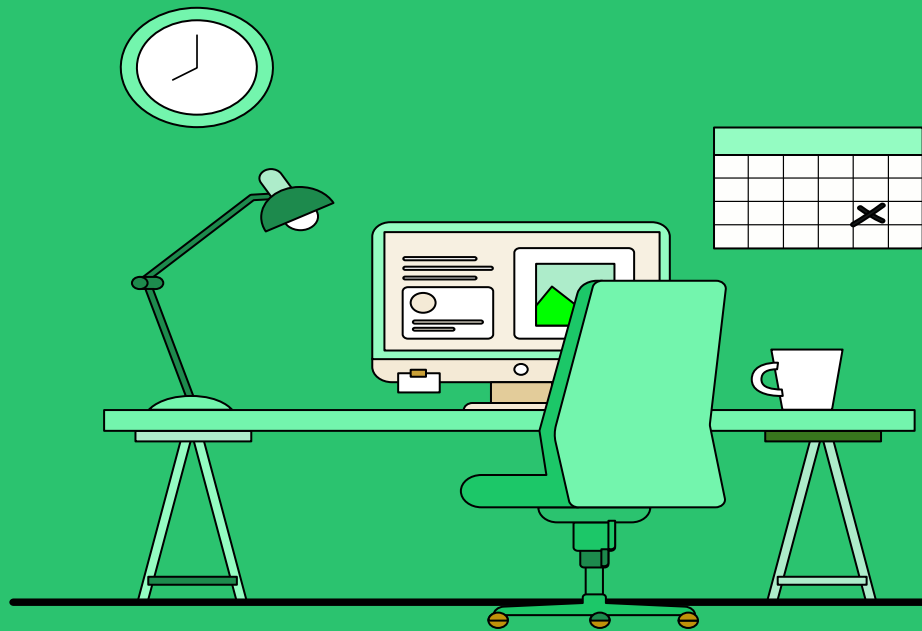


ВКАЗІВНИКИ

C++



ДАНІ В ПАМ'ЯТІ

Оголошення .
Ініціалізація.
Виділення пам'яті.



```
char c = '$';  
int a=5;  
float f=8.5;  
char c='A';
```

char	1
unsigned char	1
short	2
unsigned short	2
enum	2
long	4
unsigned long	4
int	4
unsigned int	4
float	4
double	8

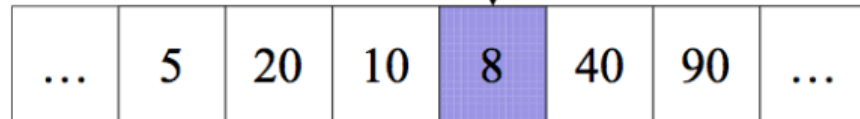
Вказівники

Вказівник - це змінна, значенням якої є адреса пам'яті, по якому зберігається об'єкт певного типу (інша змінна)

Pointer p



```
int x=8;  
int *p;  
p=&x;
```



Memory cells



Приклад 1

```
int * p;
```

p - змінна для
адреси

Приклад 2

```
int vr = 1;  
int* ptr = &vr;
```

ptr зберігає
адресу vr

Операції з вказівниками

**& отримання
адреси змінної**

*** отримання
значення змінної
«розіменування»**

```
float a;  
float * adr_a;  
adr_a = & a;  
a = * adr_a;
```

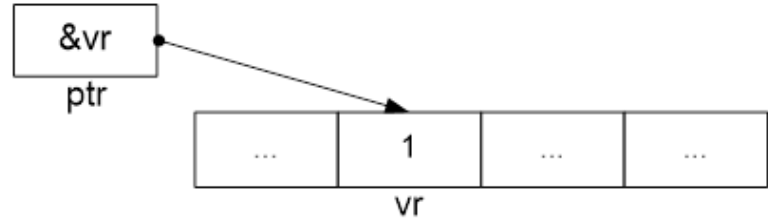
Операції з вказівниками

Адреса та значення

```
int a=5; int *p; p=&a;  
char c; char *pc=&c;  
    cout<<"адреса а "<<p<<"значення а "<< *p<<endl;  
cout<<sizeof(p)<<" "<< sizeof(pc)<<endl;
```

