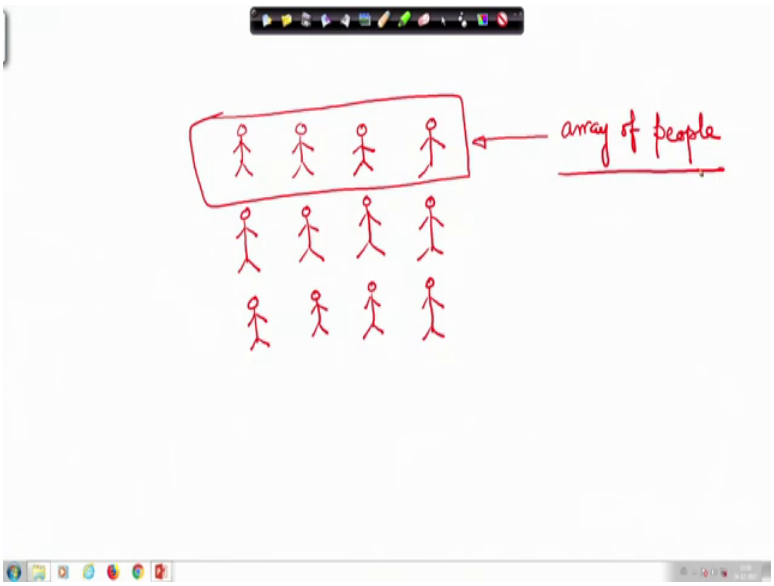
Problem Solving through Programming In C
Prof. Anupam Basu
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 26
Introduction to Arrays

Till now we have looked into, we have seen different programming constructs using which we can implement repetition; that means, a set of statements will be repeated in a loop. And in C such contracts where while do while and for loop, right? Now also we have seen how we can branch out. Coming through a sequential one after another execution of the instructions depending on the condition we can go out to one path or another path through if else structures. And we have also seen some examples by in which the combination of the, if else structure as well as the while structure are combined together to give us more powerful and useful programs.

Today we will start discussing about a new form of representation of data that is very useful and very fundamental that is called arrays ok. Arrays the word array simply means arrangement of data. So, we can think of for example, an array of soldiers or array of people standing in line.

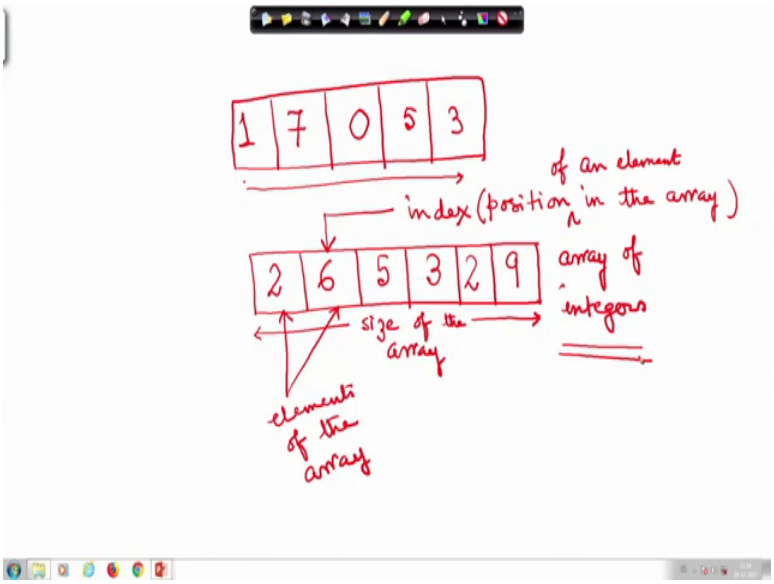
(Refer Slide Time: 01:42)



So, people are standing in a line, this is an array ok. We can also have 2 rows of people standing alright. This is a very regular structure we can have 3 rows also. So, these are arrays of people.

Now, if I just have only one row; in that case say for example, I consider one row then it is a one-dimensional array. When I consider rows as well as columns then it is a 2-dimensional array; however, we will now initially concentrate on one dimensional array like this ok. What is this? This is an array of people similarly, we can have array of numbers.

