Низови

Низови су сложени типови података и базирају се на осталим типовима промењивих.
Низови представљају скупове коначног броја елемената истог типа обележене једним именом.
Сваки податак у низу јесте елемент низа и њему се може приступити преко његове позиције у низу.
Позиција елемента у низу је одређена његовим индексом.
Индекс је целобројна вредност и први елемент у низу има индекс 0.
Елементи низа се увек смештају у повезаним меморијским локацијама.
Низ може бити једнодимензионалан, дводиемзионалан (матрица) или вишедимензионалан (коцка).
Код једнодимензионалног низа димензија је и дужина низа (низ је димензије n).
Код вишедимензионалних низова се каже да је низ димензија m x n x z.

Декларација и иницијализација низова

Декларација једнодимензионалног низа : int a[3] //не може int n = 3; int a[n];
Декларација матрице: int a[10] [10]; //низ а димензија 10x10

Пример 01: Декларисати низ од 10 елемената типа целих бројева и сваком елементу низа доделити вредност његовог индекса
int a[10], i;
for(i = 0; i < 10; i++) a[i] = i;
на излазу се добија: a[0] = 0, a[1] = 1…a[9] = 9

Иницијализација низа: int a[5] = {3, 10, -5, 4, 2}; float b[3] = {-4.89, 102.5678, -34.45}; char s[2] = {’x’, ’y’};

Приступ елементима низа

Приступ елементима низа се врши преко њиховог индекса. Ако низ има n елемената, индекси тог низа су од 0 до n-1. Зато четврти елемент низа има индекс 3.

Пример 02: Сабрати други, четврти и шести елемент низа 2, 4, -3, -5, -7, 3, 8.

```
// Sabira 2, 4 i 6 element niza
#include<iostream>
using namespace std;

int zbir(int b[])
{
	int i = 0, suma = 0;
	for (i = 0; i < 6; i++)
	{
		if ((i == 1) || (i == 3) || (i == 5)) suma += b[i];
	}
	return suma;
}

int main()
{
	int a[7] = { 2, 4, -3, -5, -7, 3, 8 };
	cout << "Zbir zeljenih elemenata niza je " << zbir(a) << endl;
	return 0;
}
```

Пример 03: Заменити места првом и четвртом елементу низа 3, 0, 1, 5.

```
// Zamena prvog i cetvrtog elementa niza
#include<iostream>
void zamena(int a[]);
using namespace std;

int main()
{
	int x[4] = {0, 3, 1, 5};
	for (int i = 0; i <= 3; i++) cout << x[i];
	cout << endl;
	zamena(x);
	return 0;
}
void zamena(int a[])
{
	int s = a[0];
	a[0] = a[3];
	a[3] = s;
	for (int i = 0; i <= 3; i++) cout << a[i];
	cout << endl;
}
```

Вишедимензионални низови

Када низ за своје елементе има друге низове, такав низ се назива вишедимензионални низ:
tip_niza ime_niza [dimenzija_1] [dimenzija_2] …[dimenzija_n];
Од вишедимензионалних низова најчешће се користе дводимензионални низови или матрице.

Дводимензионални низови

Матрице се састоје од одређеног броја врста и колона. Увек се прво наводи вредност за ред па вредност за колону.
tip_matrice ime_matrice [duzina_vrste] [duzina_kolone];
int matrica[2][3] //матрица 2 x 3, са 2 врсте и 3 колоне

Обележавање елемената матрице

Елементи матрице имају два индекса: A [i] [j], где i означава врсту а j означава колону у којој се налази одређени елемент матрице.

Унос и приказ елемената матрице

```
int A[2] [3];
for (i = 0; i < 2; i++)
	for (j = 0; j < 3; j++)
		cin >> A[i] [j];
```

```
int A[2] [3];
for (i = 0; i < 2; i++)
	for (j = 0; j < 3; j++)
		cout << A[i] [j];
```

Иницијализација матрица

Пример:
int A[2] [3] = {{1, 2, 3}, {4, 5, 6}}; //матрица A је иницијализовала прву па другу врсту

Ако се у некој димензији не наведу сви елементи, онда се аутоматски дописују 0:
int A[2] [3] = {{1, 2}, {4}}; // је исто што и int A[2] [3] = {{1, 2, 0}, {4, 0, 0}};

Пример 01: Написати програм којим се уносе елементи у матрицу mxn (m,n < 100)а затим и приказати матрицу.

```
// unos elemenata mxn (<100) i prikaz elemenata matrice
#include<iostream>
using namespace std;

int main()
{
	int a[100][100], m, n, i, j;
	cout << "Uneti dimenzije matrice: ";
	cin >> m >> n;
	cout << endl << "Uneti elemente matrice:" << endl;
	for (i = 0; i < m; i++)
	{
		for (j = 0; j < n; j++)
		{
			cout << "A(" << i << " , " << j << ") = ";
			cin >> a[i][j];
		}
	}
	cout << endl << "Elementi matrice su:" << endl;
	for (i = 0; i < m; i++)
	{
		for (j = 0; j < n; j++) cout << a [i] [j] << "\t";
		cout << endl;
	}
	return 0;
}
```

```
Uneti dimenzije matrice: 2 3

Uneti elemente matrice:
A(0 , 0) = 4
A(0 , 1) = 6
A(0 , 2) = -3
A(1 , 0) = 9
A(1 , 1) = 0
A(1 , 2) = 0

Elementi matrice su:
4       6       -3
9       0       0
Press any key to continue . . .
```

Приступање елементима матрице

Приступање елементима вишедимензионалног низа се врши преко индекса: A[0] [0] је адреса првог елемента матрице.

