Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

(Week 11) Lecture 21 and 22

Objectives: **Learning objectives of this lecture are**

- **Dynamic Array**
- **Searching**
- **Sorting**

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

Lecture # 21 Dynamic Array and Linear Search

In this lecture we shall be discussing about dynamic array, searching and sorting techniques. In previous lecture we discussed static array. Now what if we think of declaring a static array with user defined size (input from user). In other words size of an array should not be provided at compile time rather to give it run time by taking input from user. Figure given below illustrates that what happens when we declare an array by taking size of array as input.

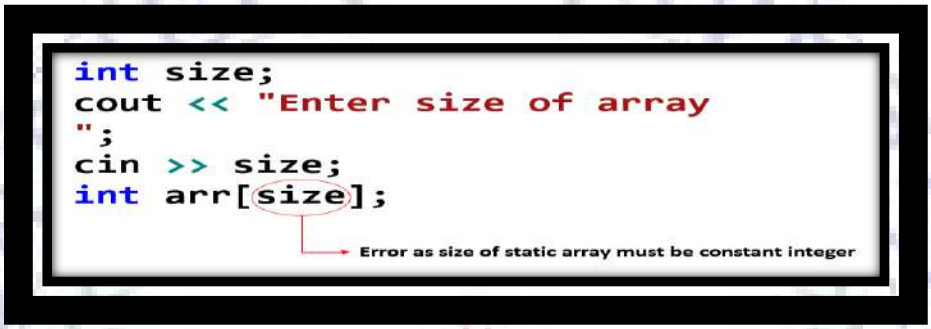


Figure 1 Static Array (Error)

Dynamic Array:

When we think of dynamic array a question arises that Why dynamic array? Is static array not enough? Answer is so simple look at figure 1 it is clear that static array fails when we think of making an array by providing size of array at run time (taking input or some other source). To overcome this failure dynamic array is used. How to declare a dynamic array? Figure given below provide answer to this question.

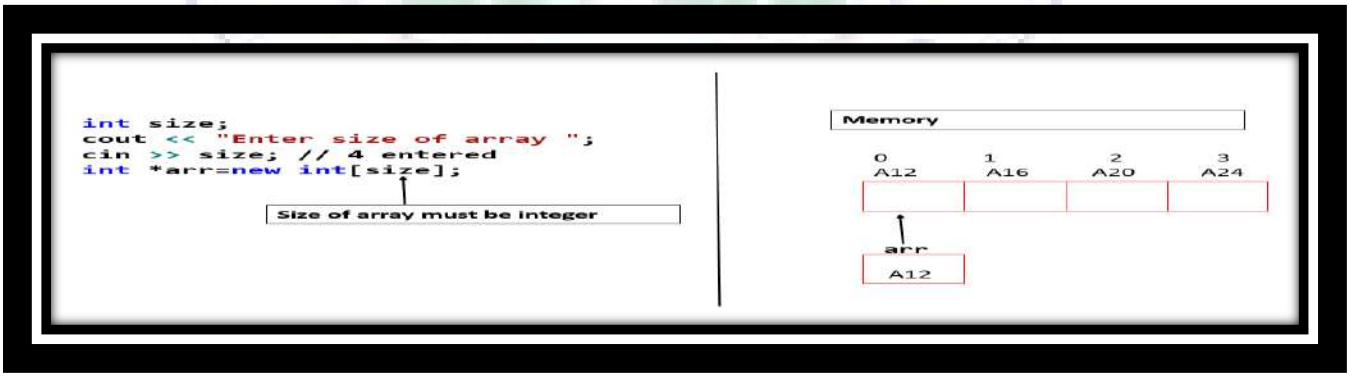


Figure 2 Dynamic Array

Muhammad Athar email id: athar@northern.edu.pk, Whatsapp# 0333-5077664

Keyword ***new*** is being used for declaring dynamic array. With the help of new key word consecutive memories can be declared as in above figure it is shown that ***new int[size]*** declares a 4 memories assuming value of size is 4. After declaring four memories address of first memory is returned and is stored in pointer. One thing to note in C# and JAVA array is created using new key word so we will be declaring our arrays using new keyword whether size is known or not.

