



# Лекція №5-6

---

Опрацювання одновимірних  
масивів

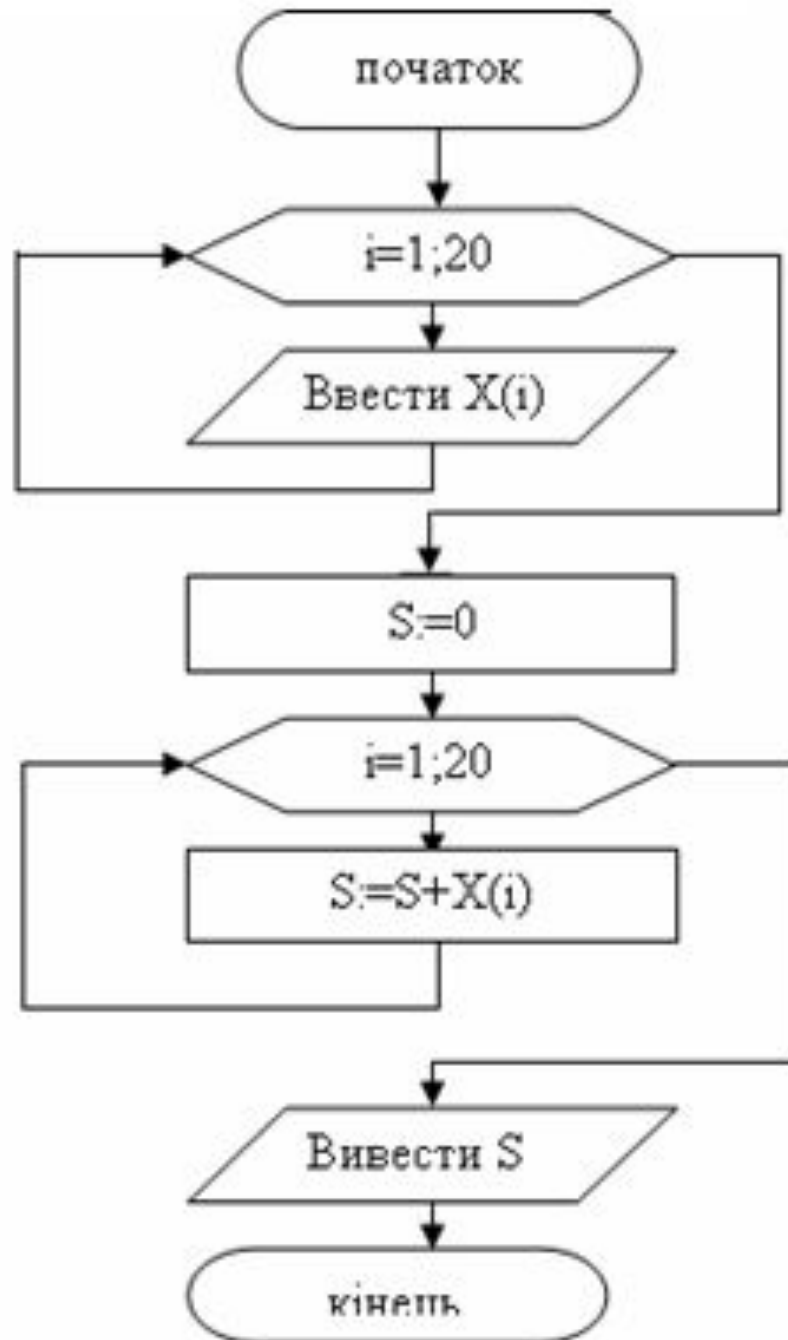


# План

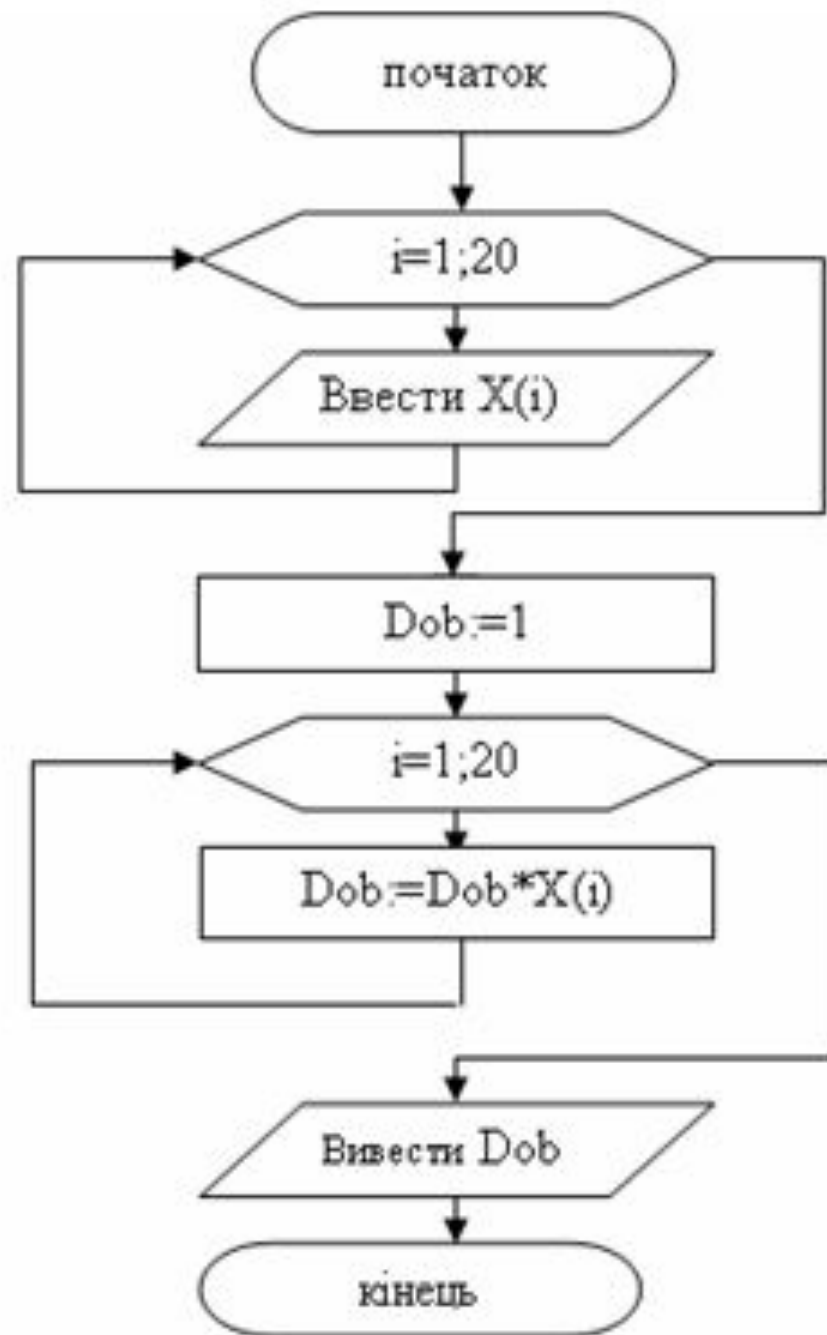
---

- Обчислення суми елементів в ОМ
- Обчислення добутку елементів в ОМ
- Обчислення кількості елементів в ОМ
- Пошук максимального(мінімального) елемента ОМ
- Формування за певною умовою нового масиву на основі заданого