A[1] [0] је адреса елемента који се налази у другој врсти и првој колони матрице.

A[0] [1] је адреса елемента који се налази у првој врсти и другој колони матрице.

Квадратна матрица

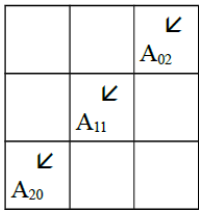
Посебна врста матрице је квадратна матрица. Она има исти број врста и колона. Код ње постоје и главна и помоћна дијагонала. Дијагонала матрице дели матрицу на горњи и доњи троугао. Индекси елемената на главној дијагонали су исти.

За горњи троугао увек важи да је i<j

За доњи троугао увек важи да је i>j

A_{00}		
	A_{11}	
		A_{22}

Индекси елемената на помоћној дијагонали имају збир једнак n-1, где је n број врста и колона.
Изнад помоћне дијагонале однос је i+j < n-1
Испод помоћне дијагонале однос је i+j > n-1



Пример 01: Унет је природан број n. Формирати квадратну матрицу A реда n у којој су сви елементи на главној дијагонали једнаки 1, изнад ње једнаки 2 а испод ње једнаки 3.

```
// Kvadratna matrica nxn, na glavnoj dijag 1, iznad 2, ispod 3
#include<iostream>
using namespace std;

int main()
{
	int a[20][20], n, i, j;
	cout << "Uneti dimenzije matrice: ";
	cin >> n;
	for (i = 0; i < n; i++)
	{
		for (j = 0; j < n; j++)
		{
			if (i < j) a[i][j] = 2;
			if (i == j) a[i][j] = 1;
			if (i > j) a[i][j] = 3;
		}
	}
	cout << endl << "Elementi matrice su: " << endl;
	for (i = 0; i < n; i++)
	{
		for (j = 0; j < n; j++) cout << a[i][j] << "\t";
		cout << endl;
	}
	return 0;
}
```

```
Uneti dimenzije matrice: 4

Elementi matrice su:
1       2       2       2
3       1       2       2
3       3       1       2
3       3       3       1
Press any key to continue . . .
```

Матрице

Матрице могу бити параметри функција али не могу бити вредности функција. Када се користи матрица као параметар функције не наводи се прва димензија матрице али све остале димензије се наводе.
int matrica(int a[] [10], int n);
Међутим, чак иако се наведе број редова као димензија, компајлер ће игнорисати тај податак.
Приликом позивања функције, када се матрица прослеђује као стварни параметар, треба навести само име низа без димензија: matrica(a, n);

Пример: Написати дефиницију функције за унос матрице целих бројева највећих димензија 100 x 100

```
void unos_matriceN (int a[] [100], int m, int n)
{
	int i, j;
	cout << ”Unesi elemente matrice po redovima: ” << endl;
	for(i = 0; i < n; i++)
	{
		cout << i + 1 << ”. red” << endl;
		for (j = 0; j < m; j++)  cin >> a [i] [j];
	}
}
```

Пример: Написати дефиницију функције за приказ матрице целих бројева највећих димензија 100 x 100

```
void prikaz_matriceN (int a[] [100], int m, int n)
{
	int i, j;
	cout << endl;
	cout << ”Elementi matrice su: ” << endl;
	for(i = 0; i < n; i++)
	{
		for (j = 0; j < m; j++)  cout << a [i] [j] << endl;
		cout << endl;
	}
}
```

Задатак05: Коришћењем функција унети елементе матрице па их приказати.

```
//koriscenjem funkcije uneti elemente matrice a zatim ih i
prikazati
#include <iostream>
using namespace std;
void unos_matrice(int niz[] [100], int, int);
void prikaz_matrice(int niz[][100], int, int);
int main()
{
	int a[100][100], m, n, i, j;
	cout << "Unesi dimenzije matrice" << endl;
	cin >> m >> n;
	unos_matrice(a, m, n);

	cout << endl << "prikaz matrice:" << endl;
	for (i = 0; i < m; i++)
	{
		for (j = 0; j < n; j++)
			cout << a[i][j] << "\t";
		cout << endl;
	}
	return 0;
}
void unos_matrice(int niz[][100], int x, int y)
{
	cout << "Unesi elemente matrice po redovima:\n";
	for (int i = 0; i < x; i++)
	{
		cout << i + 1 << ". red" << endl;
		for (int j = 0; j < y; j++) cin >> niz[i][j];
	}
```

```
Unesi elemente matrice po redovima:
1. red
9
2
3
-6
10
2. red
4
5
-3
-20
1
3. red
0
5
3
4
-2

prikaz matrice:
9       2       3       -6      10
4       5       -3      -20     1
0       5       3       4       -2
Press any key to continue . . .
```

```
}

void prikaz_matrice(int niz[][100], int x, int y)
{
	cout << endl;
	cout << "Elementi matrice su: \n";
	for (int i = 0; i < x; i++)
	{
		for (int j = 0; j < y; j++) cout << niz[i][j] << "\t";
		cout << endl;
	}
}
```