Program Number 1: Input number of storing appeared in programming fundamentals exam. Declare an array of type int for storing marking size of array is number of student appeared in exam. After declaring an array take input.

Solution:

```
#include "iostream"
using namespace std;
void main()
{
	int total_students;
	cout << "Enter total number of students ";
	cin >> total_students;

	int* marks = new int[total_students];
	for (int i = 0; i < total_students; i++)
	{
		cout << "Enter marks of student number " << i + 1 << " ";
		cin >> marks[i];
	}
}
```

Searching: There are two types of searching techniques Linear and Binary search in this lecture we shall be discussing linear search.

Linear Search: Starting a search in a sequence following example will make it more clear about linear search.

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

Each box Contains programming related book and we have to search C++ book from given boxes

Open Box 0 and look whether it has C++ book inside it

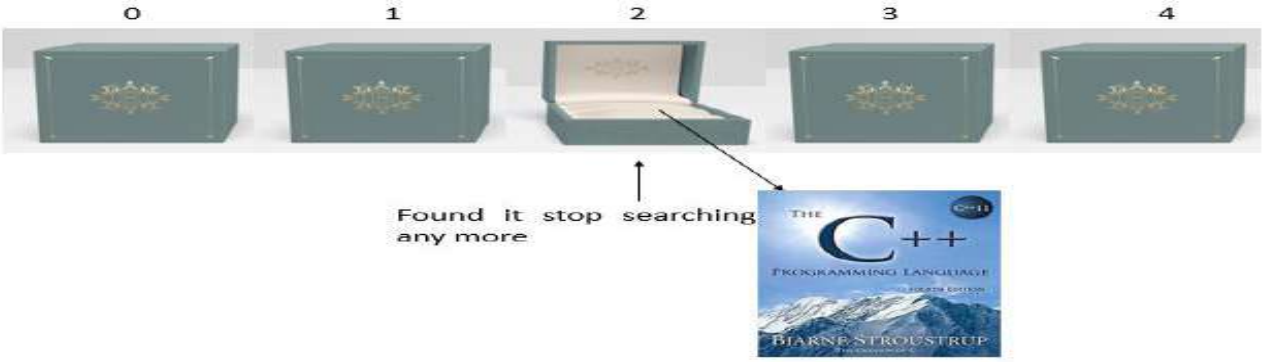
It contains NODE.JS Look in next box

It contains PYTHON Look in next box

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**



Linear search is also known as sequential search. As it moves in a sequence to search an item let us write a code for searching a number from an integer array of size 5

Program Number 2: Declare an integer array of size 5. Input a number to search if number found display message number found otherwise not found.

Solution:

```
#include "iostream"
using namespace std;
void main()
{
int* number = new int[5];
for (int i = 0; i<5; i++)
{
cout << "Enter number " << i + 1 << endl;
cin >> number[i];
}
cout << "Enter number to search ";
int searchnum;
cin >> searchnum;
```

```
if (number[0] == searchnum)
cout << "Found " << endl;
else if (number[1] == searchnum)
cout << "Found " << endl;
else if (number[2] == searchnum)
cout << "Found " << endl;
else if (number[3] == searchnum)
cout << "Found " << endl;
else if (number[4] == searchnum)
cout << "Found " << endl;

else
cout << "Not found " << endl;

}
```

```
for (int i = 0; i < 5; i++)
{
if (number[i] == searchnum)
cout << "Found ";

else
cout << "Not Found ";
}
```

```
bool found = false;
for (int i = 0; i < 5; i++)
{
if (number[i] == searchnum)
{
found = true;
}
}
if (found)
cout << "Found " << endl;
else
cout << "Not Found " << endl;
```

Not correct because code give at left side

```
else
cout << "Not Found ";
```

Is executed after checking all indexes. For this we will use flag (a variable to keep track of found/Not found)

Modified Code:

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

```
#include "iostream"
using namespace std;
void main()
{
	int* number = new int[5];
	for (int i = 0; i<5; i++)
	{
		cout << "Enter number " << i + 1 << endl;
		cin >> number[i];
	}
	cout << "Enter number to search ";
	int searchnum;
	cin >> searchnum;


	bool found = false;
	for (int i = 0; i < 5; i++)
	{
		if (number[i] == searchnum)
		{
			found = true;
			break;
		}
	}
	if (found)
		cout << "Found " << endl;
	else
		cout << "Not Found " << endl;


}
```

Let us dry run above code by considering following values in a loop.