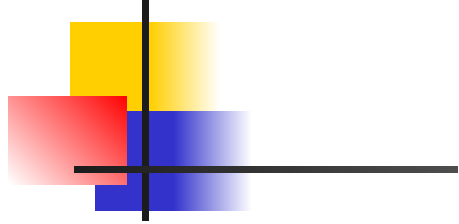
# Сума



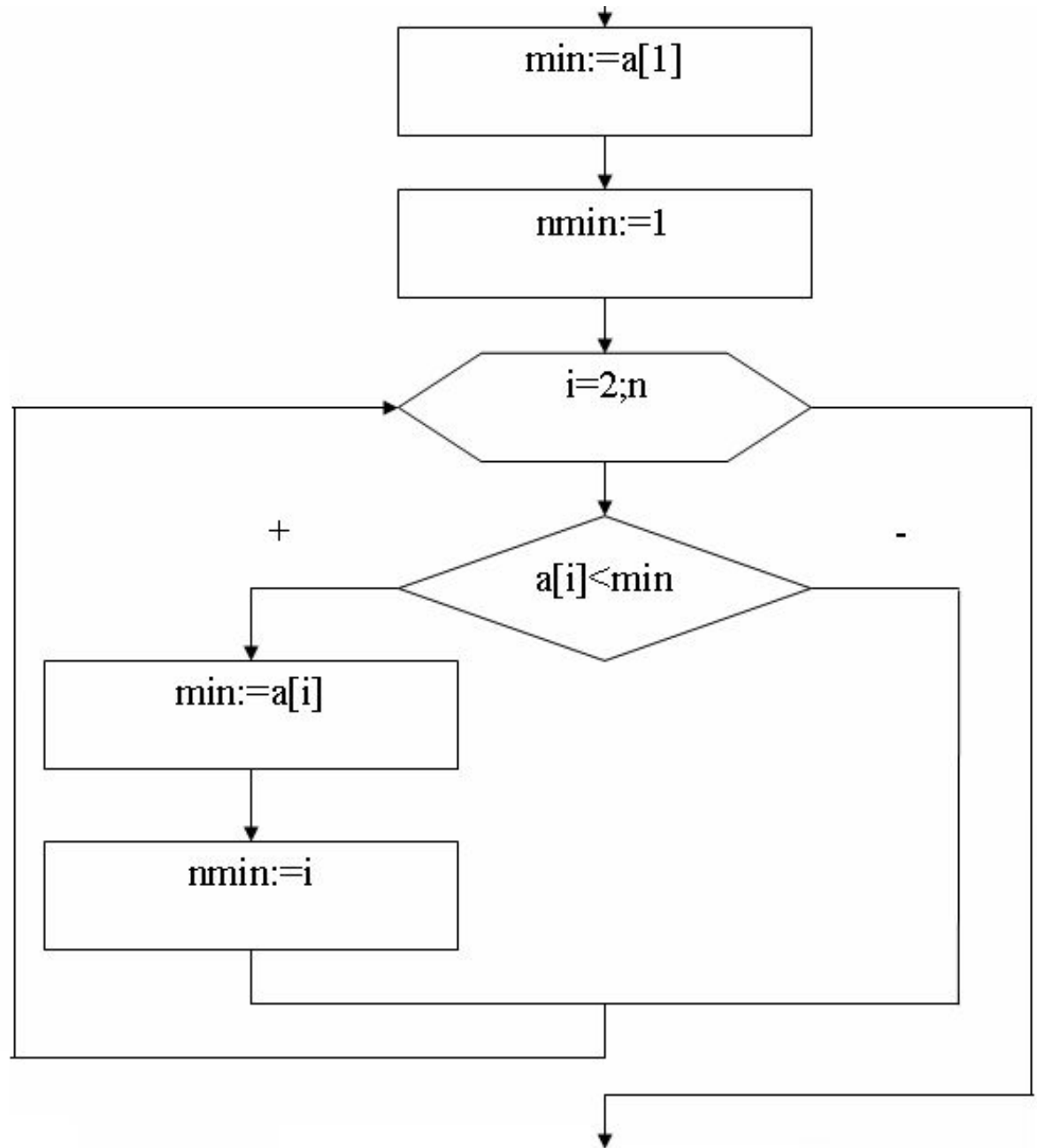
# Добуток







# Пошук мінімального



# Пошук мінімального елемента масиву

Задано масив  $A\{1..30\}$ .

Визначити мінімальний елемент масиву.

