= 120)
    {
        cout << "*";
        if (i % 12 == 0) //če je števec deljiv z 12
        {
            cout << endl; //skočimo v novo vrsto
        }
        i++;
    }
    cin >> i;
    return 0;
}
```

```
//RAZLIČICA Z GNEZDENO ZANKO:
```

```
int main()
{
    int i = 0, k = 0; //začnejo z 1, ker je 0 deljiva z 12 in bi šli takoj v
                    //novo vrsto
    while (i < 10) //ta zanka ponavlja izpis vrstice
    {
        while (k < 12) //ta zanka izpiše vrstico 12 zvezdic
        {
            cout << "*";
            k++; //naslednja zvezdica
        }
        cout << endl;
        i++; //naslednja vrstica
        k = 0; //števec zvezdic v vrstici gre na 0
    }
    cin >> i;
    return 0;
}
```

6. naloga

```
int main()
{
    int r, a, b;
    while (true)
    {
        cout << "IZBERI:" << endl;
        cout << "a-krog\nb-kvadrat\nc-pravokotnik\nd-izhod" << endl;
        char izbira = _getch();
        if (izbira == 'a')
        {
            cout << "Vnesi polmer kroga: ";
            cin >> r;
            double pl = M_PI * pow(r, 2);
            cout << "Polscina kroga: " << round(pl * 1000) / 1000 << endl;
            cout << "Pritisni ENTER na nadaljevanje!" << endl;
            _getch(); //zaustavimo program pred brisanjem, da lahko
                    //preberemo vsebino konzole
            system("cls"); //brisanje konzole
        }
        if (izbira == 'b')
        {
            cout << "Vnesi stranico kvadrata: ";
            cin >> a;
            double pl = pow(a, 2);
            cout << "Polscina kvadrata: " << pl << endl;
            cout << "Pritisni ENTER na nadaljevanje!" << endl;
            _getch();
            system("cls");
        }
        if (izbira == 'c')
        {
            cout << "Vnesi stranico a: ";
            cin >> a;
            cout << "Vnesi stranico b: ";
            cin >> b;
            double pl = a * b;
            cout << "Polscina pravokotnika: " << pl << endl;
            cout << "Pritisni ENTER na nadaljevanje!" << endl;
            _getch();
            system("cls");
        }
        if (izbira == 'd')
        {
            exit(0); //ukaz za končanje programa
        }
    }
    return 0;
}
```

Zanka for

1. naloga

```
int main()
{
    int a;
    for (int st = 1000; st >= 500; st = st - 10)
    {
        cout << st << " ";
    }
    cin >> a;
    return 0;
}
```

2. naloga

```
int main()
{
    int a;
    for (double st = 2.4; st <= 2.5; st = st + 0.01)
    {
        cout << st << " " << endl;
    }
    cin >> a;
    return 0;
}
```

3. naloga

```
int main()
{
    srand(time(NULL));
    int vsota = 0, i = 0;
    for (; i < 10; i++)
    {
        int met = rand() % 6 + 1;
        vsota = vsota + met;
        cout << "Met st. " << i + 1 << ": " << met << " Pritisni ENTER" << endl;
        _getch();
    }
    cout << "Povprečje metov: " << vsota / (double)i;
    cin >> i;
    return 0;
}
```

//Števec i smo deklarirali pred zanko, ker smo ga potrebovali za izračun povprečja.

//Pri izračunu smo ga pretvorili v decimalno število, da dobimo decimalke.

4. naloga

```
int main()
{
    int a;
    for (int n = 0; n < 12; n++)
    {
        double clen = 1 / (1 + pow(n, 2));
        cout << n + 1 << ". clen: " << round(clen * 10000) / 10000 << endl;
    }
    cin >> a;
    return 0;
}
```

Znakovni nizi

1. naloga

```
#include <string> //kadar delamo z znakovnimi nizi, potrebujemo knjižnico string
int main()
{
    string stavek;
    cout << "Vnesi stavek:" << endl;
    getline(cin, stavek); //metoda za vnos znakovnega niza s presledki
    cout << "Število znakov v stavku: " << stavek.length();
    cin >> stavek;
    return 0;
}
```

2. naloga

```
int main()
{
    string stavek;
    cout << "Vnesi stavek:" << endl;
    getline(cin, stavek);
    cout << stavek[0] << " "; //izpis prve črke (ker ta ni za presledkom)
    for (int i = 0; i < stavek.length(); i++)
    {
        if (stavek[i] == ' ') //izpis črk, ki sledijo presledkom
        {
            cout << stavek[i + 1] << " ";
        }
    }
    cin >> stavek;
    return 0;
}
```

3. naloga

