



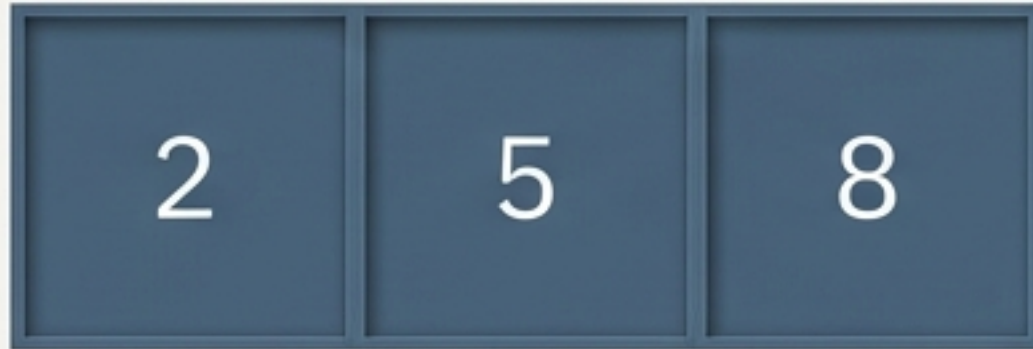
Algoritmul de Interclasare: O Demonstrație Vizuală

Cum combinăm două liste sortate într-una singură, pas cu pas.

Datele de Intrare: Punctul de Plecare

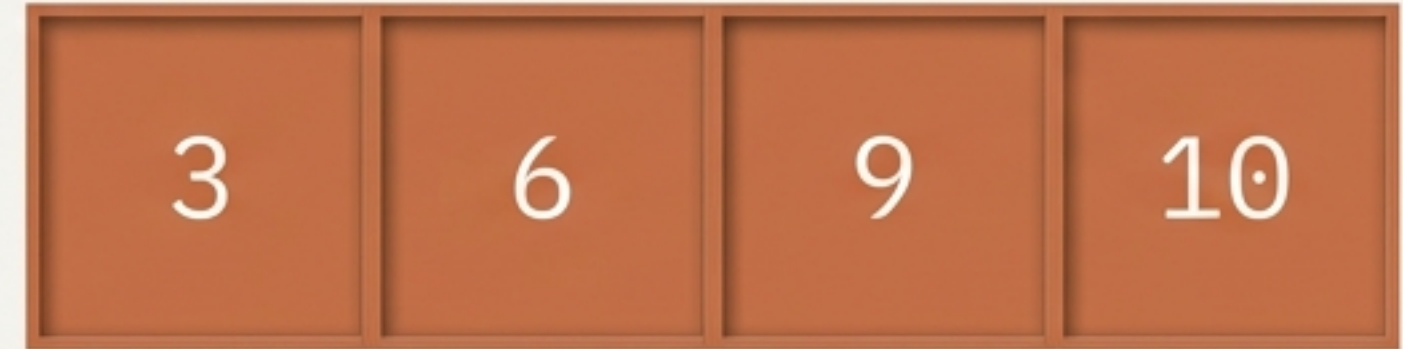
Pornim cu doi vectori, A și B, care sunt deja sortați în ordine crescătoare. Scopul nostru este să îi combinăm într-un al treilea vector, C, care va conține toate elementele în ordine perfectă.