A(1)	A(2)	A(3)	...	A(30)
21,5	20	24	...	20,5

③

min   
nmin

①

min   
nmin

②

$A(2) < \text{min}$   
 $20 < 21,5$  (так!)

④

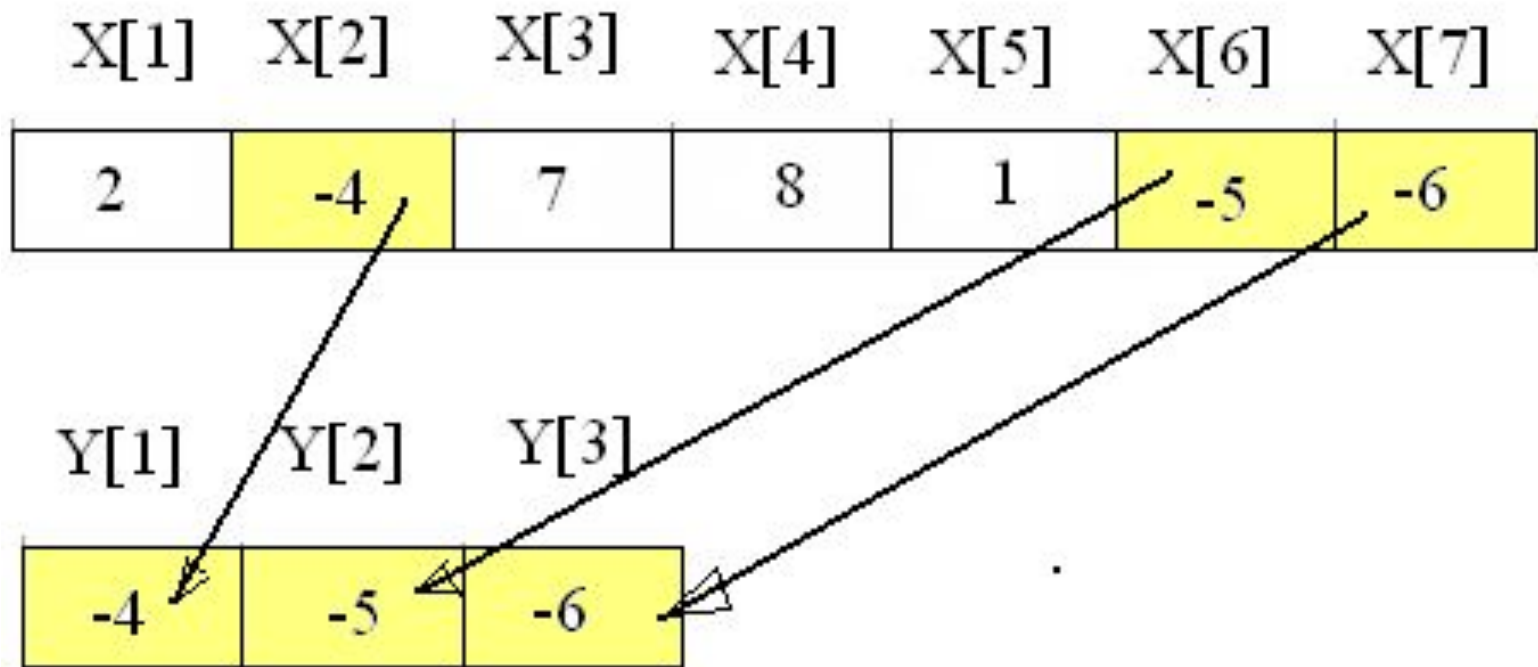
$A(3) < \text{min}$   
 $24 < 20$  (ні!)

