

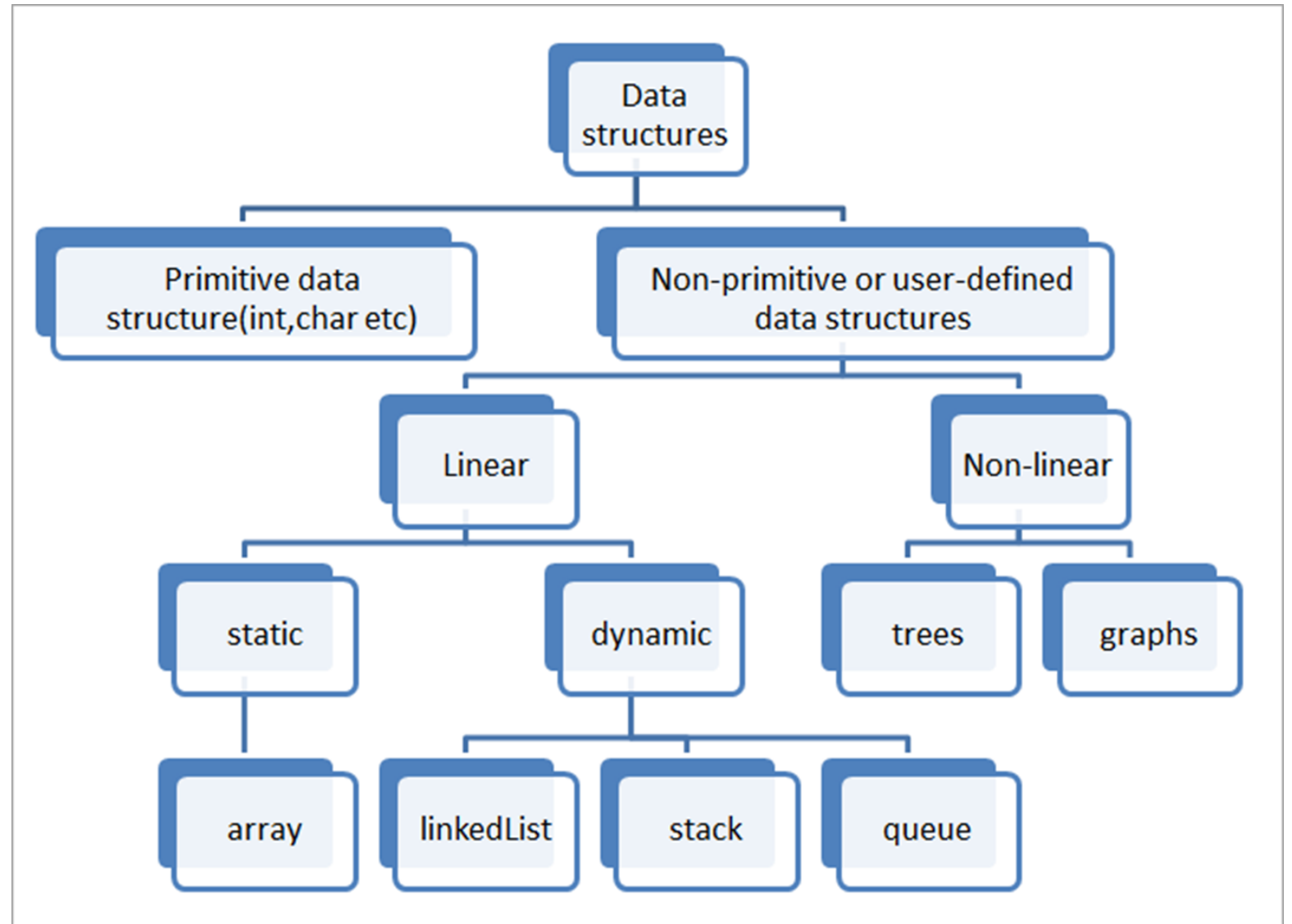
# Arrays

PC Lo6 Arrays and Structures

# Data Structure

## Definition

A data structure is a way of organizing the data. So we can classify data structures as shown into primitive or standard data structures and non-primitive or user-defined data structures.