Vectorul A



Dimensiune: $n = 3$

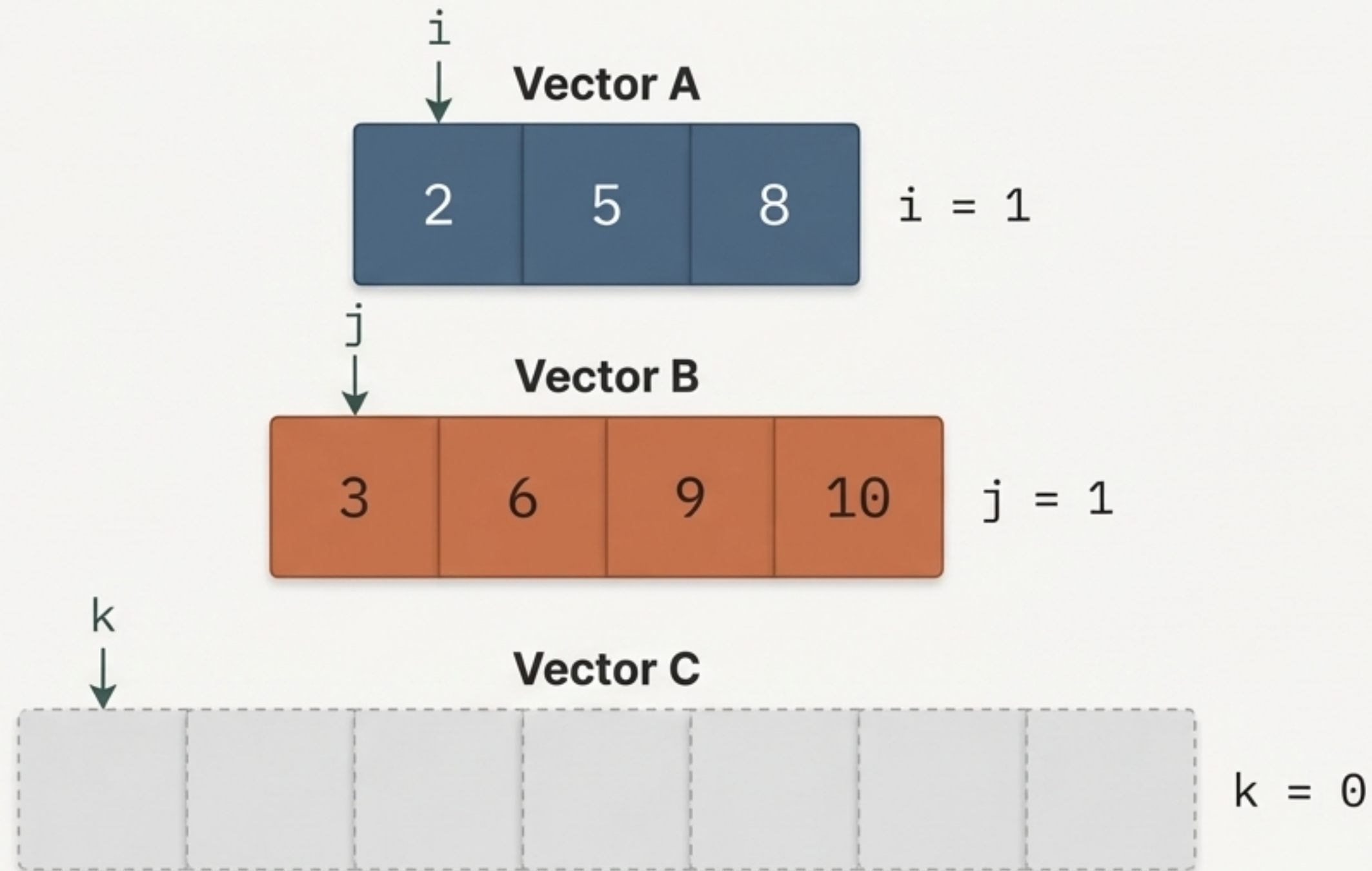
Vectorul B



Dimensiune: $m = 4$

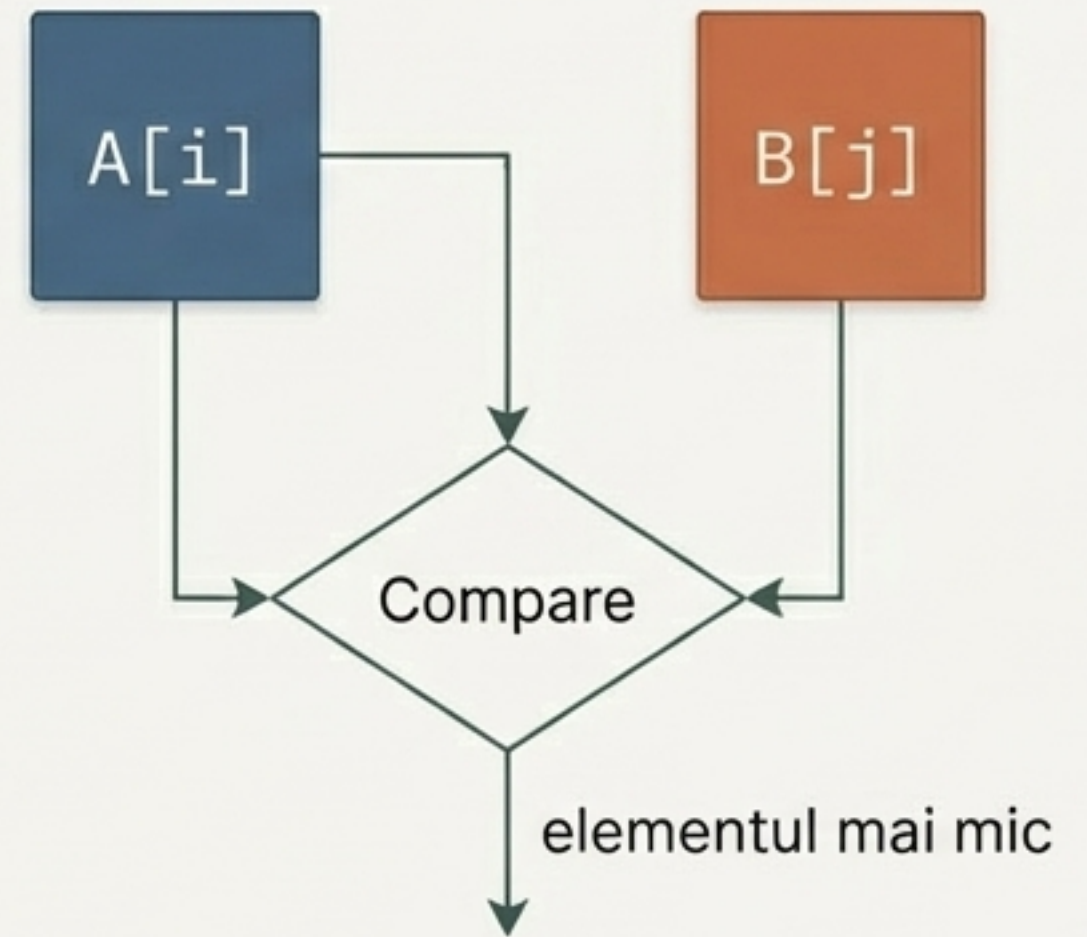
Inițializarea: Pregătirea Instrumentelor

Folosim trei indici (sau pointeri) pentru a naviga prin vectori. `i` parcurge A, `j` parcurge B, iar `k` construiește noul vector C.