Присвоєння вказівників

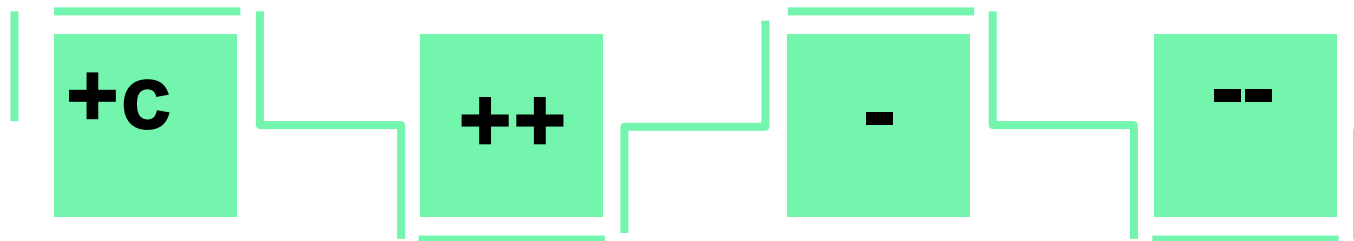
```
#include <iostream>
using namespace std;
int main ()
{float PI = 3.14159, * p1, * p2;
p1 = p2 = & PI;
cout << "За адресою p1 =" << p1 << " зберігається * p1 =" << * p1;
cout << "За адресою p2 =" << p2 << " зберігається * p2 =" << * p2;
return 0;
}
```



ОПЕРАЦІЇ З ВКАЗІВНИКАМИ

інкремент

декремент



додавання і
віднімання
вказівників з
константою

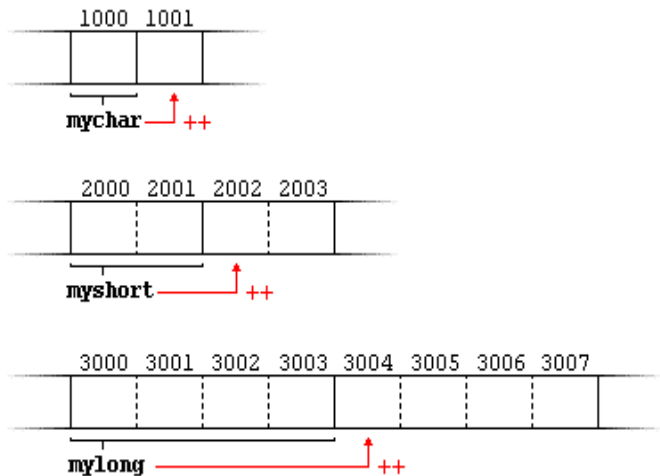
різниця
вказівників

ДОДАВАННЯ **КОНСТАНТИ** ДО ВКАЗІВНИКА

```
#include <iostream>
using namespace std;
int main ()
{
double mas [10] = {1.29, 3.23, 7.98, 5.54, 8.32, 2.48, 7.1};
double * p1;
p1 = &mas[0];    // p1=mas;
cout << * p1 << endl;
p1 = p1 + 3;
cout << * p1 << endl;
return 0;
}
```

ЧАС ДЛЯ КОДУ

Потесстимо код з попереднього слайду, змінивши константу



`++` означає, що ми хочемо перейти до наступної об'єкта в пам'яті, який знаходиться за поточним і на який вказує покажчик.

`--` означає перехід назад до попереднього об'єкта в пам'яті.