(Refer Slide Time: 02:54)



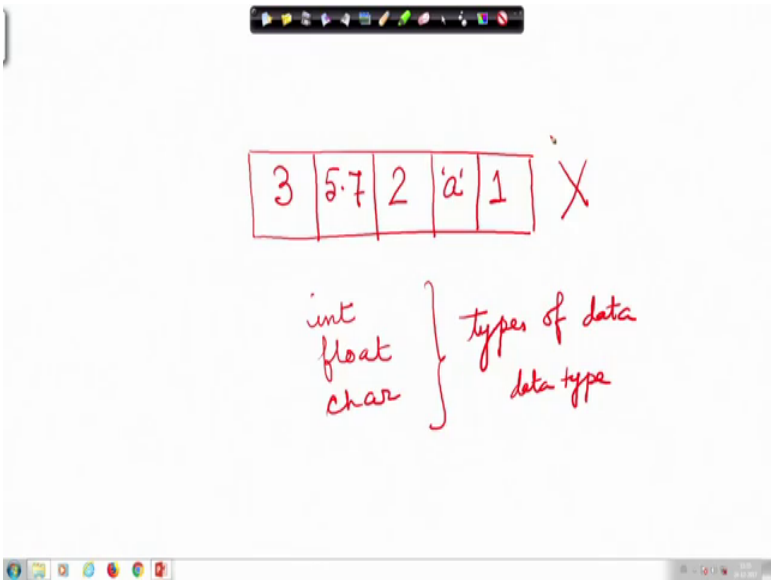
For example, 1 7 0 5 3 that is again an arranging of data arrangement of data in a linear near fashion ok, this is the first element and this is the last element of this arrangement. So, this is also an array, array of some numbers. So, in an abstract way we can say that an array can be drawn in this form it can be an array of integer or an array of real numbers depending on what I want to store here, suppose this b and array of integers. Then each of this places can have one integers stored in this 2 6 5 3 may be again 2 and 9 so this is an array.

Now, how any elements are there in the array? Each of these are elements of the array how many elements are there in the array 1 2 3 4 5 6. So, this is the size of the array and these are the elements of the array this all these are elements of the array. Also another thing we need to know is how do I identify one element of the array, say this element 6

of the array this array how do I identify that what is it is identification, the identification is the position. So, this is a second position alright this is the first position this is the third position 4th position so and so forth.

Now, this positions in the array is known as index. Index is the position in the array; index determines position in the array position of what? Position of an element in the array so, for the time being we assume that arrays are linear structures linear arrangement of data. Now one point to remember is one array can store data of the same type. For example, it is not possible to have an array.

(Refer Slide Time: 06:16)

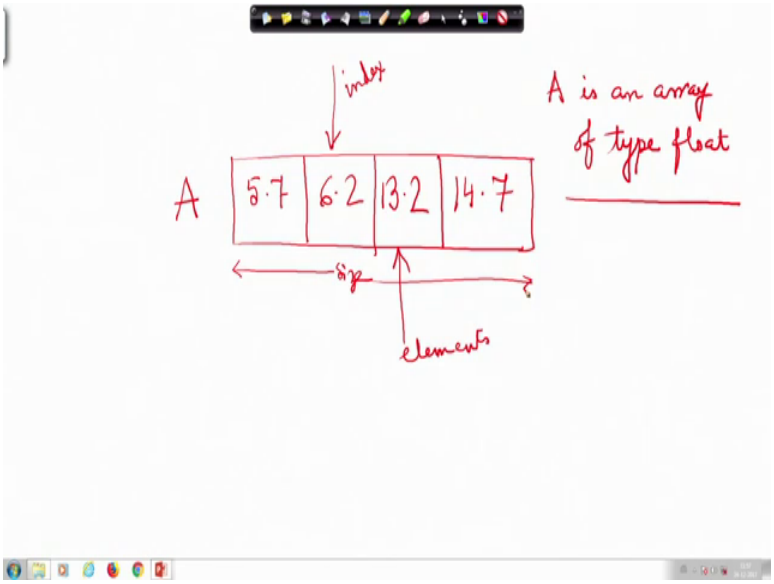


In which there will be some integers some real numbers and some again characters ok. So, that is not possible that is not allowed.