Faza 1: Comparația Element cu Element

Logica centrală a algoritmului se bazează pe o singură regulă de comparație, repetată până când unul dintre vectori este epuizat.



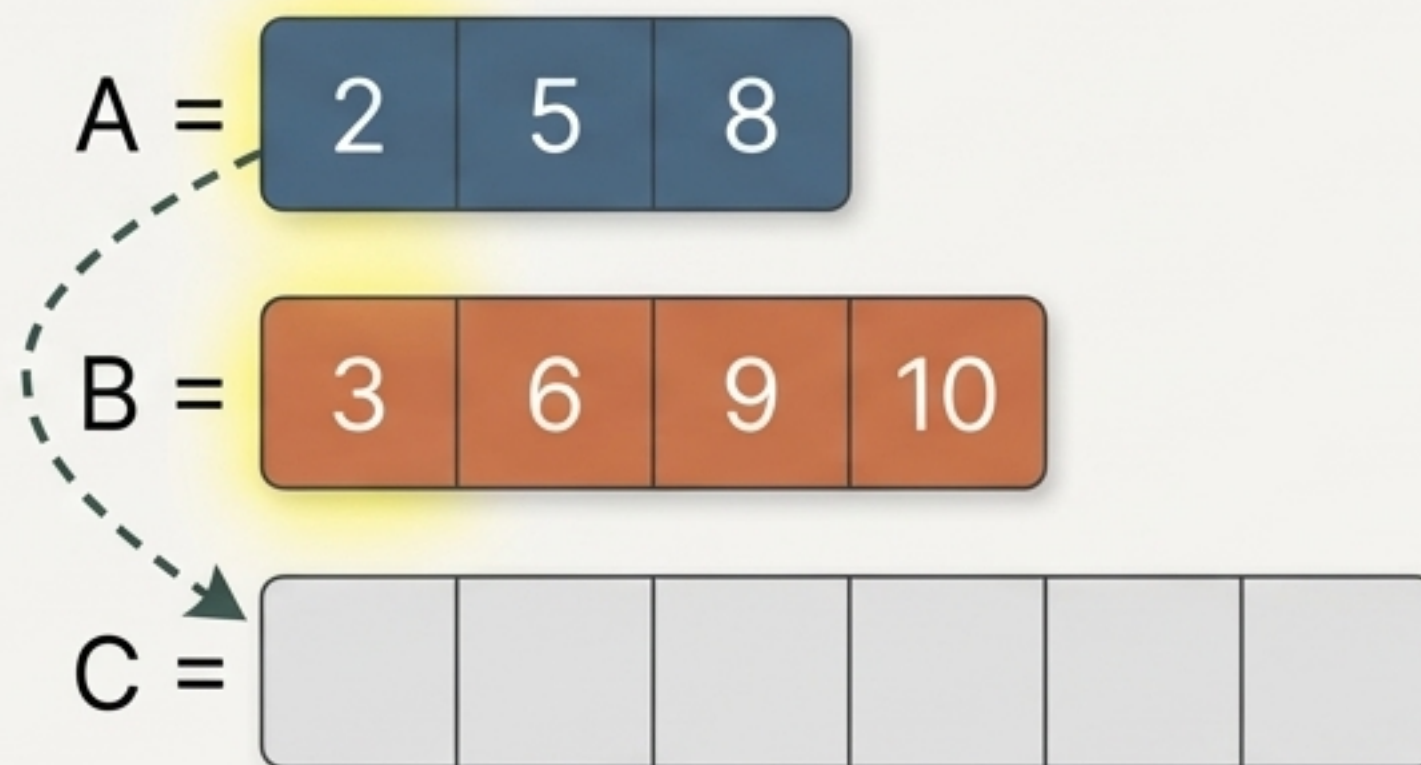
Condiția Buclei Principale

Cât timp $i \leq n$ și $j \leq m$

Pasul 1: Primul Câștigător este cel mai Mic

$i=1, j=1, k=0 \mid A=[2, 5, 8] \mid B=[3, 6, 9, 10]$

1. **Comparare:** $A[1]$ vs $B[1] \rightarrow 2$ vs 3.
2. **Decizie:** $2 < 3$. Elementul mai mic este 2 din vectorul A.
3. **Acțiune:** Se copiază 2 în C.