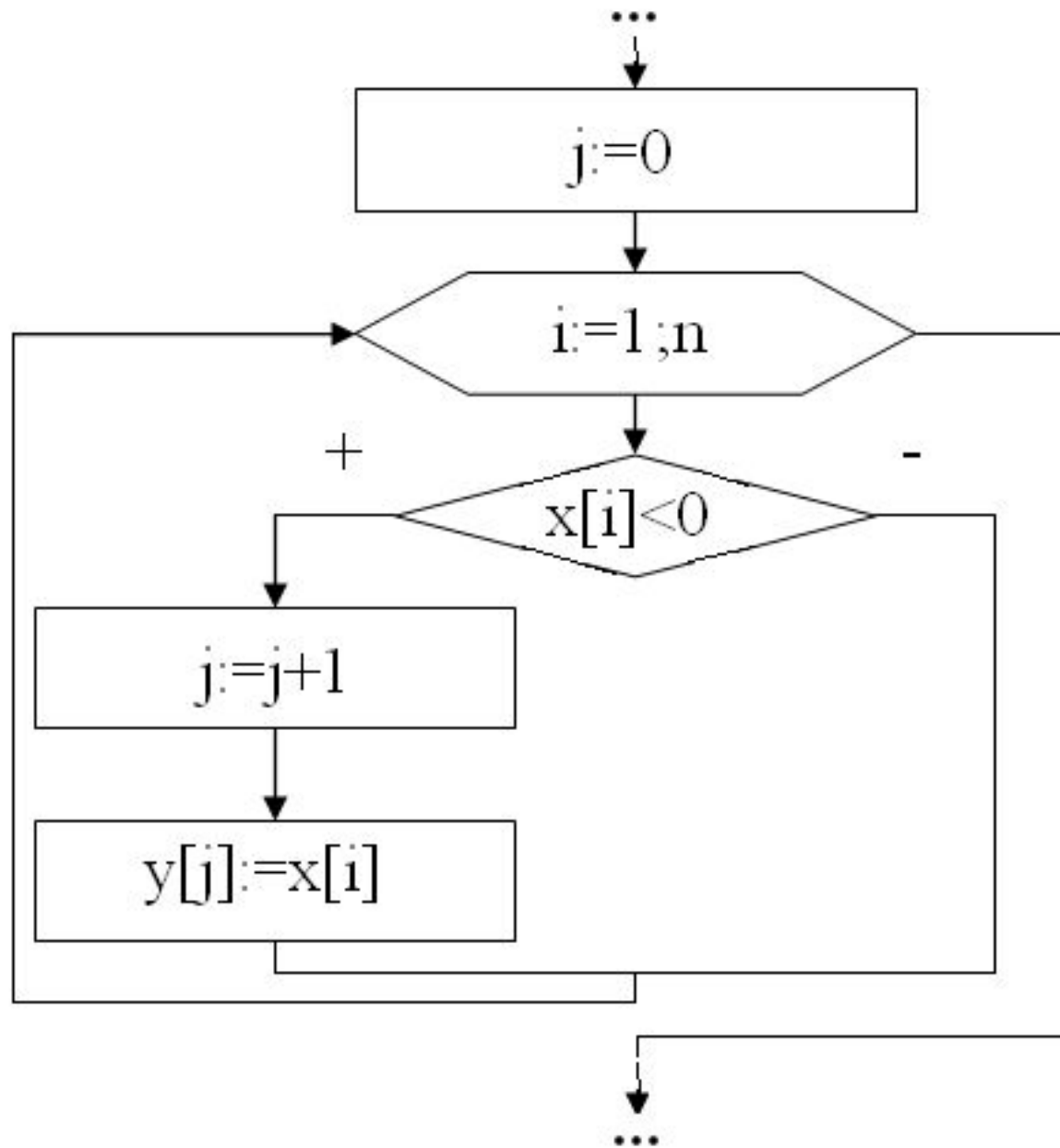
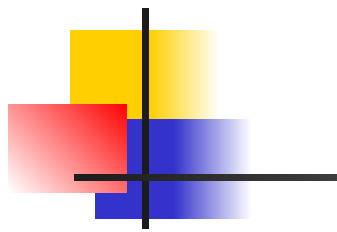
## Формування за певною умовою нового масиву на основі заданого