```
int main()
{
    int stevec = 0;
    string stavek;
    cout << "Vnesi stavek:" << endl;
    getline(cin, stavek);
    for (int i = 0; i < stavek.length(); i++)
    {
        if (stavek[i] >= '0' && stavek[i] <= '9') //če je trenutni znak števka
        {
            stevec++; //se števec poveča za 1
        }
    }
    cout << "V stavku je " << stevec << " stevk.";
    cin >> stavek;
    return 0;
}
```

4. naloga

```
int main()
{
    string stavek1, stavek2;
    cout << "Vnesi prvi stavek:" << endl;
    getline(cin, stavek1);
    cout << "Vnesi drugi stavek:" << endl;
    getline(cin, stavek2);
    //v pogojnih stavkih primerjamo dolžini obeh stavkov:
    if (stavek1.length() >= stavek2.length())
    {
        if (stavek1.length() != stavek2.length())
        {
            cout << "Prvi stavek je daljsi." << endl;
        }
        else
        {
            cout << "Stavka sta enako dolga." << endl;
        }
    }
    else
    {
        cout << "Drugi stavek je daljsi." << endl;
    }
    cin >> stavek1;
    return 0;
}
```

5. naloga

```
int main()
{
    string dvojisko = "";    //sem bomo shranjevali ostanke po deljenju z 2
    int desetisko;
    cout << "Vnesi desetisko stevilo:" << endl;
    cin >> desetisko;
    while (desetisko > 0)    //ponavljamo postopek, dokler je število večje od 0
    {
        int ost = desetisko % 2;                //poiščemo ostanek po deljenju z 2
        dvojisko = to_string(ost) + dvojisko;    //in ga z leve strani dodamo v niz
        desetisko = desetisko / 2;              //število delimo z 2
    }
    cout << "Dvojisko stevilo: " << dvojisko;
    cin >> desetisko;
    return 0;
}
//Ostanek v niz dajemo z leve strani, ker moramo ostanke brati od zadnjega proti
//prvem, da dobimo dvojiško število. Pred dodajanjem moramo ostanek spremeniti v
//znakovni niz.
```


Break, continue in switch

1. naloga

```
int main()
{
    short int vsota = 0, stev;    //uporabimo tip short int,
                                //ker vsota ne bo veliko število
    for (;;)                    //neskončna zanka
    {
        cout << "Vnesi stevilo do velikosti 200: ";
        cin >> stev;
        vsota = vsota + stev;
        if (vsota > 200)
        {
            vsota = vsota - stev;    //vrnemo vsoto na prejšnjo vrednost
            break;                  //prekinemo zanko
        }
    }
    cout << "Vsota je: " << vsota;
    cin >> stev;
    return 0;
}
```

2. naloga

```
int main()
{
    //Pripravimo dve spremenljivki, prva je pravo geslo, ki je skrito v
    // programski kodi, druga pa geslo, ki ga vnese uporabnik:
    string geslo = "Geslo123", vnos = "";
    //VNOS GESLA:
    cout << "Vnesi geslo:" << endl;
    for (int i = 0; i<12; i++)          //dovolimo vnos 12 znakov
    {
        char znak = _getch();          //vnesemo znak
        if (znak == 13)                //če je ENTER
        {
            break;                    //konec vnosa
        }
        cout << "*";                  //izpiše *
        vnos = vnos + znak;            //dodamo znak v niz z imenom vnos
    }
    //PREVERIMO VNESENO GESLO:
    if (geslo == vnos)
    {
        cout << "\nGeslo je pravilno." << endl;
    }
    else
    {
        cout << "\nGeslo ni pravilno." << endl;
    }
    cin >> geslo;
    return 0;
}
```

3. naloga