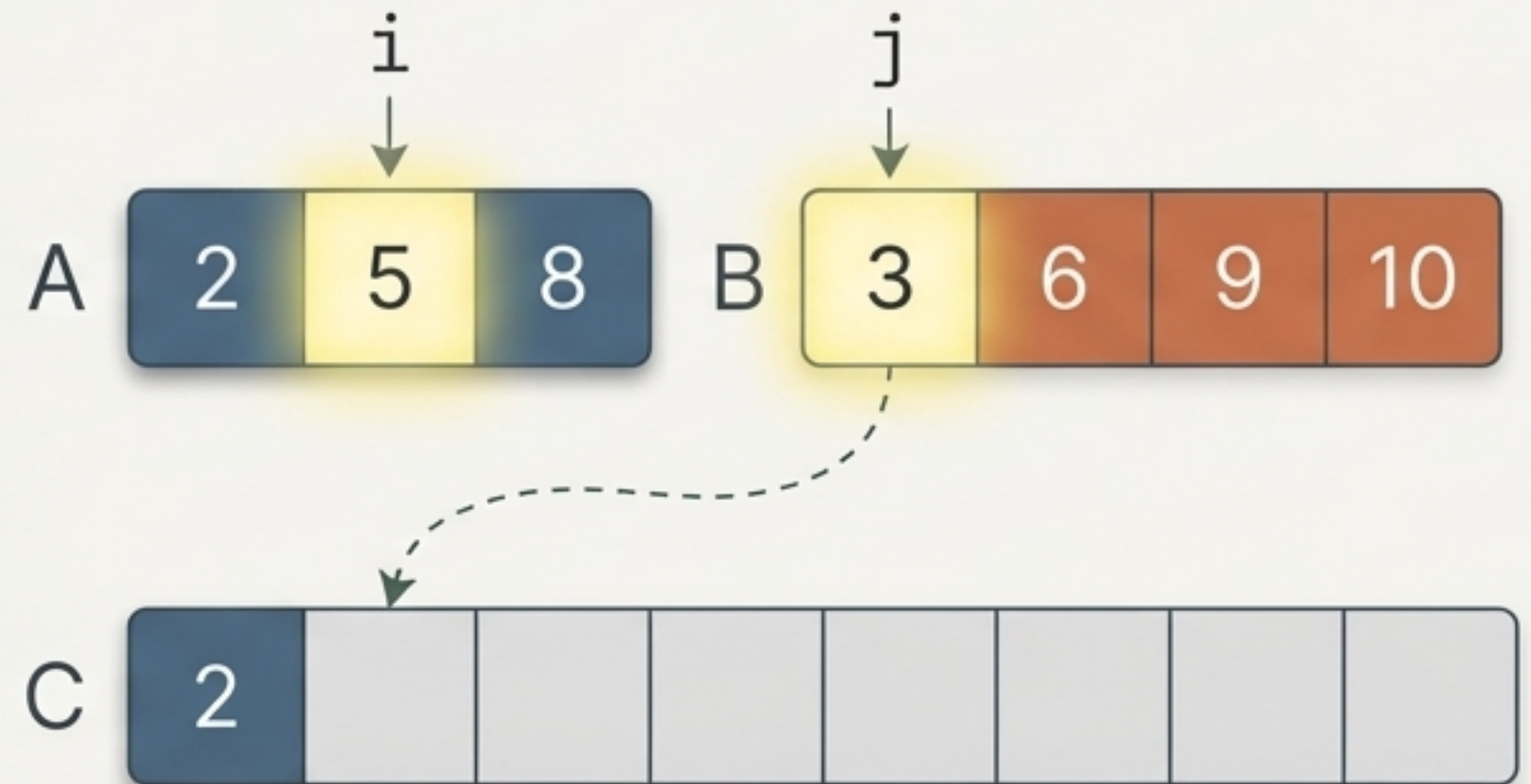


Indicii se actualizează: $i \rightarrow 2, k \rightarrow 1$

Pasul 2: Vectorul B Intră în Joc

`i=2, j=1, k=1 | A=[2, 5, 8] | B=[3, 6, 9, 10] | C=[2, , , , , ,]

1. ****Comparare****: `A[2]` vs `B[1]` → `5` vs `3`.
2. ****Decizie****: `3 < 5`. Elementul mai mic este `3` din vectorul B.
3. ****Acțiune****: Se copiază `3` în `C`.



Indicii se actualizează: `j → 2`, `k → 2`

Pașii 3, 4 și 5: Continuarea Procesului de Comparație

Algoritmul continuă în același mod: compară elementele indicate de `i` și `j`, mută valoarea mai mică în `C` și incrementează indicele corespunzător.