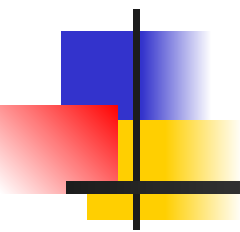
- ...Сформувати масив  $Y$  із непарних елементів масиву  $Z$ ...
- ...Сформувати масив  $Y$  із елементів масиву  $Z$ , які задовольняють умову  $z_i \geq 10$ ...
- ...Сформувати масив  $Z$  із елементів масиву  $Y$ , які розташовані на парних місцях...



# Сутність процесу формування НОВОГО масиву







# Впорядкування одновимірних масивів даних



# План

---

- Впорядкування одновимірних масивів методом "Бульбашки"
- Впорядкування методом вибору

- Впорядкування масиву — це процес перестановки елементів масиву з метою розміщення елементів масиву у визначеному порядку.

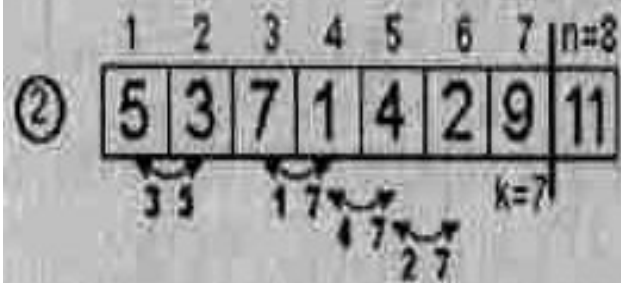
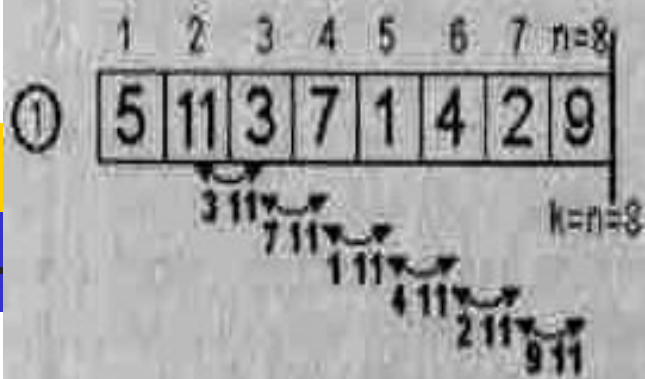
- Заданий одновимірний масив  $A = \{a_1, a_2, a_3, \dots, a_n\}$
- Масив  $A$  є впорядкованим за зростанням елементів, якщо для них виконується ланцюжок нерівностей  **$a_1 < a_2 < a_3 < \dots < a_n$**
- Масив  $A$  є впорядкованим за неспаданням елементів, якщо для них виконується ланцюжок нерівностей  **$a_1 \leq a_2 \leq a_3 \leq \dots \leq a_n$**
- Масив  $A$  є впорядкованим за спаданням елементів, якщо для них виконується ланцюжок нерівностей  **$a_1 > a_2 > a_3 > \dots > a_n$**