```
int main()
{
    double rezultat = 0;
    int a, b;
    cout << "Program za izračun ploscine, obsega, in diagonale pravokotnika" << endl;
    cout << "Vnesi dolžino stranice a: ";
    cin >> a;
    cout << "Vnesi dolzino stranice b: ";
    cin >> b;
    cout << "IZBERI:" << endl;
    cout << "A-Obseg\nB-Ploščina\nC-Diagonala" << endl;
    char izbira = _getch();
    switch (izbira)    //preverimo, kaj je uporabnik izbral in tista možnost se izvede
    {
        case 'A':
            rezultat = 2 * (a + b);
            cout << "\nObseg pravokotnika: ";
            break;
        case 'B':
            rezultat = a * b;
            cout << "\nPloščina pravokotnika: ";
            break;
        case 'C':
            rezultat = round(sqrt(pow(a, 2) + pow(b, 2)) * 100) / 100;
            cout << "\nDiagonala pravokotnika: ";
            break;
        default: cout << "Napacna izbira!! Si uporabil veliko crko?"; break;
    }
    cout << endl << rezultat;
    cin >> rezultat;
    return 0;
}
```

Napredna raba metod

1. naloga

```
char metoda(int a, int b, string s);    //napoved metode
//GLAVNA METODA:
int main()
{
    double rezultat = 0;
    char znak = metoda(11, 22, "neko besedilo");    //klic metode
    cin >> rezultat;
    return 0;
}
//METODA:
char metoda(int a, int b, string s)
{
    return 'x';    //vrnemo neko znakovno vrednost
}
```

2. naloga

```
void metoda1();    //napoved metode1
void metoda2();
//GLAVNA METODA:
int main()
{
    int x;
    metoda1();    //klic prve metode
    cin>>x;
    return 0;
}
//METODA1:
void metoda1()
{
    cout << "Metoda1" << endl;
    metoda2();    //klic druge metode
}
```

3. naloga

```
double vzporedno(int R1, int R2);
double zaporedno(int R1, int R2);
//GLAVNA METODA:
int main()
{
    int R1, R2;
    cout << "Vnesi upornost za R1: ";
    cin >> R1;
    cout << "Vnesi upornost za R2: ";
    cin >> R2;
    cout << "Skupna upornost je " << zaporedno(R1, R2) << " ohm." << endl;
    cout << "Skupna upornost je " << vzporedno(R1, R2) << " ohm." << endl;
    cin >> R1;
}
//ZAPOREDNA VEZAVA:
double zaporedno(int R1, int R2)
{
    cout << "Zaporedna vezava:" << endl;
    return R1 + R2;
}
//VZPOREDNA VEZAVA:
double vzporedno(int R1, int R2)
{
    cout << "Vzporedna vezava:" << endl;
    return R1 * R2 / (double)(R1 + R2);
}
```

4. naloga

```
void nakljucna_stevila(int spodnja, int zgornja);
//GLAVNA METODA:
int main()
{
    int prvo, drugo;
    cout << "Vnesi prvo stevilo: ";
    cin >> prvo;
    cout << "Vnesi drugo stevilo: ";
    cin >> drugo;
    if (prvo < drugo)
    {
        nakljucna_stevila(prvo, drugo); //klic metode
    }
    else
    {
        cout << "Napacen vnos!!" << endl;
    }
    cin >> prvo;
}
//METODA ZA IZPIS NAKLJUČNIH VREDNOSTI:
void nakljucna_stevila(int spodnja, int zgornja)
{
    srand(time(NULL));
    for (int i = 0; i<10; i++)
    {
        cout << (rand() % (zgornja - spodnja) + spodnja) << " ";
    }
}
```

Večvrednostne spremenljivke

1. naloga

```
int main()
{
    char male_crke[25];           //tabela za 25 znakov
    return 0;
}
```

2. naloga

```
int main()
{
    int x;
    int veckratniki[33];         //tabela za 33 celih števil
    for (int i = 0; i < 33; i++)
    {
        veckratniki[i] = 3 * (i + 1);
    }
    cin >> x;
    return 0;
}
//Indeks se vedno začne pri številu 0, zato je začetna vrednost števca
//v zanki enaka 0. V izrazu za izračun večkratnika števec povečamo za 1,
//da se večkratniki začnejo s številom 3.
```

3. naloga

```
int main()
{
    int x;
    string dnevi[] = { "pon", "tor", "sre", "cet", "pet", "sob", "ned" };
    for (int i = 0; i < size(dnevi); i++)
    {
        cout << dnevi[i] << endl;
    }
    cin >> x;
    return 0;
}
```

4. naloga

```
int main()
{
    int x;
    srand(time(NULL));
    double stevila[50];
    for (int i = 0; i < size(stevila); i++) //ustvarimo števila med 1000 in 2000
    {
        stevila[i] = (rand() % 1000 + 1000) / 100.0; //pri deljenju s 100
                                                    //dobimo 2 decimalki
    }
    //Decimalne dele dobimo tako, da številu odštejemo celi del:
    double vsota = 0;
    for (int i = 0; i < size(stevila); i++)
    {
        vsota = vsota + (stevila[i] - (int)stevila[i]);
    }
    cout << "Vsota decimalnih delov: " << vsota << endl;
    cout << "Povprečje decimalnih delov: " << vsota / size(stevila);
    cin >> x;
    return 0;
}
```


5. naloga