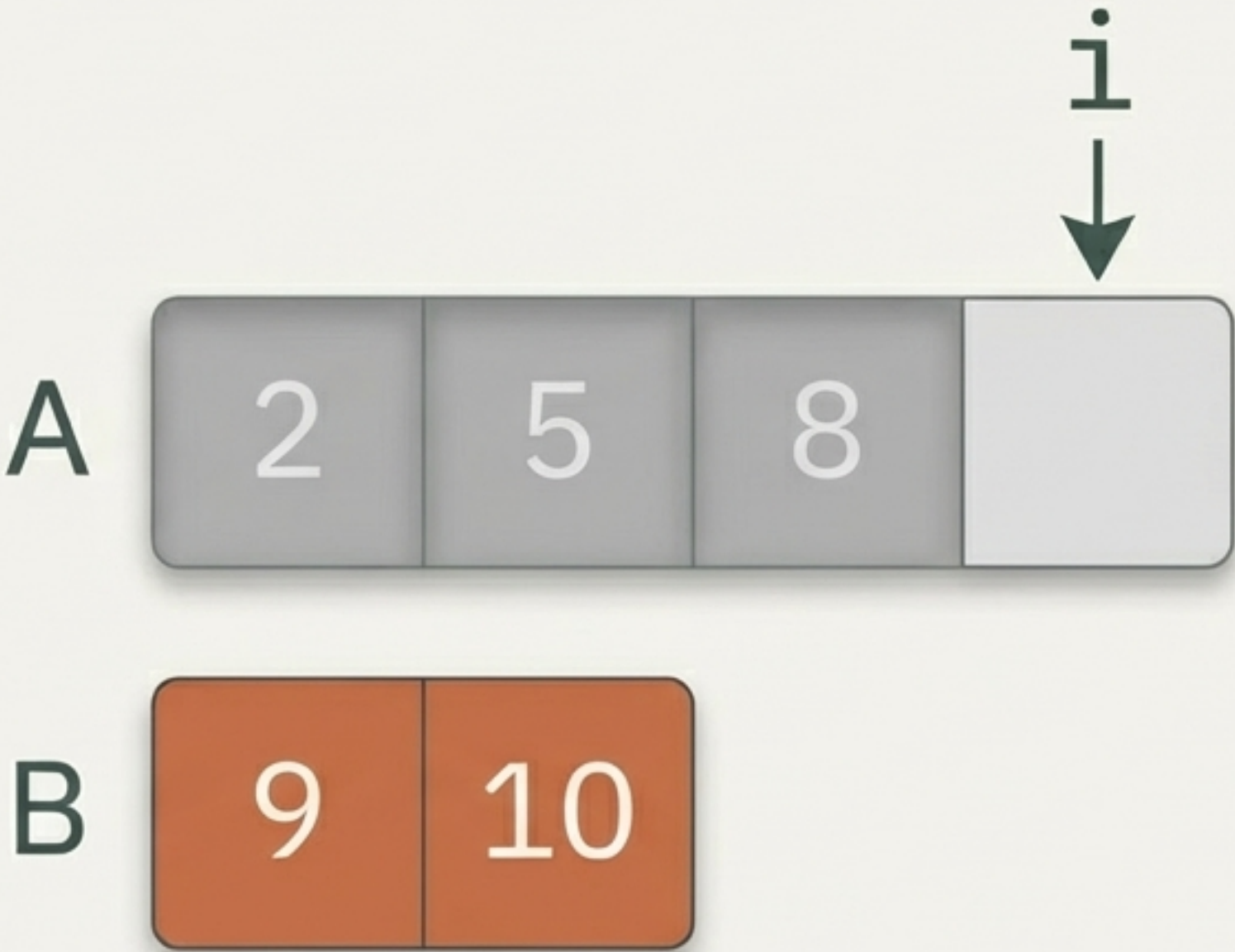
Pas	Stare Inițială (i, j)	Comparare	Decizie	Acțiune	Stare Finală C	Indici Actualizați
3	(2, 2)	A[2]=5 vs B[2]=6	5 < 6	Se ia 5 din A	[2,3,5, , , ,]	i → 3, k → 3
4	(3, 2)	A[3]=8 vs B[2]=6	6 < 8	Se ia 6 din B	[2,3,5,6, , ,]	j → 3, k → 4
5	(3, 3)	A[3]=8 vs B[3]=9	8 < 9	Se ia 8 din A	[2,3,5,6,8, ,]	i → 4, k → 5

Sfârșitul Fazei 1: Vectorul A a Fost Epuizat

Starea Curentă: `i=4, j=3, k=5`

Bucula principală se oprește. De ce?
Deoarece condiția `i <= n` nu mai este îndeplinită.

Condiția:	Cât timp $i \leq n$ și $j \leq m$
Înlocuire:	Cât timp $4 \leq 3$ și $3 \leq 4$
Evaluare:	Cât timp FALS și ADEVĂRAT
Rezultat:	Condiția este FALSĂ .



Faza 2: Copierea Elementelor Rămase

Unul dintre vectori a fost epuizat, dar celălalt mai poate avea elemente. Faza 2 asigură copierea acestor elemente rămase direct în vectorul C, menținând ordinea.



Verificare pentru Vectorul A

Cât timp $i \leq n$

$4 \leq 3 \rightarrow$ **Fals**. Bucla este sărită.

