CLASS PROBLEM

🔷 Problem Statement

Humein ek array diya gaya hai, aur task ye hai ki:

- Array ka **sabse chhota element** ko `0` se replace (ya swap) karna hai.
- Usse bada element ko `1` se,
- Next bada element ko `2` se ... aur aise hi array ke **har element ko uske sorted order ke rank se replace karna hai**.

👉 Matlab final array ke elements ki jagah unke **relative rank (0-based)** honge.

Yaani, smallest element = `0`, next smallest = `1`, aur waise hi aage.

Example

Array: `{19, 12, 23, 8, 16}`

Step by step:

- Smallest = `8` → replace with `0`
- Next smallest = `12` → replace with `1`
- Next smallest = `16` → replace with `2`
- Next smallest = `19` → replace with `3`
- Largest = `23` → replace with `4`

👉 Final array banega:

`{3, 1, 4, 0, 2}`

🔷 Ab Code ka Step-by-Step Logic

```
int arr[] = {19,12,23,8,16};
int n = sizeof(arr) / sizeof(arr[0]);
```

➡️ Array define kiya aur uska size `n` nikala.

```
vector<int> visited_or_not(n,0);
int x = 0;
```

➡️ `visited_or_not` ek tracker array banaya jo batayega kaunsa element process ho chuka hai.

➡️ `x` rakhta hai current replacement number (0,1,2,...).

```
for(int i=0;i<n;i++){
    int mn = INT_MAX;
    int mindx = -1;
```

➡️ Har iteration me, abhi tak unprocessed elements me se **minimum element** dhoondhna hai.

➡️ `mn` minimum value store karega, `mindx` uska index.

```
for(int j = 0 ;j<n;j++){
    if(visited_or_not[j] == 1) continue;
    if(mn > arr[j]){
        mn = min(mn,arr[j]);
        mindx = j;
    }
}
```

➡️ Pure array ko traverse karke **unvisited sabse chhoti value** nikal rahe ho.

➡️ `visited_or_not[j] == 1` matlab woh element pehle hi process ho chuka hai → skip karo.

➡️ Agar `mn > arr[j]` to `mn` ko update karo aur `mindx` store karo.

```
arr[mindx] = x;
visited_or_not[mindx] = 1;
```

```
x++;
```

➡️ Jo minimum element mila usko replace kardo current rank `x` se.

➡️ Usko visited mark karo.

➡️ `x++` so that next minimum ko agla rank mile (0,1,2...).

```
for(int i=0;i<n;i++){
   cout<<arr[i]<<" ";
}
```

➡️ Final array print kardo.

🔷 Code ka Output

Tere code ka output hoga:

```
3 1 4 0 2
```

🔷 Short Summary

Tera code basically **Selection Sort ka variant** hai jisme:

- Har step par minimum element choose karke usko rank (`x`) assign kar raha hai.
- Final output ek **rank-mapped array** hota hai jisme chhota element = `0`, usse bada = `1`, etc.

Initial State

```
arr = {19, 12, 23, 8, 16}
visited_or_not = {0, 0, 0, 0, 0}
x = 0
```

Iteration 1 (i = 0)

- Minimum element among unvisited = **8** (index 3).
- Replace with `x = 0`.

```
arr = {19, 12, 23, 0, 16}
visited_or_not = {0, 0, 0, 1, 0}
x = 1
```

Iteration 2 (i = 1)

- Unvisited = {19, 12, 23, 16}
- Minimum = **12** (index 1).
- Replace with `x = 1`.

```
arr = {19, 1, 23, 0, 16}
visited_or_not = {0, 1, 0, 1, 0}
x = 2
```

Iteration 3 (i = 2)

- Unvisited = {19, 23, 16}
- Minimum = **16** (index 4).
- Replace with `x = 2`.

```
arr = {19, 1, 23, 0, 2}
visited_or_not = {0, 1, 0, 1, 1}
x = 3
```

Iteration 4 (i = 3)

- Unvisited = {19, 23}

- Minimum = **19** (index 0).
- Replace with `x = 3`.

```
arr = {3, 1, 23, 0, 2}
visited_or_not = {1, 1, 0, 1, 1}
x = 4
```

Iteration 5 (i = 4)

- Unvisited = {23}
- Minimum = **23** (index 2).
- Replace with `x = 4`.

```
arr = {3, 1, 4, 0, 2}
visited_or_not = {1, 1, 1, 1, 1}
x = 5
```

Final Output

```
3 1 4 0 2
```

⚡ Iska matlab:

- Original `8` ban gaya `0`
- `12` ban gaya `1`
- `16` ban gaya `2`
- `19` ban gaya `3`
- `23` ban gaya `4`