0	1	2	3	4
25	5	3	6	2

Value input for searchnum ➔ 6

Value of found ➔ false. Note that found variable will have one of two values true/false. We initialized it as false and when a number is found its value is changed to true.

int i=0;

Loop Iterations

Conditional Part	Body	Increment

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

i < 5 ➔ 0 < 5 ➔ True	if (number[i] == searchnum) means
	if (number[0] == searchnum) means
	if (25 ==6) ➔ false will not execute body of if and will move to increment portion
		i++➔ i➔1

i < 5 ➔ 1 < 5 ➔ True	if (number[i] == searchnum) means
	if (number[1] == searchnum) means
	if (5 ==6) ➔ false will not execute body of if and will move to increment portion
		i++➔ i➔2

i < 5 ➔ 2 < 5 ➔ True	if (number[i] == searchnum) means
	if (number[2] == searchnum) means
	if (3 ==6) ➔ false will not execute body of if and will move to increment portion
		i++➔ i➔3

i < 5 ➔ 3 < 5 ➔ True	if (number[i] == searchnum) means
	if (number[3] == searchnum) means
	if (6 ==6) ➔ true execute body of if
	found=true; ➔ will change value of found to true.
	break; statement will break scope of loop and will be out of loop.
		i++➔ i➔1

Out if loop.

```
if (found) ➔ found =true means if (true) ➔ will execute body if.
              cout << "Found " << endl;
       else
              cout << "Not Found " << endl;