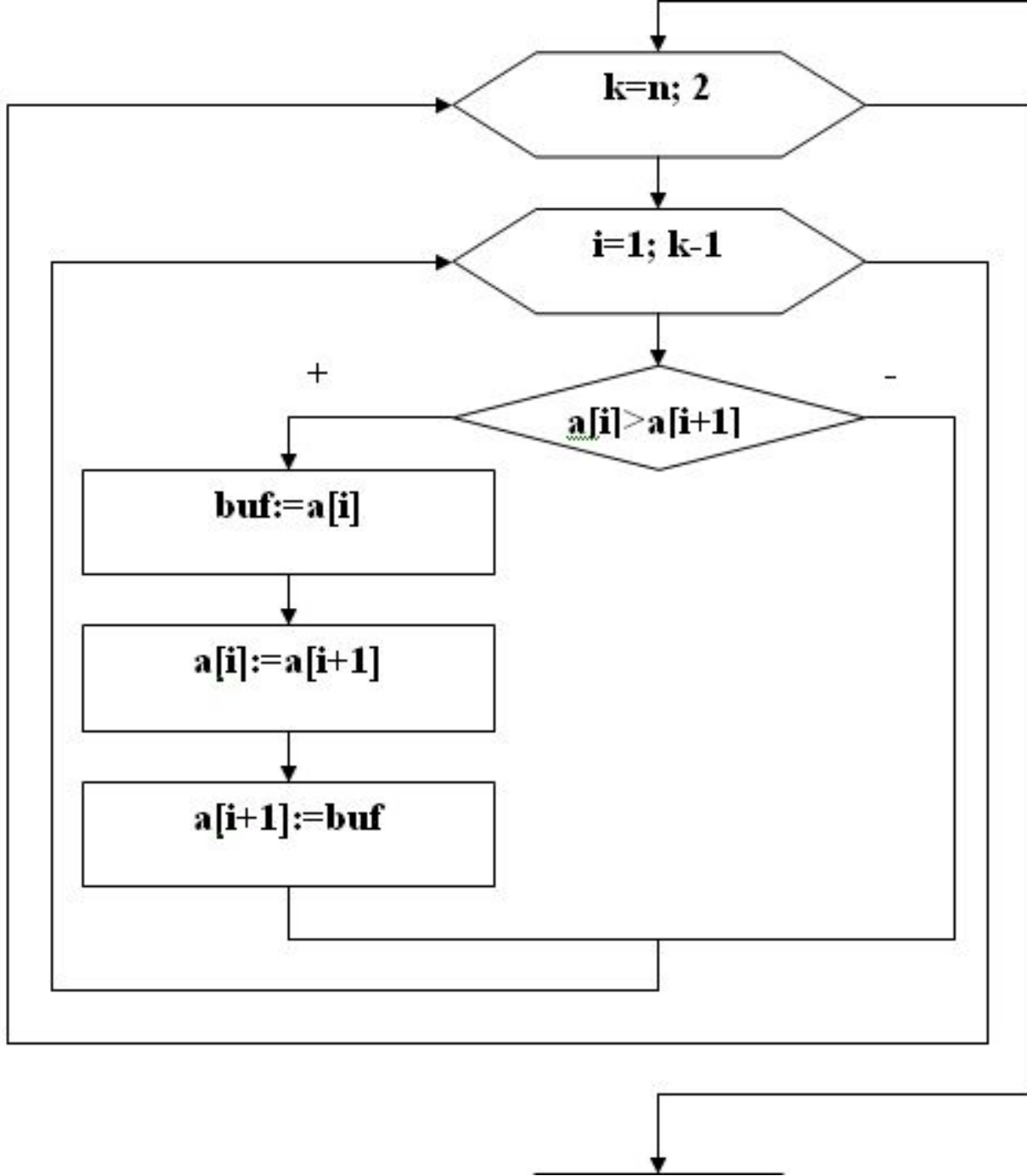


# Метод "Бульбашки"

---

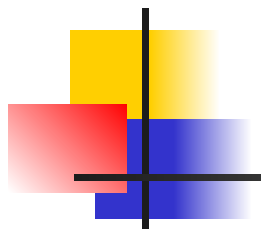
- Зліва направо по черзі порівнюються два сусідні елементи, і якщо їх взаєморозміщення не відповідає заданій умові впорядкування, то вони міняються місцями. Далі беруться два наступні сусідні елементи і так далі до кінця масиву.
- Після одного такого проходу на останній  $n$ -ій позиції масиву стоятиме максимальний елемент («спливла» перша «бульбашка»). Оскільки максимальний елемент вже стоїть на останній позиції, то другий прохід обміну виконуватиметься до  $n-1$  елемента. І так далі. Всього потребується  $n-1$  прохід

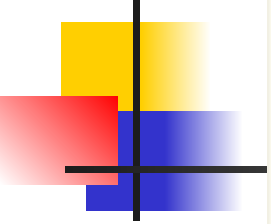






K=8	I=1							A[1]>A[2] (?)
		I=2						A[2]>A[3] (?)
			I=3					A[3]>A[4] (?)
				I=4				A[4]>A[5] (?)
					I=5			A[5]>A[6] (?)
						I=6		A[6]>A[7] (?)
							I=7	A[7]>A[8] (?)
K=7	I=1							A[1]>A[2] (?)
		I=2						A[2]>A[3] (?)
			I=3					A[3]>A[4] (?)
				I=4				A[4]>A[5] (?)
					I=5			A[5]>A[6] (?)
						I=6		A[6]>A[7] (?)
K=6	I=1							A[1]>A[2] (?)
		I=2						A[2]>A[3] (?)
			I=3					A[3]>A[4] (?)
				I=4				A[4]>A[5] (?)
					I=5			A[5]>A[6] (?)
K=5	I=1							A[1]>A[2] (?)
		I=2						A[2]>A[3] (?)
			I=3					A[3]>A[4] (?)
				I=4				A[4]>A[5] (?)
K=4	I=1							A[1]>A[2] (?)
		I=2						A[2]>A[3] (?)
			I=3					A[3]>A[4] (?)
K=3	I=1							A[1]>A[2] (?)
		I=2						A[2]>A[3] (?)
K=2	I=1							A[1]>A[2] (?)





