

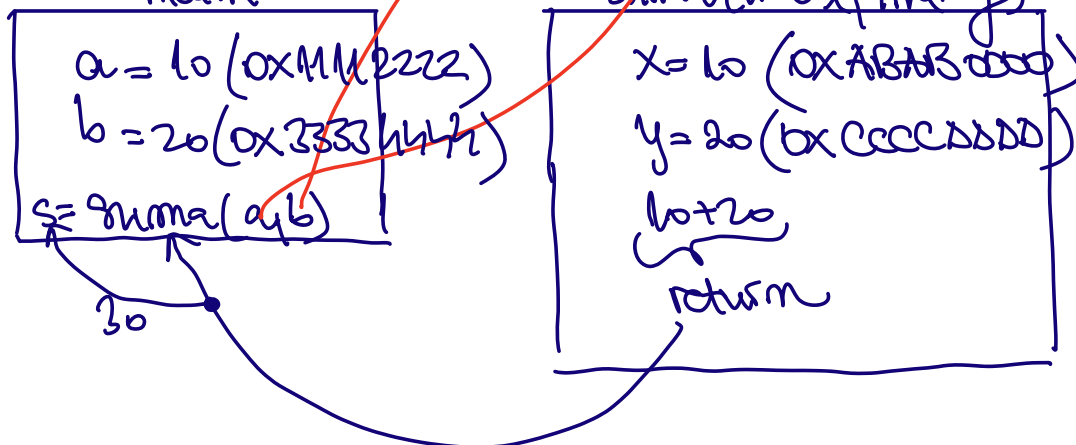
## Coș - continuare

### Transmiterea parametrilor prin adresă

În C/C++ parametrii se transmit implicit prin valoare.

```
int suma(int x, int y)
{
    return x+y;
}
```

```
int main()
{
    int a=10, b=20;
    int s = suma(a, b);
    return 0;
}
```



Se transmite prin adresă pointerii, adică tabloul 1D (vectorul

în matricea (tabloul 2D))

Exercițiu

Să se citească un vector de  $n$  elemente, să se afișeze vectorul

și să se implementeze o funcție care calculează maximumul

și maximumul, revenind cu ambele valori în `main`.

Atât citirea cât și afișarea să fie funcții.

```

int main()
{
int a[100];
    int *n, mx, mn;
    int m; // citire număr de elemente
    printf("n="); scanf("%d", &m);
    n = (int*) malloc(m * sizeof(int));
    citirevector(m, n);
    afisarevector(m, n);
    mx = calculaminoxvector(m, n);
    printf("%d %d", mx, mn);
    return 0;
}

```

// apel de funcții

returnăm cu valoarea maximă  
mx = calculaminoxvector(m, n);

val. minimă  
&mn

re  
prime  
în adresa  
transmisă

0x12341234

Înainte de main se declară funcțiile, cu corp. de  
se implementează, prim valoare

0x12341234

```

void citirevector(int lungime, int* vector)
{
    for (int i=0; i<lungime; ++i)
    {
        printf("Dati elementul: ");
        scanf("%d", &vector[i]);
    }
}