So, an array will is allowed to store data of only one type. We now that int float, char all these are defining different types of data or data type, right or data type. So, an array can store data of only one data type whatever that is it can be all floating point numbers, it can be all characters, it can be all integers whatever, alright?

So, these are 2 words of cautions. So, what have we learnt? We have learnt that array is an arrangement of data of the same type and the array has a is identified as a whole by a name and say for example, this array this is not a valid array so let us have a valid array.

(Refer Slide Time: 07:51)



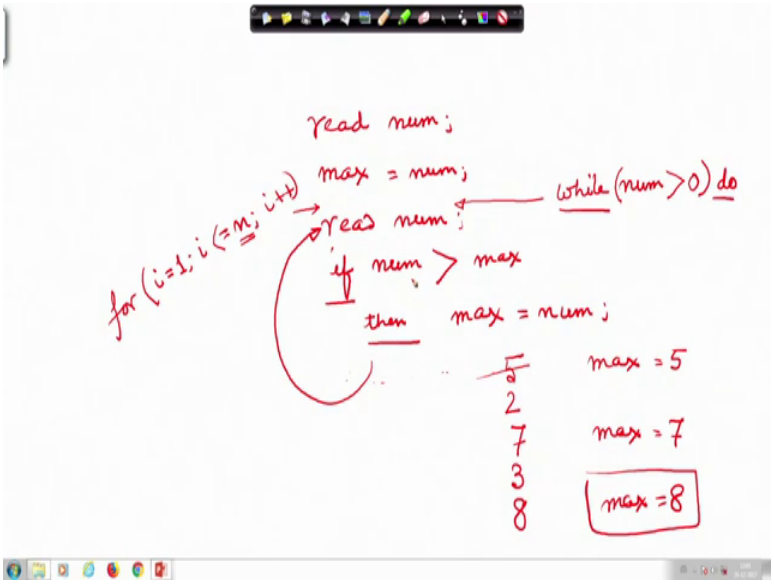
Let us have a valid array of 4 elements say of real numbers may be, right?

So, this array will have name let us say arbitrarily put a name A ; that means, A is the name to this array and this array has got a type, what is the type of this array? The type is float so array A is a float. So, A is a is an array of type float; that means, it can store only floating point data. Also, an array consists of different elements each of these are elements and each element each element can be a floating-point number. And the particular element is identified by the index of an array, alright?

Now, and also, we have we know that an array has got a size, given this let us think of why are we talking about all these things, why is this needed after all? We have encountered till now quite a few example problems for example; let us again come back to the old problem of finding the maximum of a set of integers.

Now, in that case what did you do? We read one integer we initialized a variable.

(Refer Slide Time: 10:01)



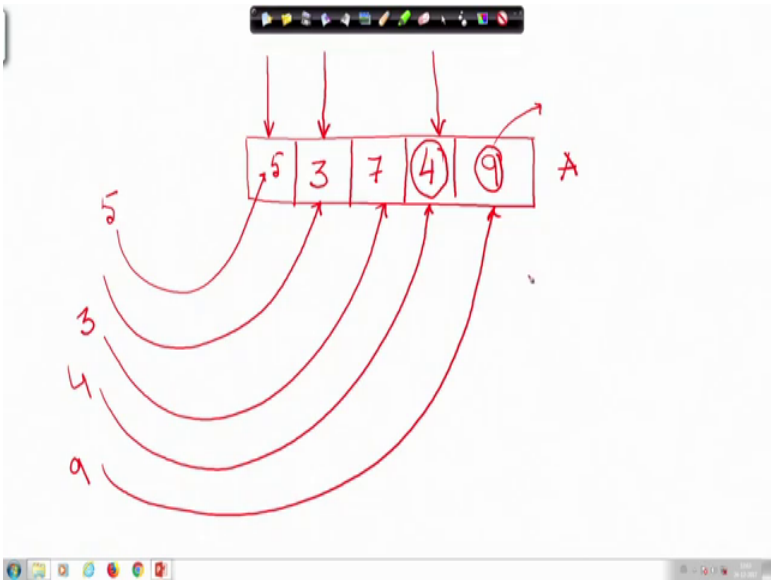
So, first integer we read let me once again do it. So, if you recall we I am writing the flow chart I am not writing the c program. Read a number you know how to read a number by scanf then put that number to be max read again read number. And if number is greater than max then I am writing pseudo code. So, I can write then max is getting the number, again I read number and in that way I go on or you can say that ok, why should I write in this way? I shall simply implement a loop wile loop or for loop. So, that this thing is carried on till I stop, say for n numbers or for ah. So, here I can use for I assigned 1 to I less than equal to n I plus plus to implement the loop here, but that is possible when I know the total numbered of numbers.

