String function → “stoi “ uses

🔍 Goal of the Code:

👉 From an array of strings representing numbers (some with leading zeroes), find the **largest number** 🏆 — and also show the original string form! 🔢💥

📦 Step-by-Step Breakdown:

🧠 1. The Setup

```
string arr[]={"0123","0023","456","00182","940","2901"};
```

📦 An array of strings — looks like numbers but with **leading zeroes** 🚦

💡 Example: `"0123"` is the same as `123`

🚀 2. Convert First Element

```
int max=stoi(arr[0]);
string maxS=arr[0];
```

🎯 `stoi()` converts a string into an **integer** 🧱

📌 We set the first number as our starting **maximum**!

🔁 3. Loop to Find the Max

```
for(int i=1;i<=5;i++){
    int x=stoi(arr[i]);
    if(x>max){
        max=x;
        maxS=arr[i];
```

```
    }
  }
```

🔁 We go through each element (from index 1 to 5):

1️⃣ Convert string to number using `stoi()`

2️⃣ Compare with current `max`

3️⃣ If it's bigger, update both:

- `max` ➕ the number
- `maxS` ➕ the original string

🧠 Smart! You **preserve the string version**, not just the number.

📣 4. Final Output

```
cout<<maxS<<endl;
cout<<max<<endl;
```

🖨️ This will print:

1. 🧵 The original string (with leading zeroes if any)
2. 🔢 The actual max number (without leading zeroes)

🧪 Example Output

Given:

```
{"0123","0023","456","00182","940","2901"}
```

✅ Output:

```
2901
2901
```

⚠️ Small Fix Alert!

```
for(int i=1;i<=5;i++)
```

⚠️ Works here because array has exactly 6 elements

✅ Better & safer:

```
for(int i=1;i<6;i++)
```

or ideally:

```
int n = sizeof(arr)/sizeof(arr[0]);
for(int i=1; i<n; i++)
```

✅ Final Takeaway

This code teaches you how to:

- 🔁 Loop through string arrays
- 🔄 Convert strings to numbers (`stoi`)
- 🧠 Track max values
- 🧵 Keep both number and string versions