```
int main()
{
    int x;
    char znaki[100]; //predvidimo, da bo stavek krajši od 100 znakov
    srand(time(NULL));
    string samog = "aeiou"; //iz tega niza naključno jemljemo samoglasnike
    string stavek;
    cout << "Vnesi stavek:" << endl;
    getline(cin, stavek);
    strcpy_s(znaki, stavek.c_str()); //prenos stavka v znakovno tabelo
    for (int i = 0; i < size(znaki); i++)
    {
        switch (znaki[i])
        {
            case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u': znaki[i] = samog[rand() % 5]; break;
        }
    }
    cout << "Dobljeni stavek:" << znaki;
    cin >> x;
    return 0;
}
//V stavek switch vstopimo na katerikoli samoglasnik, izstopimo
//pa s stavkom break.
```

Večdimenzionalne tabele

1. naloga

```
int main()
{
    int a = 1;
    int stevila[5][8];
    //VNOS VREDNOSTI V TABELO:
    for (int i = 0; i < size(stevila); i++)
    {
        for (int k = 0; k < size(stevila[0]); k++)
        {
            stevila[i][k] = a;           //vnesemo vrednost
            a++;                         //povečamo vrednost za 1
        }
    }
    //IZPIS VSEBINE TABELE V KONZOLO:
    for (int i = 0; i < size(stevila); i++)
    {
        for (int k = 0; k < size(stevila[0]); k++)
        {
            cout << setw(3) << stevila[i][k];
        }
        cout<<endl;    //gremo v novo vrsto
    }
    cin >> a;
    return 0;
}
//Metoda setw() določi širino izpisa za posamezno število. Na ta način
//uredimo stolpce. Nahaja se v knjižnici iomanip.
```

2. naloga

```
int main()
{
    int vrstica, stolpec;
    string besede[3][5] = { { "avto", "vlak", "kolo", "barka", "letalo" },
        { "figa", "hruška", "breskev", "sliva", "češnja" },
        { "pes", "mačka", "miš", "srna", "zajec" } };
    cout << "Vnesite številko vrstice od 1 do 3: ";
    cin >> vrstica;
    cout << "Vnesite številko stolpca od 1 do 5: ";
    cin >> stolpec;
    cout << "Izbrana beseda: " << besede[vrstica - 1][stolpec - 1];
    cin >> vrstica;
    return 0;
}
//Pri obeh indeksih izbrane besede odštejemo 1, ker se indeksi začnejo z ničlo.
```

3. naloga

```
int main()
{
    char znaki[3][5];
    //VNOS ZNAKOV:
    cout << "Vnesi 15 znakov!";
    for (int i = 0; i < size(znaki); i++)
    {
        for (int k = 0; k < size(znaki[0]); k++)
        {
            znaki[i][k] = _getch();
        }
    }
    //PRENOS ZNAKOV V NIZ:
    string nizznakov = ""; //prazen niz
    for (int i = 0; i < size(znaki); i++)
    {
        for (int k = 0; k < size(znaki[0]); k++)
        {
            nizznakov = nizznakov + znaki[i][k];
        }
    }
    cout<<"\nDobljeni niz znakov: " << nizznakov;
    cin >> nizznakov;
    return 0;
}
```

4. naloga

```
int main()
{
    char crke[10][10];
    //VNOS ČRK V TABELO:
    char crka = 'A'; //pripravimo prvo črko
    for (int i = 0; i < size(crke); i++)
    {
        crke[i][i] = crka++; //vnesemo črko in se premaknemo na naslednjo
    }
    //PRIKAZ TABELE V KONZOLI:
    for (int i = 0; i < size(crke); i++)
    {
        for (int k = 0; k < size(crke[0]); k++)
        {
            cout << setw(2) << crke[i][k];
        }
        cout << endl;
    }
    cin >> crka;
    return 0;
}
//Pri vnosu v tabelo zadostuje le ena zanka, ker sta indeksa diagonale enaka.
//Pri izpisu vidimo le diagonalo, ostale vrednosti so 0,
//v tabeli ASCII kodi 0 ne pripada noben znak.
```