If I didn't know the number of numbers there are some other ways of doing it say for example, I am reading integers and I say when I want to stop I will enter a 0. So, here I could have put in something like this while num is greater than 0 do so we do this as long as num is greater than 0 I am going on doing this. And if num is 0 or num that we decide or some negative number I know that my job is done. So, but in this case if I want to remember what were the numbers already given to me suppose the numbers that were given were 5 initially here then 2. So, nothing was done next was 5 then 7 then here there has been a interchange. So now, here max becomes 7 then again it was 3. So, nothing was done then it was 8 and max became 8.

So, that will work for this sort of program, but if I want to know what where the numbers he was saying max is 8, but what where the numbers that where input by the user we have not remembered them anywhere is not it, you have not remembered them anywhere. So, here in my program, I will not be able to say what was the second number entered, what was the first numbered entered, was there any 0 entered or was there any anything greater than 3 entered, we cannot answer these questions because we read this and we operated on this and we forgot them.

Now, if you can think of that if I had used an array.

(Refer Slide Time: 14:02)

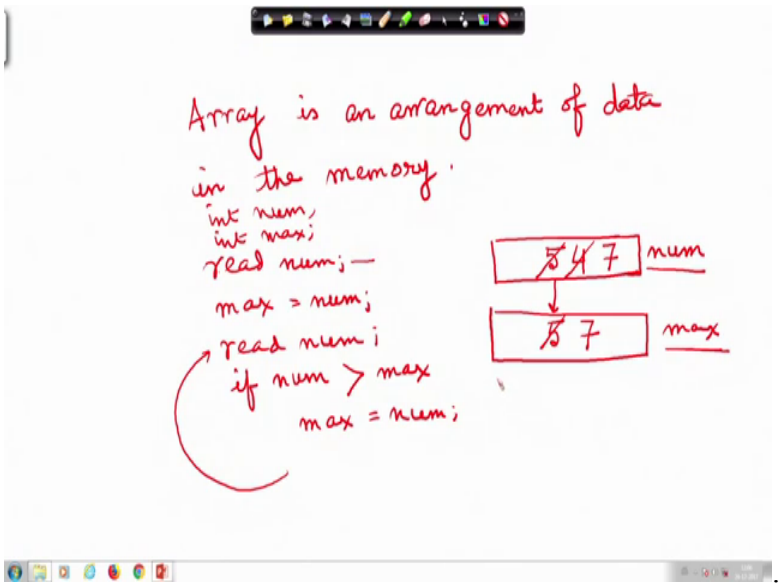


This sort of thing and I had as the user give the numbers I stored them here 5 then 3 user gave 3. I stored 3 here then user gave 7 I stored 7 here, then the user gave something may be again 4 I stored 4 here, and then the user gave 9 I stored 9 here. Then even after or 8 here I even after reporting the max that this is the max I have got these in my array. So, my name of the array is still A which A is an array of integers. Then this is stored here I have remembered them in the array.

So, I can now say what was my first element I go to the first element of the array using the index value ok, and second, I go to the this value ok. What was the 4th value? First second third 4th I come to this point and I can say the 4th value was this. So, all these things I can remember because I have been able to store them in the form of an array.

So, that is. So, array is again an arrangement in the memory alright.