```

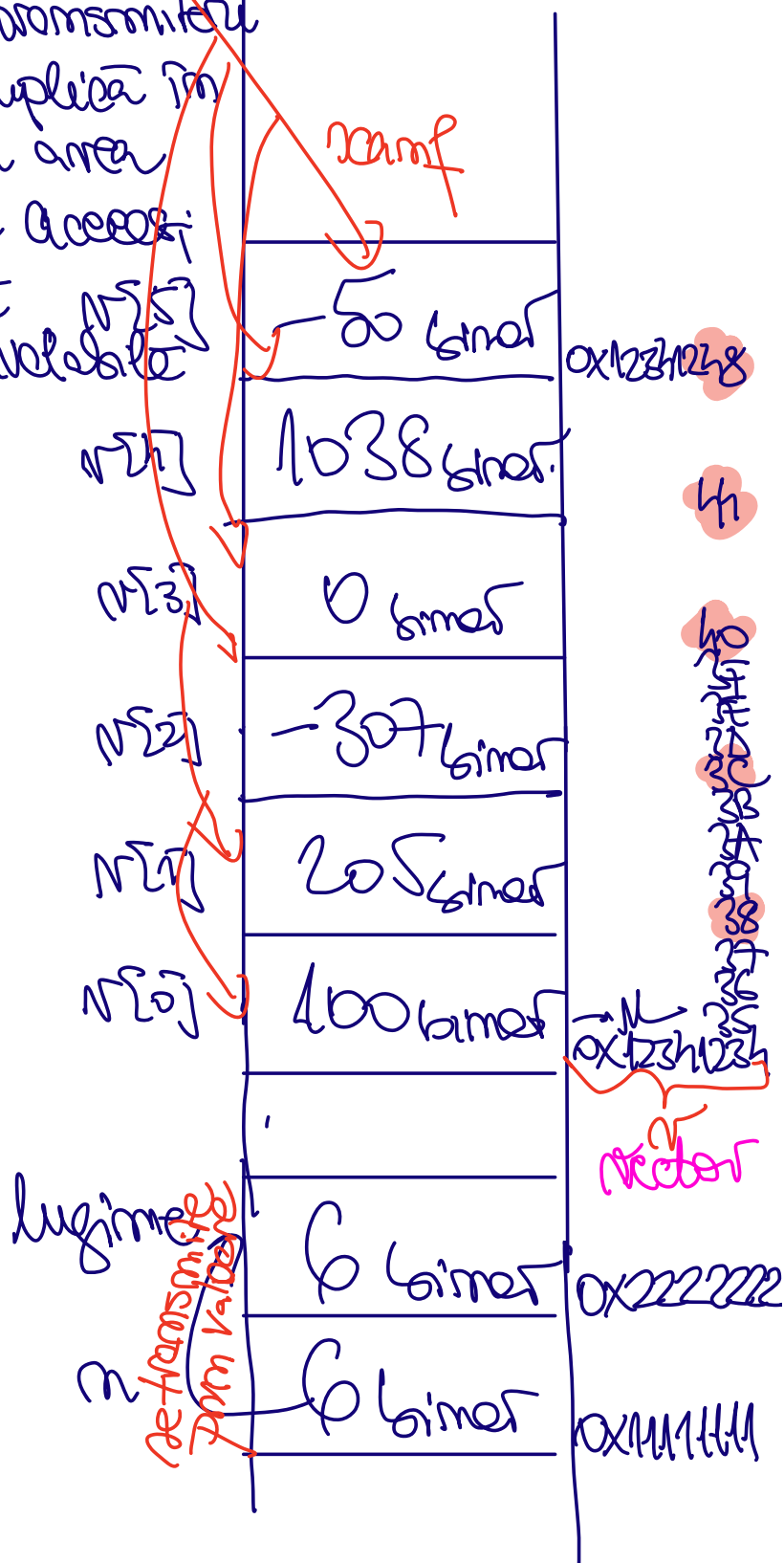
prim adresa

```

void afisarevector(int lungime, int* vector)
{
    for (int i=0; i<lungime; ++i)
        printf("%d", vector[i]);
}

```

Elementele în cazul transmiterii  
 printr-adresă nu se duplicatează în  
 memorie, doar vom avea  
 două denumiri câte o adresă  
 adresă, Una volatile  
 în main, cealaltă volatile  
 în funcție.



int calculmaxminvector (int lungime, int\* v,  
int\* adrmimum)

```
{ int maxim = v[0], minim = v[0];  
  for (int i=1; i<lungime; ++i)  
  {  
    if (v[i] > maxim) maxim = v[i];  
    if (v[i] < minim) minim = v[i];  
  }
```

~~return maxim, minim;~~ → return poate reveni  
cu o singură valoare

\*adrmim = minim;

Continutul adresei nu va val. minimă  
return maxim;

}

void calculmaxminvector (int lungime, int\* v,  
int\* adrmimum,  
int\* adrmmaxim)

```
{ int maxim = v[0], minim = v[0];  
  for (int i=1; i<lungime; ++i)  
  {  
    if (v[i] > maxim) maxim = v[i];  
    if (v[i] < minim) minim = v[i];  
  }
```

~~return maxim, minim;~~ → return poate reveni  
cu o singură valoare

\*adrmim = minim;

Continutul adresei nu va val. minimă

```
*adrimax = maximi;
```

```
}
```

```
calculamxnumvector(n, r, &mm, &mx);
```

mm = -50

1038 = mx

Interschimbare două valori cu funcție implementată

```
void my_swap(int nr1, int nr2)
```

```
{ int tmp = nr1;
```

```
nr1 = nr2;
```

```
nr2 = tmp;
```

```
}
```