Verificare pentru Vectorul B

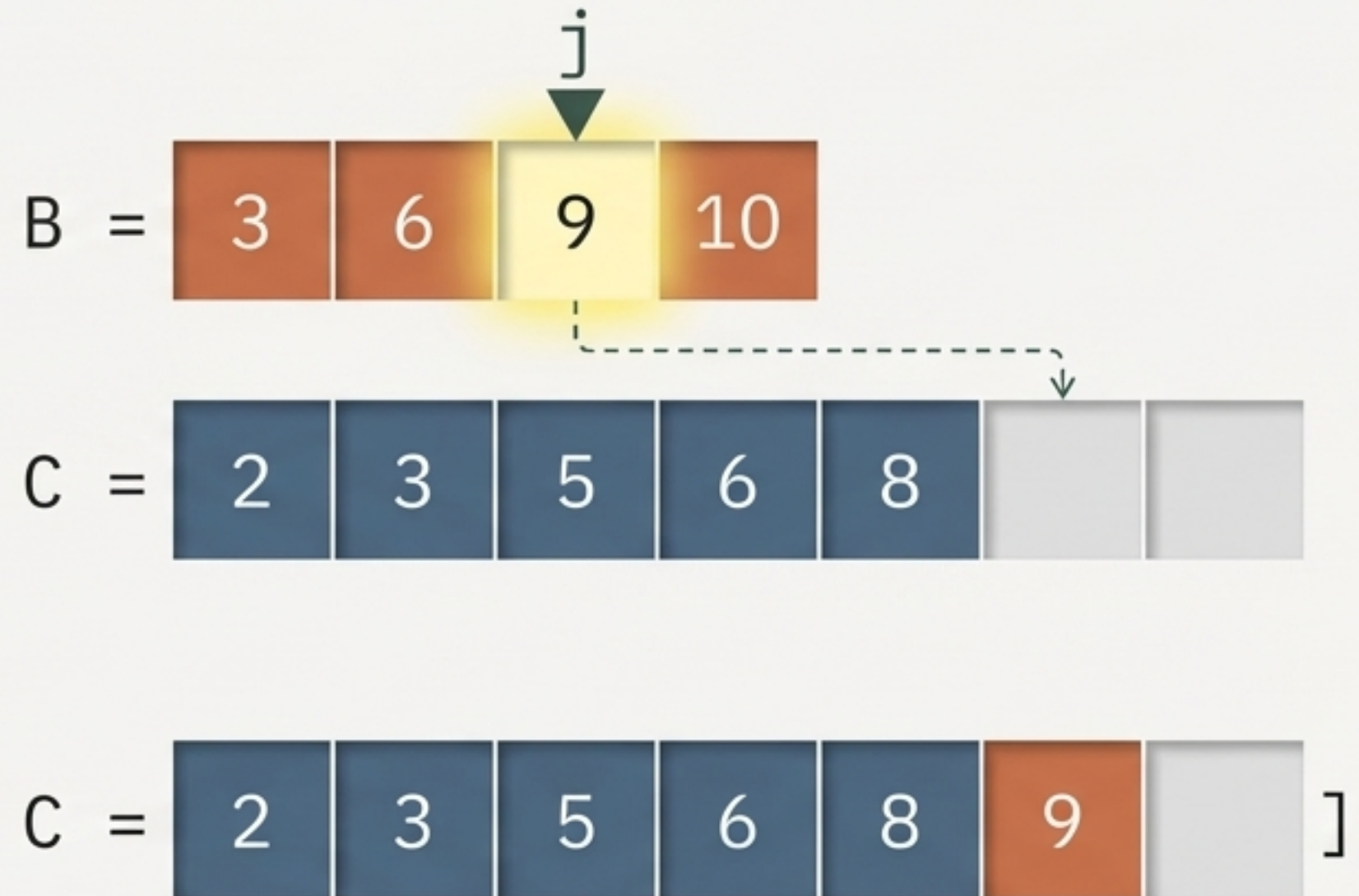
Cât timp $j \leq m$

$3 \leq 4 \rightarrow$ **Adevărat**. Această buclă va rula.

Pasul 6: Copierea Directă a Elementului B[3]

$i=4, j=3, k=5$ | $B=[3, 6, 9, 10]$ | $C=[2, 3, 5, 6, 8, ,]$

1. **Acțiune:** Nu se mai face nicio comparație. Se ia direct elementul $B[3] = 9$.
2. **Copiere:** Se adaugă 9 în C .