(Refer Slide Time: 15:49)

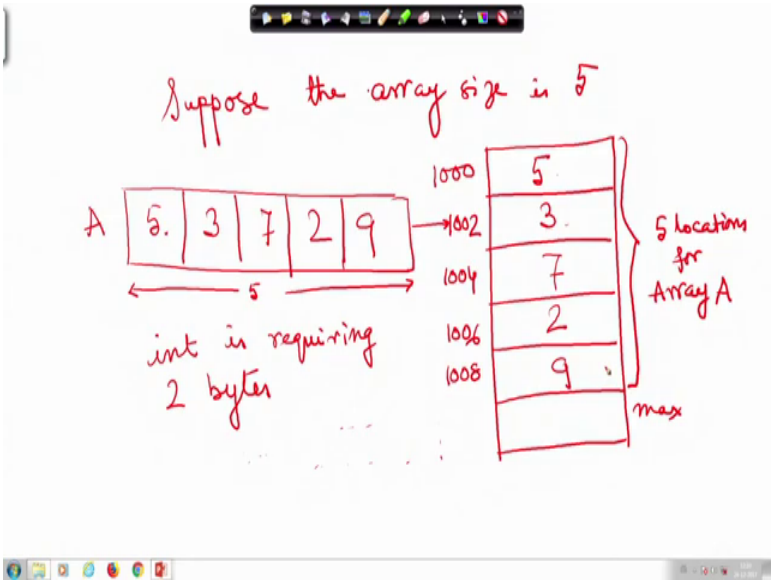


So, an array when I. You know that every variable is stored in a memory location, right? So, when we had num and the program that I had written that num initially I sorry I initially I read num. And then I do this and then again read num if num is greater than max, max assign num when I do this thing in a loop or whatever, what is happening? Whenever I have got variable num is a variable one location in the memory. This is num and max is another variable I have declared all of them here int num, int max I have declared them.

So, here there is another location which is known as max. So, when I did this read num and suppose some data 5 was read that 5 is stored here alright, 5 is stored here when I come here and I put max num assigned to max so it comes here. Now I read num again so suppose num is 4 and no update in max next time I read it 7. So, this is updated here and this becomes 7. So, this is the process that goes on alright.

Now, may so every time I am losing the old value of num and storing the current value of num and similarly whenever there is an update required I take the current value the max up to now ok. So, I am having 2 variables of integers, but if I had done it in the form of an array then I could have thought of like this that I have got suppose the array size is 5.

(Refer Slide Time: 18:58)



Suppose the array size is 5. So, that means, conceptually I am talking of something like this that there are 5 positions in this array. And as I am getting the data it is coming over here 5 3 7 2 9 whatever. Now actually internally what it is happening internally this is being stored in the memory the array size is 5 here. So, I have got 5 locations in my memory 4 5.

So, these 5 locations of the memory are being given to the array a suppose the name of the array is array A. So, for the array A so, the first element is coming here when I am actually writing it here it is coming here, second element is coming here, next element is coming here, next here next here so these are the memory locations. So, an array is actually a a representation in the memory which are contiguous and you know each memory location has got an address.

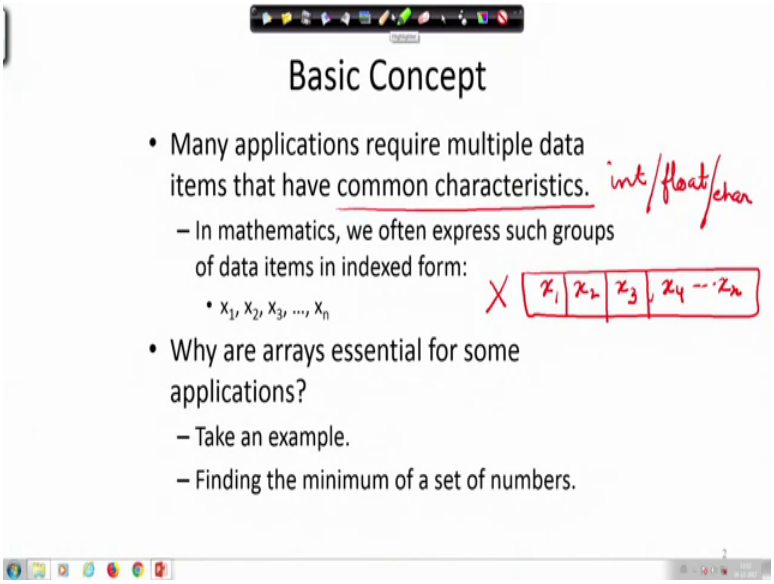
So, suppose hypothetically the address of this is 1000 and each of them are integers and integer if I say integer is requiring 2 bytes. Then if this is thousand location the next one will be 1002, next one will be 1004, then 1006, then 1008, 1008 to 1009, right?