## Linear structures

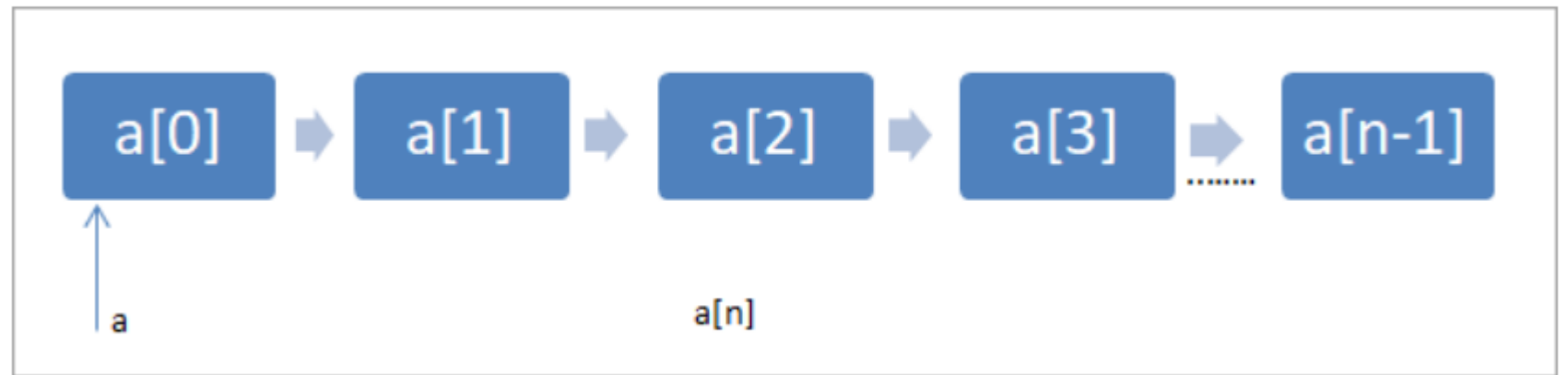
Linear data structures have all their elements arranged in a linear or sequential fashion. Each element in a linear data structure has a predecessor (previous element) and a successor (next element)

Linear data structures are further divided into static and dynamic data structures. Static data structures usually have a fixed size and once their size is declared at compile time it cannot be changed. Dynamic data structures can change their size dynamically and accommodate themselves.

The most popular example of linear static data structure is an array.

# Arrays

An array is a sequential collection of elements of the same type. Each element of the array can be accessed using its position in the array called an index or subscript of the array. The name of the array points to the first element in the array.



## Basic operations

- Declaration
- Values assignment
- Output elements
- Array traversal

## Declaration

General mode:

**<type> <name> <range>;**

**Examples:**

```
int a[20];  
float b[50];  
char z[30];
```

# Values assignment

## Direct assignment

```
<type> <name> [<range>]={val0,    val1,    ...,    valn-1};
```

The values will be assigned consecutively to the elements of the table from left to right. If the number of values is less than the number of items, the last items in the array will not receive default values.

## Examples:

```
int a[5]={7,12,6,45,31};
```

```
float b[5]={4.3,2.7}
```

Values  
assignment

scanf()

For an array, the physical dimensions are fixed. It sets the upper limits for the processed dimensions, the lower ones being flexible. When using dimensions smaller than the set limits - the indicators of the real values for the number of rows and columns is required.



## C tablouri unidimensionale

```
printf("Nr elemente : "); scanf("%d", &n);  
for (i=0;i<n; i++)  
    { printf("Element %d : ",i); scanf("%d", &a[i]); }
```

**pentru compilatoarele on-line partea de mesaje nu este necesară astfel, codul se simplifică:**

```
scanf("%d", &n);  
for (i=0;i<n; i++)  
    scanf("%d", &a[i])
```

Values assingnment

scanf()

Output array  
elements

printf()

## Output in C

horizontal

```
for (i=0;i<n; i++)  
    printf("%d ",a[i]);
```

vertical

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

Example:

sum of array  
elements

```
#include <stdio.h>

int a[100], s=0, n,i;
int main()
{
    // citire date
    printf("Nr elemente : "); scanf("%d", &n);
    for (i=0;i<n; i++)
    { printf("Element %d : ",i); scanf("%d", &a[i]); }
    // procesare date
    for (i=0;i<n; i++) s+=a[i]; // s = s + a[i]
    // afisare rezultat
    printf("Suma %d : ",S);
    return 0;
}
```