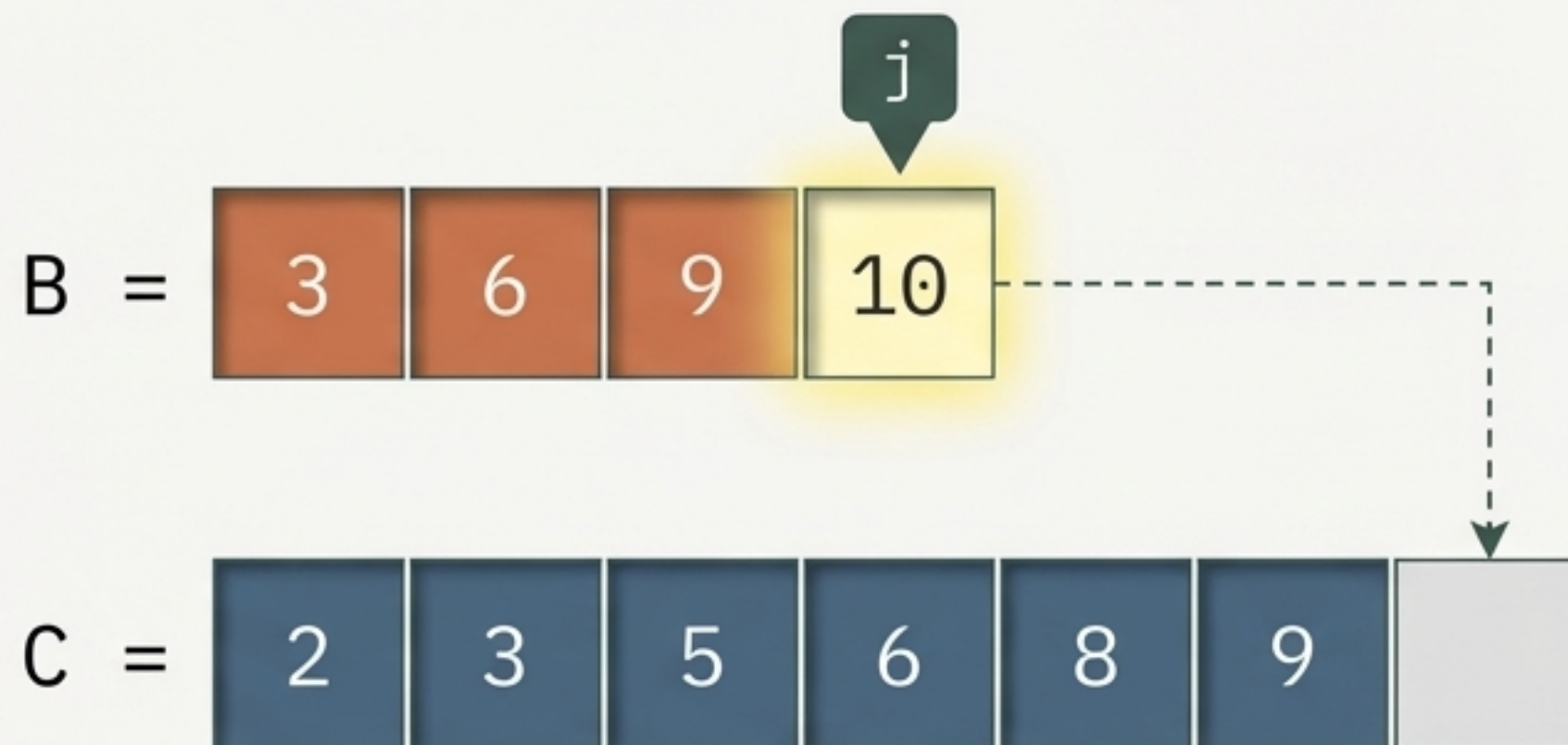
Indicii se actualizează: $j \rightarrow 4, k \rightarrow 6$

Pasul 7: Copierea Ultimului Element, B[4]

$i=4, j=4, k=6$ | $B=[3, 6, 9, 10]$ | $C=[2, 3, 5, 6, 8, 9,]$

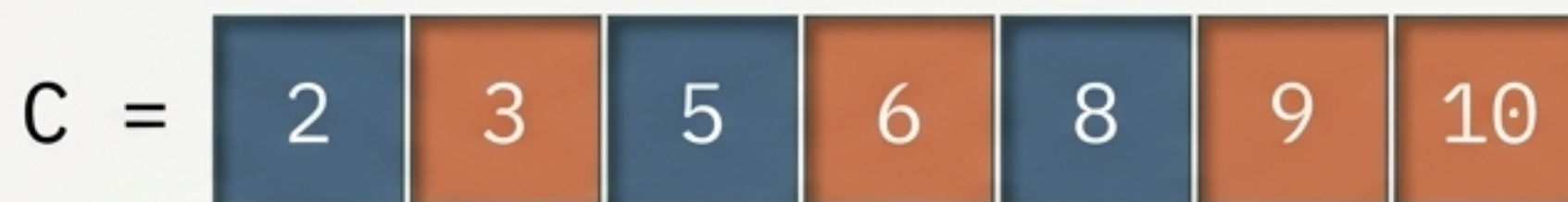
1. Acțiune:

Se ia elementul rămas $B[4] = 10$.



2. Copiere:

Se adaugă 10 în C.