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

Note that value of found true means number is found at some index as it changed its value from true to false.

Lecture # 20 Sorting

In this lecture we shall be discussing sorting techniques. Sorting means arranging data in some order either ascending or descending. There are several sorting techniques few of them are mentioned below.

- ✓ Selection Sort
- ✓ Bubble Sort
- ✓ Insertion Sort
- ✓ Count Sort
- ✓ Quick Sort

You may be thinking why there is a need of so many sorting techniques when one sort is available. These techniques differ in term of complexity (time/space) that will be discussed in future courses. Now let us discuss selection sort in detail.

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

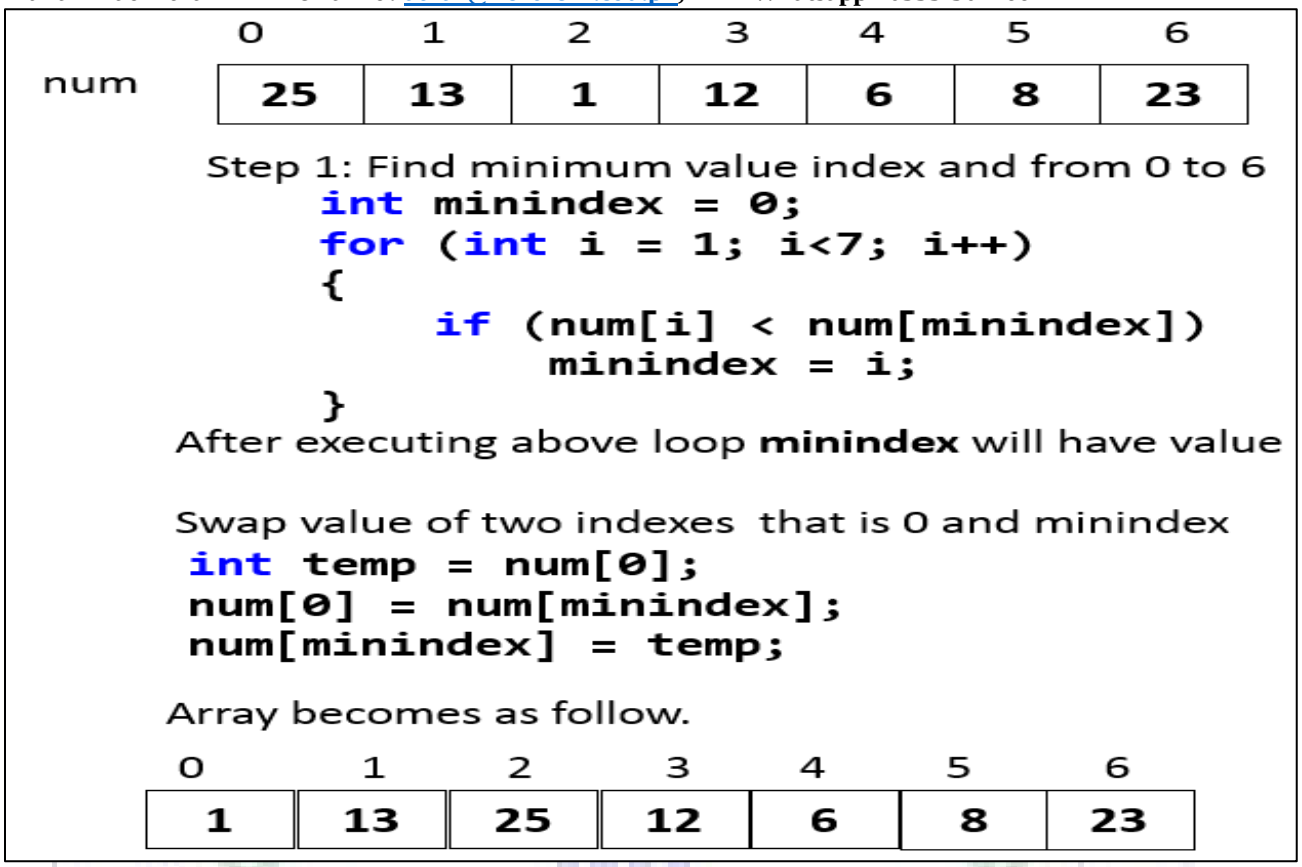


Figure 1 Step 1 Selection Sort

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

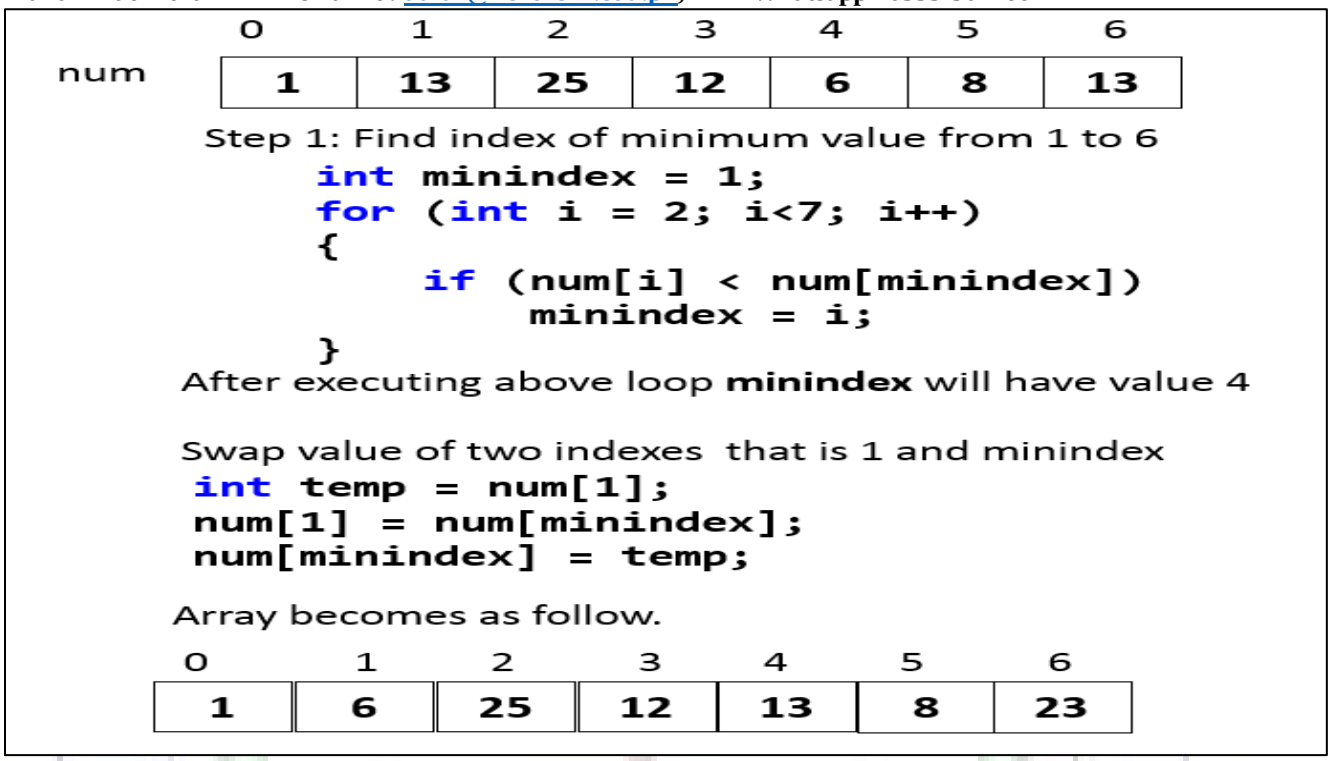
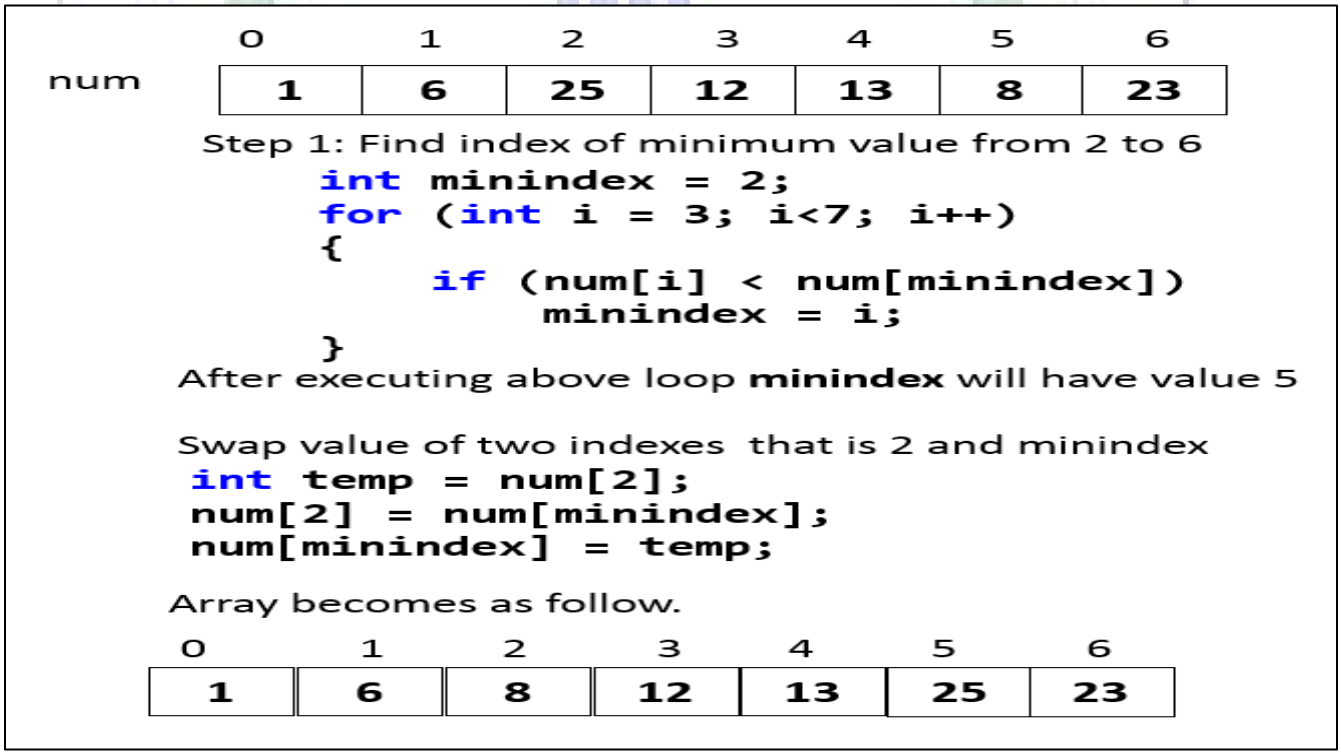


Figure 2 Step 2 Selection Sort



Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

Figure 3 Step 3 Selection Sort

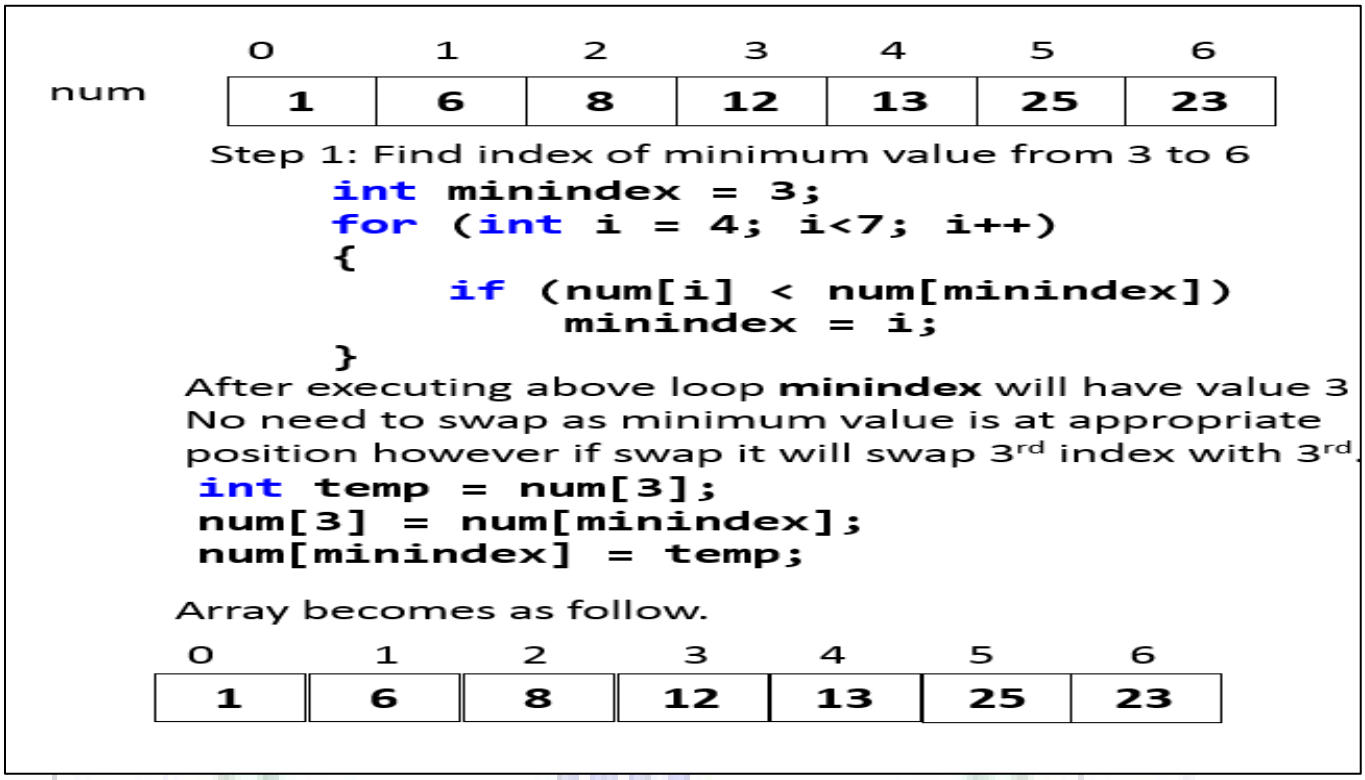
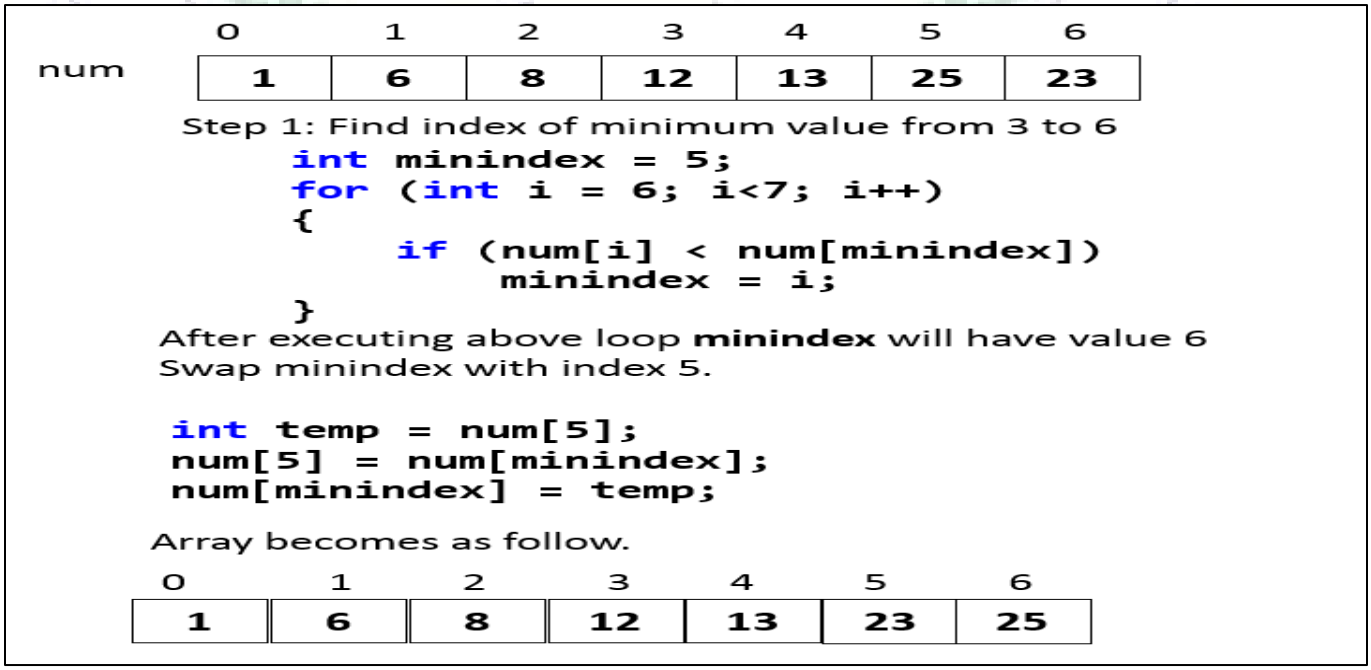


Figure 4 Step 4 Selection Sort



Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

Figure 5 Step 5 Selection Sort