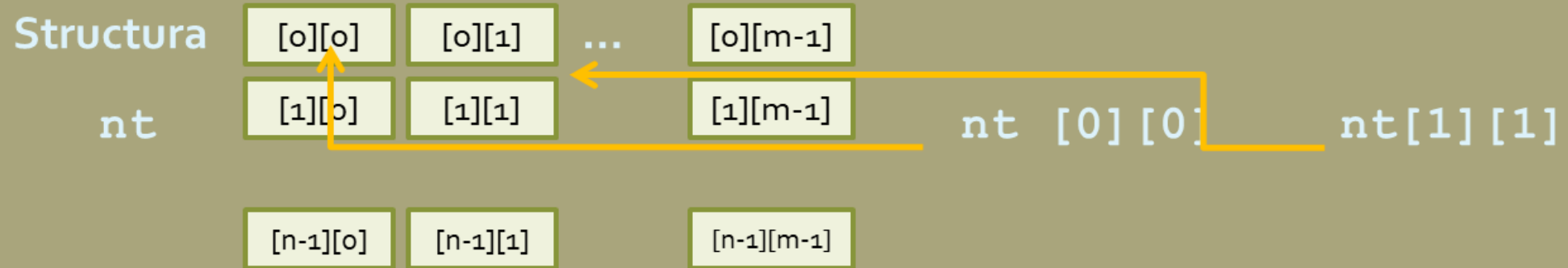
2D arrays

# Description

Tablourile bidimensionale - matrice de linii și coloane, în care numerotarea liniilor și coloanelor începe de la 0, la fel ca și a coloanelor. Prin urmare, pentru un tablou cu N linii și M coloane se va folosi

## Declarare

```
<tip elemente> <nume tablou> [linii N] [coloane M];
```



Elementul cu indicii [I][J] va fi poziționat la intersecția liniei I + 1 cu coloana j + 1.

## Exemple:

```
int a[20][10]; float b[5][10];
```

## Values assignment

### Direct assignment

```
<type> <name> [<range>][<range>] = {val0, val1, ..., valn-1};
```

The values will be assigned consecutively to the elements of the table from left to right, from top to bottom. If the number of values is less than the number of items, the last items in the array will not receive default values.

### Examples:

```
int a[2][2]={7,12,6,45};
```

```
float b[2][3]={4.3,2.7};
```

# Values assignment

## scanf()

2D arrays in C

```
printf("Nr linii, coloane : ");  
scanf("%d%d", &n, &m);  
for (i = 0; i < n; i++)  
    for (j = 0; j < m; j++)  
    {  
        printf("Linie %d, coloana %d  ", i, j);  
        scanf("%d", &a[i][j]);  
    }
```

No lyrics // online compilers:

```
scanf("%d%d", &n, &m);  
for (i = 0; i < n; i++)  
    for (j = 0; j < m; j++)  
        scanf("%d", &a[i][j]);
```

Printing

printf()

```
for (i = 0; i < n; i++)  
{  
    for (j = 0; j < m; j++)  
        printf("%kd", a[i][j]);  
    printf("\n");  
}
```

k – number of printing places



Example:

the average of  
even valued  
elements in a  
2D array

```
#include <stdio.h>
int a[100][100], N,M,i,j,l=0; float S=0;
void main()
{
    // read data
    printf("rows, cols : "); scanf("%d%d", &N, &M);
    for (i=0;i<N; i++)
        for (j = 0;j < M; j++)
        {
            printf("Row %d, col %d  ",i,j);
            scanf("%d", &a[i][j]);
        }
    // data processing
    for (i = 0;i < N; i++)
        for (j = 0;j < M; j++) if (a[i][j] % 2 ==0)
        {S+=a[i][j]; l++;}
    // print result
    printf("Average %f : ",S/l);
}
```

Example:

printing lines  
with maximal  
sum from a 2D  
array

```
#include <stdio.h>

int a[100][100], sum,l,N,M,i,j, b[100];

int main()
{ // citire date
  printf("Nr linii, coloane : "); scanf("%d%d", &N, &M);
  for (i=0;i<N; i++)
    for (j=0;j<M; j++)
{ printf("Linie %d, coloana %d  ",i,j); scanf("%d", &a[i][j]);}
  // procesare date
  for (i=0; i<N; i++)
    { b[i]=0; for (j=0;j<M; j++) b[i]+=a[i][j]; }
  l=sum=0;
  for (i=0; i<N; i++) if (b[i]>sum) {sum=b[i]; l=i;}
  // afisare rezultat
  for (i=0;i<N; i++)
    if b[i]=sum
      {for (j=0;j<M; j++) printf("%kd",a[i][j]); printf("\n") }
}
```