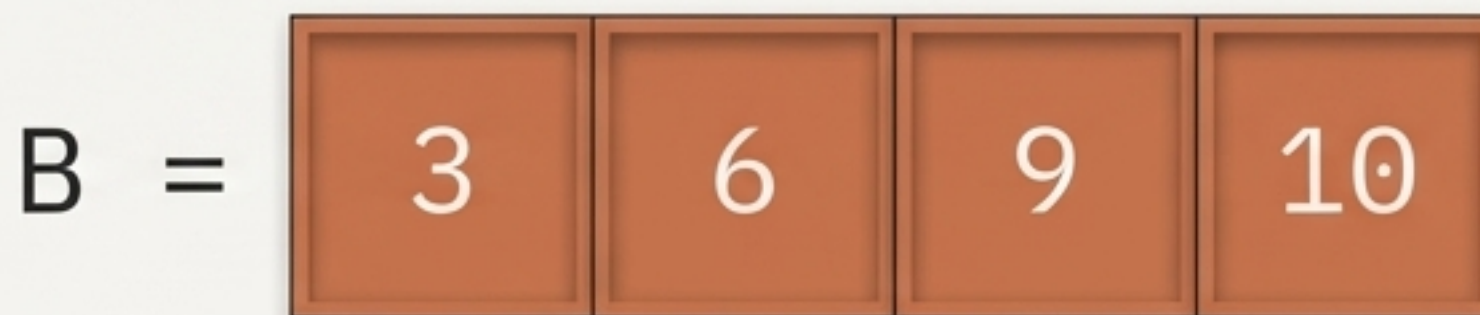
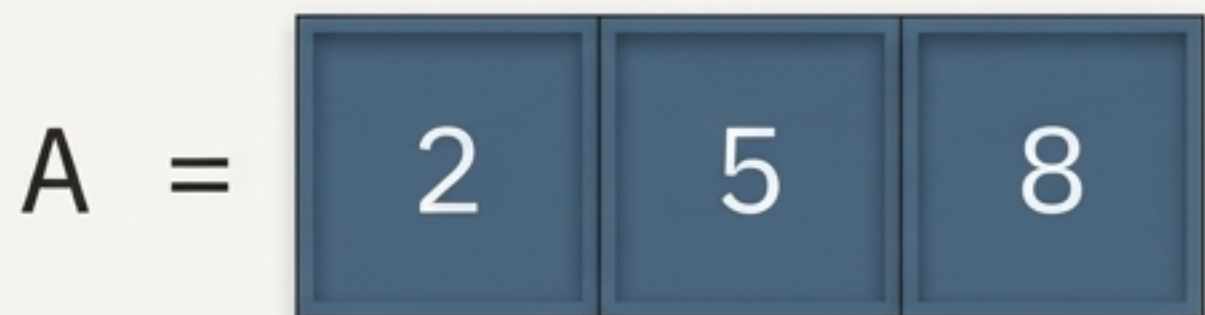


Indicii se actualizează: $j \rightarrow 5, k \rightarrow 7$.

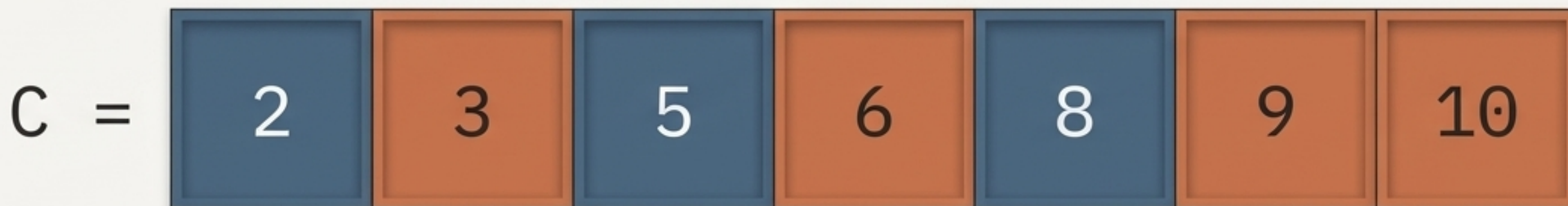
Bucula Cât timp $j \leq m$ devine $5 \leq 4$ (Fals) și se termină.

Rezultat Final: Vectorul C Interclasat

După parcurgerea ambelor faze, vectorul C conține toate elementele din A și B, combinate într-o singură secvență perfect sortată.



Vectorul Final:



Procesul Complet: O Privire de Ansamblu

Pas	i	j	k	Comparare (A[i] vs B[j])	Acțiune (Element Mutat în C)
Start	1	1	0	-	-
1	1	1	0	`A[1]=2` vs `B[1]=3` → `2<3`	Se ia `2` din A. `C[0]=2`
2	2	1	1	`A[2]=5` vs `B[1]=3` → `3<5`	Se ia `3` din B. `C[1]=3`
3	2	2	2	`A[2]=5` vs `B[2]=6` → `5<6`	Se ia `5` din A. `C[2]=5`
4	3	2	3	`A[3]=8` vs `B[2]=6` → `6<8`	Se ia `6` din B. `C[3]=6`
5	3	3	4	`A[3]=8` vs `B[3]=9` → `8<9`	Se ia `8` din A. `C[4]=8`
STOP	4	3	5	Condiție Falsă (`i > n`)	-
6	4	3	5	-	Se ia `9` din B. `C[5]=9`
7	4	4	6	-	Se ia `10` din B. `C[6]=10`
FINAL	4	5	7	-	-