• main() → automat se presupune tip de retur int

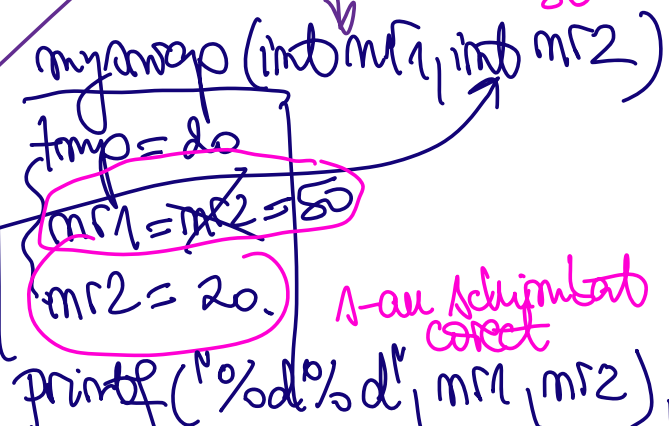
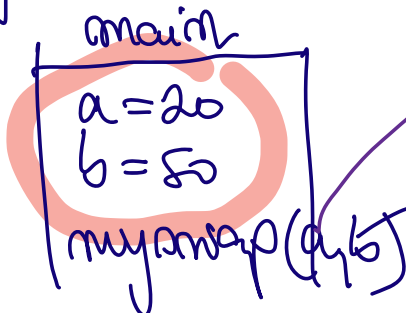
```
{ int a = 20, b = 50;
```

```
my_swap(a, b);
```

```
printf("%d %d", a, b);
```

```
return 0;
```

```
}
```



1-au schimbat corect

```

void myswap (int* adra, int* adr b)
{
    int tmp = *adra; // tmp = a;
    *adra = *adr b; // a = b;
    *adr b = tmp; // b = tmp;
}

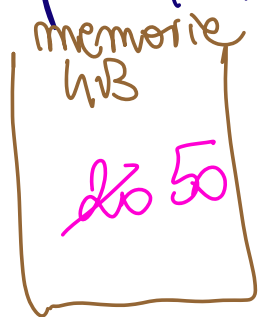
```

$a \rightarrow *adra$   
 $b \rightarrow *adr b$   
 se fac imbucirile  
 potrive

3

adresa lui a, adresa lui b

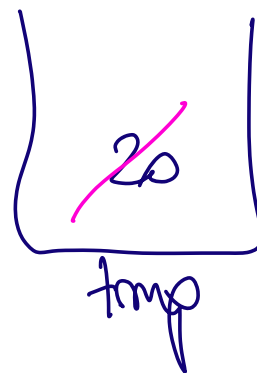
myswap(&a, &b);



a  
0x1111.1111



b  
0x2222.2222



myswap(0x1111.1111, 0x2222.2222)

tmp = \*(0x1111.1111) = 20

\*0x1111.1111 = \*(2222.2222)

Cu a lui a = 50

\*0x2222.2222 = tmp

Cu a lui b = 20

## Completare curs 5:

### Sortarea prin selecție directă:

precare element de pe  $i$  se compară cu toate  
de după el, dacă  $a[i] > a[j] \rightarrow \text{swap}$

$j = [i+1 \dots n]$

ordin incorect

0	1	2	3	4	5
20	8	-3	0	31	7

20 > 8 20 > -3 schimbare

8	20	-3	0	31	7
---	----	----	---	----	---

8 > -3 8 > -3 sch.

-3	8	20	0	31	7
----	---	----	---	----	---

-3 < 0 ✓

-3 < 31 ✓

-3 < 7 ✓

-3	20	8	0	31	7
----	----	---	---	----	---

20 > 8

-3	8	20	0	31	7
----	---	----	---	----	---

8 > 0

-3	0	20	8	31	7
----	---	----	---	----	---

0 < 31 ✓

0 < 7 ✓

-3	0	20	8	31	7
----	---	----	---	----	---

20 > 8 sch.

```
for (i = 0; i < n-1; i++)  
    for (j = i+1; j < n; j++)  
        if (a[i] > a[j])  
            {  
                tmp = a[j];  
                a[j] = a[i];  
                a[i] = tmp;  
            }
```

-3	0	8	20	31	7
----	---	---	----	----	---

$8 < 31$  ✓

$8 > 7$  sch.

-3	0	7	20	31	8
----	---	---	----	----	---

$20 < 31$ ,  $20 > 8$  sch.

-3	0	7	8	31	20
----	---	---	---	----	----

$31 > 20$

-3	0	7	8	20	31
----	---	---	---	----	----