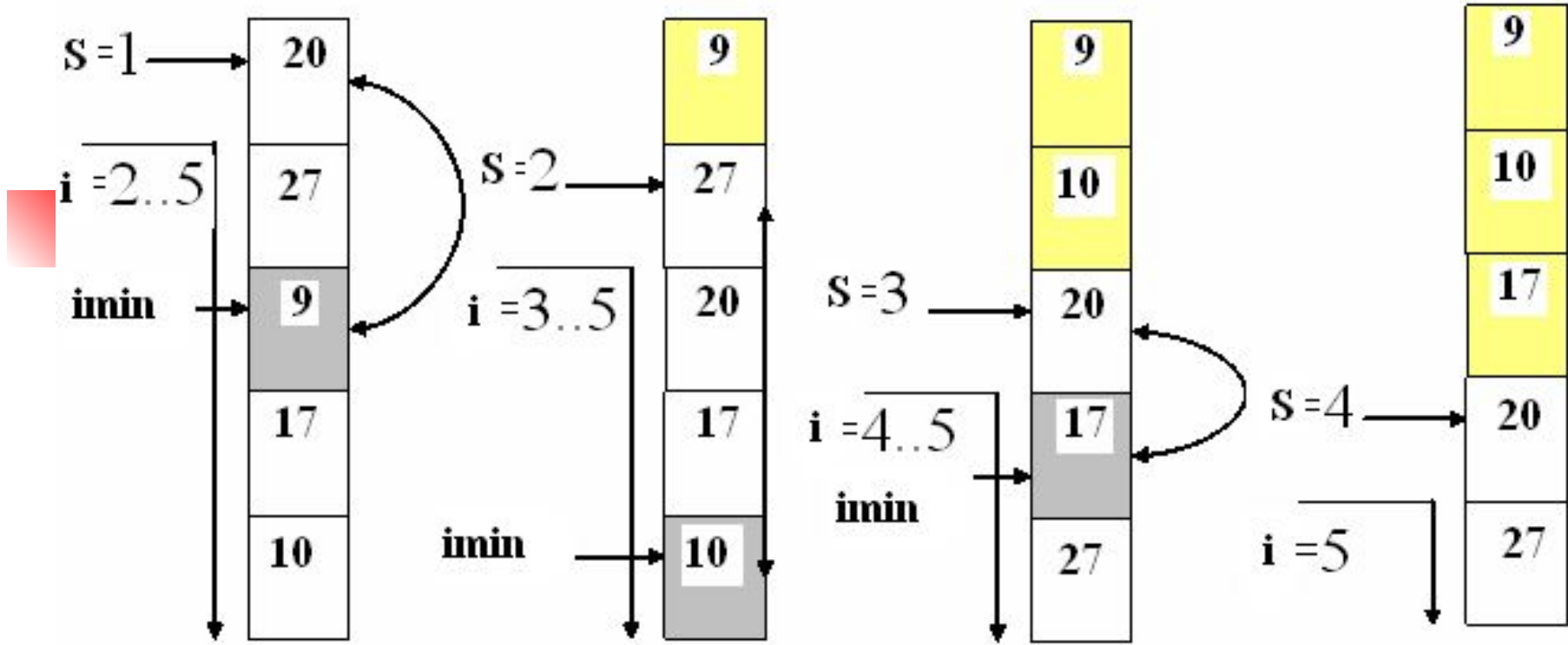
```
program SORT_BULB;
uses wincrt;
const n=10;
var a:array[1..n] of real;
    i,k:integer;
    buf:real;
begin writeln ('Введіть елементи масиву A');
    for i:=1 to n do
        begin write ('A[' ,i, ']=>');
            readln(a[i])
        end;
    for k:=n downto 2 do
        for i:=1 to k-1 do
            if a[i]>a[i+1]
            then begin
                buf:=a[i];
                a[i]:=a[i+1];
                a[i+1]:=buf
            end;
        writeln('масив відсортовано за зростанням');
        for i:=1 to n do
```