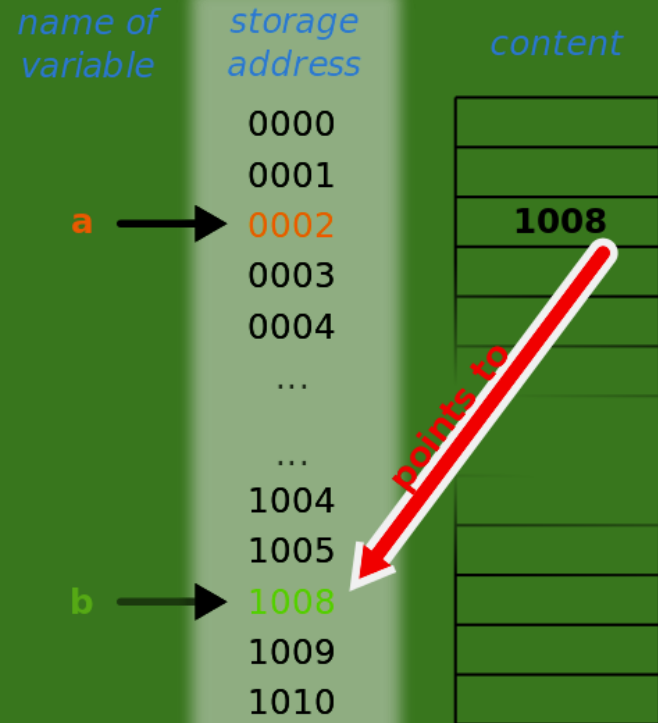
ІНКРЕМЕНТ ТА ДЕКРЕМЕНТ

`ptr++`
`ptr--`



РІЗНИЦЯ ДВОХ ВКАЗІВНИКІВ

Різниця двох вказівників - це різниця їх значень, поділена на розмір типу в байтах.



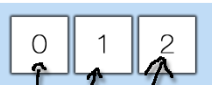
РІЗНИЦЯ ДВОХ ВКАЗІВНИКІВ

```
#include <iostream>
using namespace std;
int main ()
{
    double mas [10] = {1.29, 3.23, 7.98, 5.54, 8.32, 2.48, 7.1};
    double * p1, *p2;
    p1 = & mas [0];
    cout << * p1 << endl;
    p2 = &mas[5];
    cout << p2-p1<< endl;
    return 0;
}
```

ЧАС ДЯ КОДУ

Потесстимо код з попереднього
слайду з різними параметрами

```
int array[3] = {0,1,2};  
int *ptr = array;
```



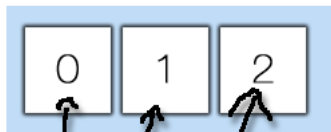
```
*ptr;  
*ptr++;  
*ptr+=2;
```

ВКАЗІВНИКИ ТА МАСИВИ

```
#include <iostream>  
using namespace std;  
int main()  
{  
    double mas [10] = {1.29, 3.23, 7.98, 5.54, 8.32, 2.48, 7.1};  
    double * p1;  
    p1 = & mas [0];  
    cout << * p1 << endl;  
    p1 = p1 + 3;  
    cout << * p1 << endl;  
  
    return 0;  
}
```

ВКАЗІВНИКИ ТА МАСИВИ

```
int array[3] = {0,1,2};  
int *ptr = array;
```



`*ptr;`

`*ptr++;`

`*ptr+=2;`

```
#include <iostream>  
#include <cstdlib>  
#include <time.h>  
using namespace std;  
int main()  
{  
    int mas[10];  
    srand(time(NULL));  
    for(int *p=&mas[0], *p2 = &mas[10]; p <  
p2; p++){  
        *p=rand()%10;  
        cout<<*p<<" "<<p<<" "<<*p+10<<endl;  
    }  
}
```


ЧАС ДЛЯ КОДУ

Напишіть код введення та виведення статичного масиву не використовуючи індекси елементів, а лише вказівники

ВКАЗІВНИКИ ТА РЯДКИ

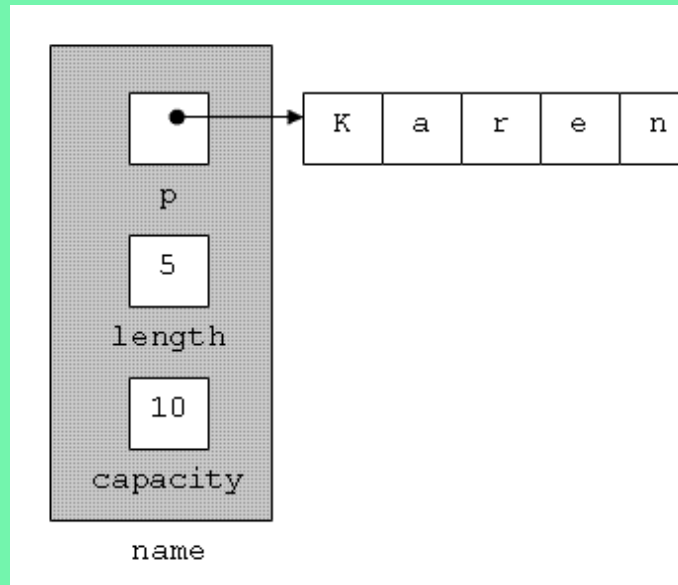
```
#include <iostream>
#include <cstring>
using namespace std;

int main() {

    string s = "Перевірка";
    string * ps;
    ps = &s;
    cout << s <<* ps << endl;

    return 0;

}
```