```
#include "iostream"
using namespace std;
void main()
{
	int* num = new int[7];
	for (int i = 0; i < 7; i++)
	{
		cin >> num[i];
	}
	int minindex = 0,temp;
	for (int i = 1; i<6; i++)
	{
		if (num[i] < num[minindex])
			minindex = i;
	}
	temp = num[0];
	num[0] = num[minindex];
	num[minindex] = temp;
	minindex = 1;
	for (int i = 2; i < 7; i++)
	{
		if (num[i] < num[minindex])
			minindex = i;
	}
	temp = num[1];
	num[1] = num[minindex];
	num[minindex] = temp;
	minindex = 2;
	for (int i = 3; i < 7; i++)
	{
		if (num[i] < num[minindex])
			minindex = i;
	}
	temp = num[2];
	num[2] = num[minindex];
	num[minindex] = temp;
	minindex = 2;
	for (int i = 3; i < 7; i++)
	{
		if (num[i] < num[minindex])
			minindex = i;
	}
	temp = num[2];
	num[2] = num[minindex];
	num[minindex] = temp;
	minindex = 3;
	for (int i = 4; i < 7; i++)
	{
		if (num[i] < num[minindex])
			minindex = i;
	}
```

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**


```
	temp = num[3];
	num[3] = num[minindex];
	num[minindex] = temp;
	minindex = 4;
	for (int i = 5; i < 7; i++)
	{
		if (num[i] < num[minindex])
			minindex = i;
	}
	temp = num[4];
	num[4] = num[minindex];
	num[minindex] = temp;
	minindex = 5;
	for (int i = 6; i < 7; i++)
	{
		if (num[i] < num[minindex])
			minindex = i;
	}
	temp = num[5];
	num[5] = num[minindex];
	num[minindex] = temp;