So, 1000 to 1001 is for the first element this one coming here 1002 to 1003 for this element, 1004 to 1005 is this element. 6 to 7 is this element 1008 to 1009 will be this element. So, 1000 to 1009; that means 10 locations we are requiring for these 5 elements of the array. Now since also in my program I have got 2 other variables one is the num or one is the max, alright? I can manage it with one more variable max is separate variable.

Suppose. So, that is here this variable is max let me delete this for you this is not required, alright? It is because this diagram has re-ductant because max is another variable which I have here. Will write a program will show later how we can write a program to deal with this.

So, everything. So, since I have computed I have red the numbers I have stored them in the memory locations and since I am going through an array I am not written overwritten one number by another therefore, I can remember all the numbers. So, this is one basic reason of usage of array where I want to remember the numbers in the memory given this let us quickly recapitulate what we have said till now.

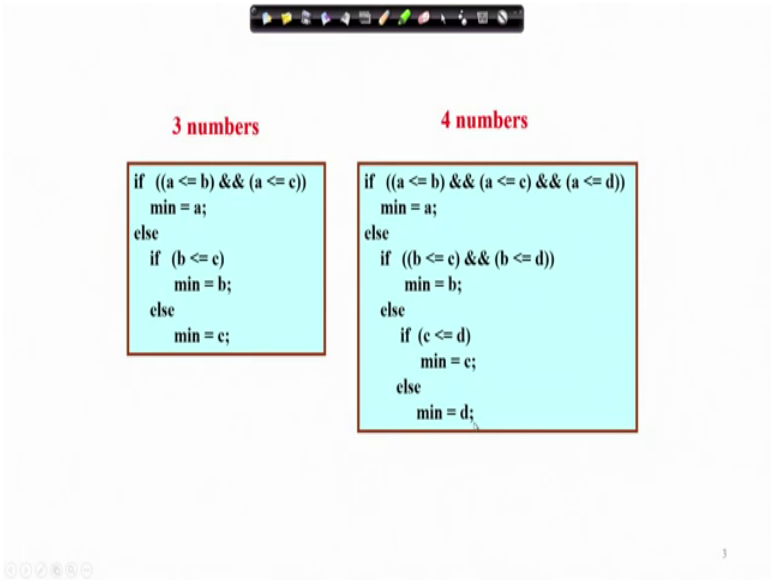
(Refer Slide Time: 23:08)



So, many applications require multiple data items, but have a common characteristics, what is that common characteristics? Common characteristics like whether they are integer or float or character whatever that is it can even more complex. So, we call say; so that numbers that we are showing say we can in mathematics we often represent them as x 1, x 2, x 3, x 4 say up to x n. And that hold thing I can represent as an array x where x 1 is the first element, x 2 is the second element, x 3 is the second third element like that

Now, here is the example finding the minimum set of numbers minimum of a set of numbers. How we can find a minimum of a set of numbers. So,. So, this is what I have already discussed.