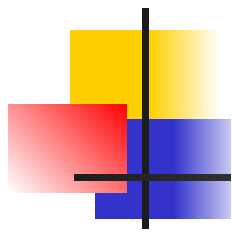
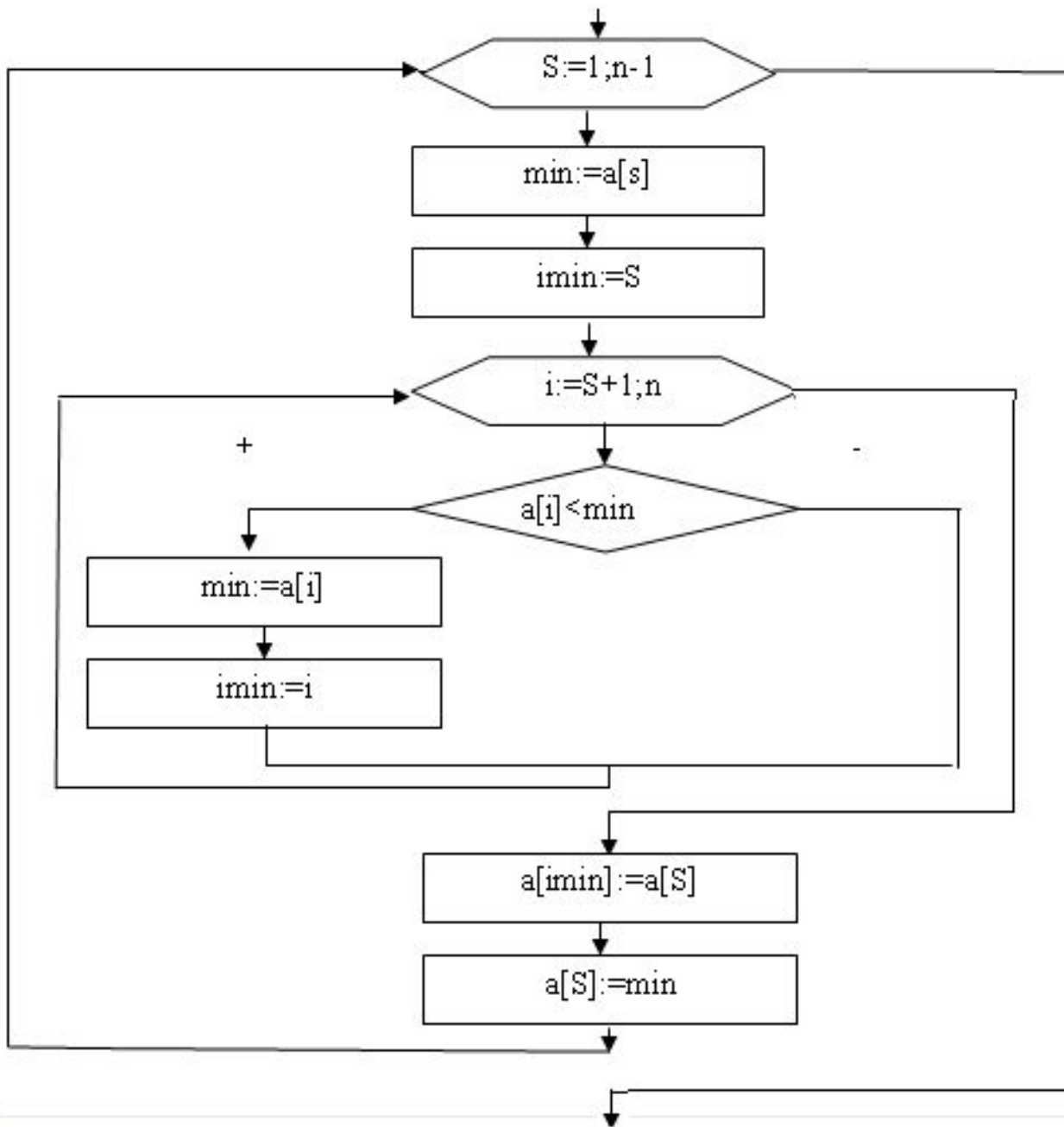


# Метод вибором

---

- Знаходимо (вибираємо) в масиві елемент з мінімальним значенням на інтервалі від 1-го елемента до  $n$ -го (останнього) елемента і міняємо його місцями з першим елементом. На другому кроці знаходимо елемент з мінімальним значенням на інтервалі від 2-го до  $n$ -го елемента і міняємо його місцями з другим елементом. І так далі для всіх елементів до  $(n-1)$ -го.







Turbo Pascal - [noname00.pas]

File Edit Search Run Compile Options Window Help

```
program SORT_vibir;
uses wincrt;
const n=10;
var a:array[1..n] of real;
    i,s,imin:integer;
    min:real;
begin writeln ('Введіть елементи масиву A');
    for i:=1 to n do
        begin write ('A[' ,i, ']=>');
            readln(a[i])
        end;
    for s:=1 to n-1 do
        begin min:=a[s];
            imin:=s;
            for i:=s+1 to n do
                if a[i]<min
                    then begin
                        min:=a[i];
                        imin:=i
                    end;
            a[imin]:=a[s];
            a[s]:=min
        end;
    writeln('масив відсортовано за зростанням');
    for i:=1 to n do
        writeln ('A[' ,i, ']=' ,a[i]:4:2);
end.
```

24:2

Modified

Insert