}
```

Using Nested Loop

```
#include "iostream"
using namespace std;
void main()
{
	int* num = new int[7];
	int minindex , temp;
	for (int i = 0; i < 7; i++)
	{
		cin >> num[i];
	}
// selection sort
	for (int j = 0; j < 7; j++)
	{
		minindex = j;

		for (int i = j+1; i < 6; i++)
		{
			if (num[i] < num[minindex])
				minindex = i;
		}
		temp = num[j];
		num[j] = num[minindex];
		num[minindex] = temp;
		minindex = 1;
	}


}
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

Count Sort:

Count sort consist of following steps

1. Find maximum value in array
2. Declare second array of size maximum +1 and initialize array with 0.
3. Now value of array 1 is considered as index of array 2 and add 1 in the value.
4. Now iterate through array 2 and keep on placing values in array 1 note that value of array 1 will be index of array 2.
5. On placing each value from array 2 in array 1 decrease array 2 until reduced to 0.
6. Count sort does not works for negative numbers.

0	1	2	3	4
2	3	5	3	1

Maximum Value of is 5 declare array of 5+1= 6 and initialize 0

0	1	2	3	4	5
0	0	0	0	0	0

Look at 0 index of array 1 its 2 add 1 in index 2 of second array. Array 2 becomes

0	1	2	3	4	5
0	0	1	0	0	0

Look at 1 index of array 1 its 3 add 1 in index 3 of second array. Array 2 becomes

0	1	2	3	4	5
0	0	1	1	0	0

Look at 2 index of array 1 its 5 add 1 in index 5 of second array. Array 2 becomes

0	1	2	3	4	5

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar **email id: athar@northern.edu.pk,** **Whatsapp# 0333-5077664**

0	0	1	1	0	1

Look at 3 index of array 1 its 3 add 1 in index 3 of second array. Array 2 becomes

0	1	2	3	4	5
0	0	1	2	0	1

Look at 4 index of array 1 its 1 add 1 in index 1 of second array. Array 2 becomes

0	1	2	3	4	5
0	1	1	2	0	1

From array 2 try to place index in array 1 and decrease value until it is reduces to 0

Value at Array 2 0 index is 0 it means it is never appeared in array.

Value at Array 2 1 index is 1 it means 1 is appeared one time so place 1 in array 1 in at index 0.

Value at Array 2 2 index is 1 it means 2 is appeared one time so place 2 in array 1 at index 1.

Value at Array 2 3 index is 2 it means 3 is appeared two times. Place 3 two times in array 1 at index 2 and 3.

Value at Array 2 4 index is 0 it means 4 is never appeared in array.

Value at Array 2 5 index is 1 it means 5 is appeared one times. Place 5 one times in array 1 at index 4.

Code

```
#include "iostream"
using namespace std;
void main()
{
	int arr[5];
	for (int i = 0; i < 5; i++)
	{
		cin >> arr[i];
	}
	int max = arr[0];
```

Northern University, Nowshera

Programming Fundamentals (ECS-121)

Muhammad Athar email id: athar@northern.edu.pk, Whatsapp# 0333-5077664

```
	for (int i = 1; i < 5; i++)
	{
		if (max < arr[i])
			max = arr[i];
	}
	int* arr2 = new int[max + 1];
	for (int i = 0; i < max + 1; i++)
	{
		arr2[i] = 0;
	}

	for (int i = 0; i < 5; i++)
	{
		arr2[arr[i]] += 1;
	}
	int index = 0;
	for (int i = 0; i < max + 1; i++)
	{
		if (arr2[i] != 0)
		{
			for (int j = 1; j <= arr2[i]; j++)
			{
				arr2[index] = i;
				index++;
			}
		}
	}
}
```