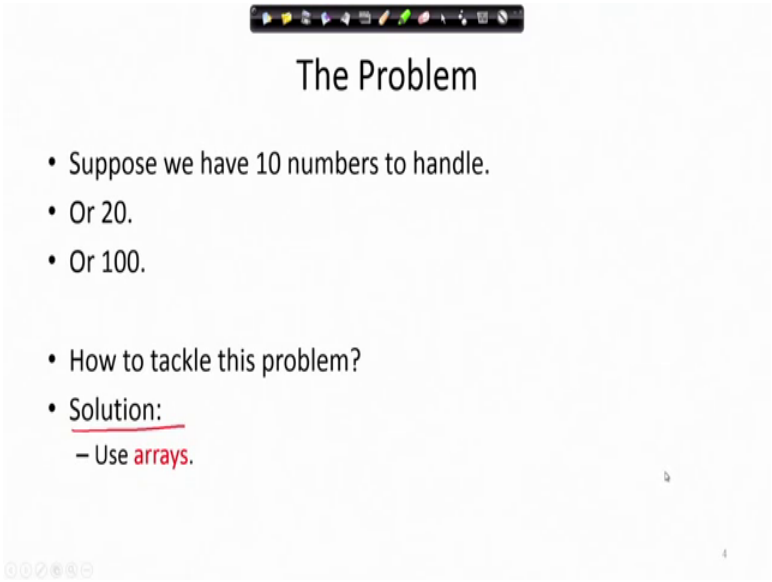
(Refer Slide Time: 24:20)



But just for the sake of revision let us look at it once again say for 3 numbers you know we can this program must be familiar to you now, you should be able to understand it quickly if a is less than b and a is less than equal to c then a is the minimum I was showing the maximum here it is an example of the minimum else if b is less than c then b is the minimum else c is the minimum.

So, first so fine as soon as I extend from 3 numbers to 4 numbers you see the program becomes a little bigger. If a is less than equal to b and a is less than equal to c and a is less than equal to d then minimum is a, otherwise if for you see if everywhere I have brought in more number of comparisons and the program code has also been bigger alright? And I am not storing the numbers as yet I am not storing the numbers that is one issue.

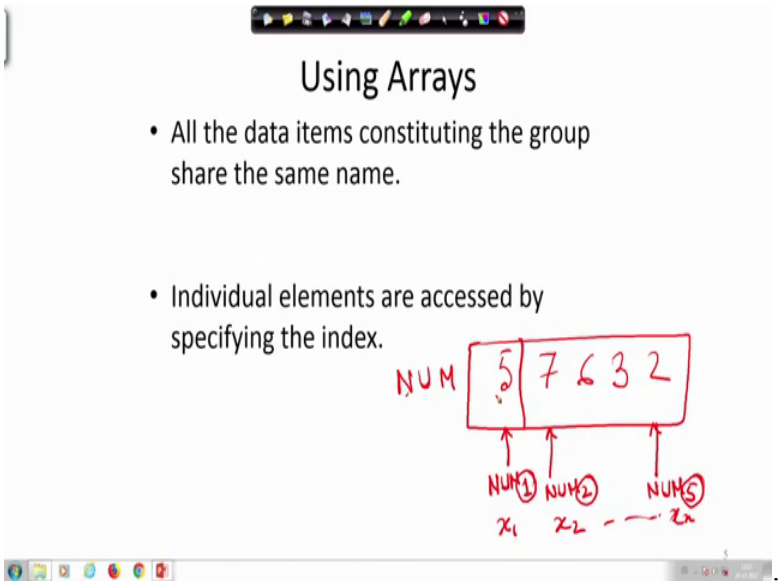
(Refer Slide Time: 25:30)



The problem is so from 3 to 4 we need needed so much extension.

Now, suppose we have got 10 numbers to handle that will be even bigger 20 numbers even bigger 100 numbers. So, how do you do that? The solution is the solution to the problem is what is a solution? The solution to the problem is use of arrays right using arrays, right?

(Refer Slide Time: 26:00)



Now as I said that all the data items constituting the group share the same name all these data items that I had shown say integers like 5 7 6 3 2 all these can share the same name x or may be let me give a meaning full name NUM.

Now, this one is NUM 1, this one is NUM 2, this one is NUM 5 ok. So, I have got just as we had shown x 1 x 2 up to x n so here n is 5 so I have got this 5. Individual elements are accessed by specifying the index; this is the index which is telling me in which position which number in this array of numbers I am looking at ok. We will look at this further the use of arrays and anymore things are to come.