ВКАЗІВНИКИ ТА РЯДКИ

```
#include <iostream>
#include <cstring>
using namespace std;
```

```
int main() {
```

```
    char s1[]="Finish";
    char *ptr=&s[0];
    for (int i=0; i<strlen(s1);i++, ptr++)
        cout << *ptr << endl;
```

```
        return 0;
```

```
}
```

```
char *str = "Microchip";
```

str



M	i	c	r	o	c	h	i	p	\0
---	---	---	---	---	---	---	---	---	----



```
str += 4
```

ЧАС ДЛЯ КОДУ

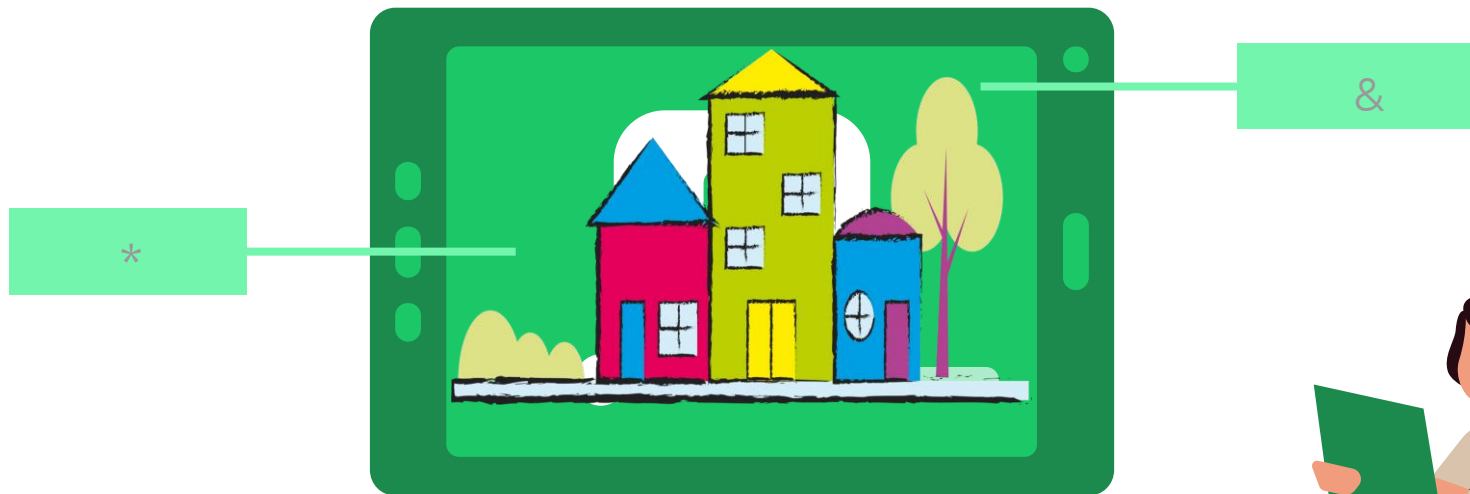
Виведіть рядок з кінця, не використовуючи індекси елементів, а лише вказівники

ПОСИЛАННЯ

```
#include <iostream>
using namespace std;
int main()
{
    int t = 13;
    int &r = t;
    cout << "Було t:" << t; r += 10;
    cout<<"\n Стало t:" << t;
    return 0;
}
```

int &r

ВКАЗІВНИКИ



address

