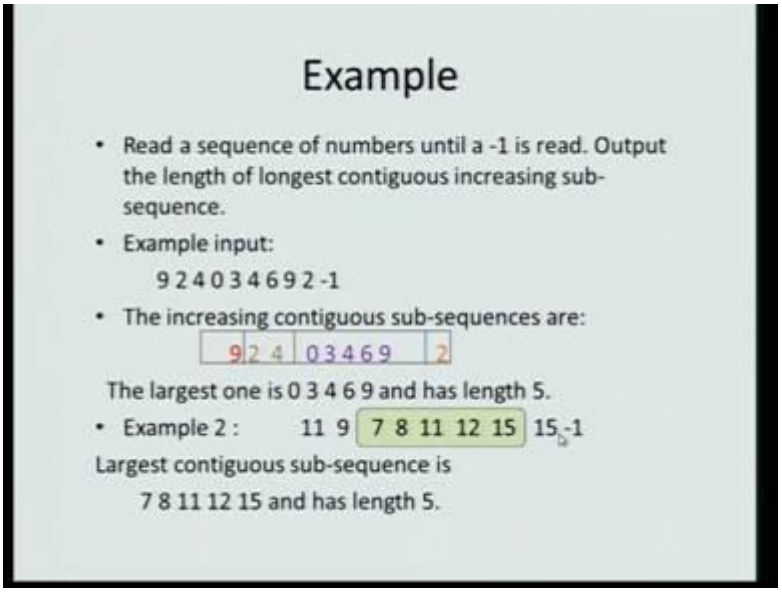
Introduction to Programming in C
Department of Computer Science and Engineering

So, we will see one more example for writing loops, we see as slightly trickier example and will cover this over multiple sessions. So, the problem is the following.

(Refer Slide Time: 00:17)

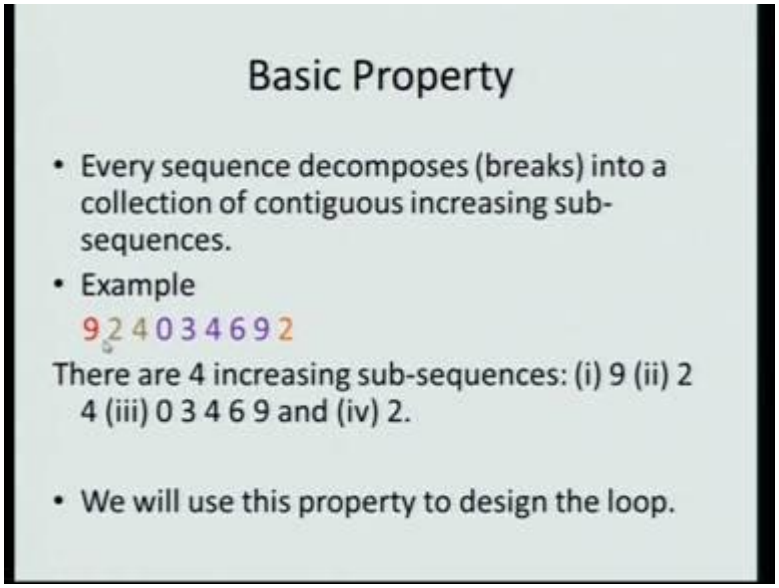


We have to read as sequence of numbers until a -1 is read, -1 indicates that then the sequence of sent it. And the question is we have to output the length of the longest contiguous increasing subsequence. So, that is a lot of words let us illustrate it with an example. Let us say that the input is of the following numbers - 9 2 4 0 3 4 6 9 2, and then finally a -1. And we have to output the length of the longest contiguous increasing subsequence. So, let us say what do I mean by a contiguous increasing subsequence? So, I say that 9 is an increasing subsequence, then the next number is 2; 2 is less than 9. So, 9 and 2 cannot be part of a subsequence, where the numbers keep on increasing.

So, 2 is the start of a new sequence, again the next number is 4. So, 2 and 4 form an increasing sequence. So, you can continue increasing a sequence. The next number is 0, 0 is less than 4, so break the sequence there. Then when you look at then succeeding numbers 0 3 4 6 9; they form an increasing sequence. And the last number is 2, which is lesser than 9. So, the increasing sequence stops here. So, these are the increasing contiguous subsequences; contiguous means together occurring adjacent to each other. So, the largest of the longest contiguous subsequence is obviously, 0 3 4 6 9, and the length of that sequence is 5.

Let us take another example 11 9 7 8 11 12 15 15 and -1. So, just to illustrate the point 11 is greater than 9. So, that cannot be an increasing sequence, 9 is greater than 7, so that is another the increasing sequence is just 9, but then 7 8 11 12 15; these are increasing. And I decided to stop here even though the next number was 15, because I am interested in and increasing subsequence. So, 15 and 15 are equal numbers. So, we break it ((Refer Time: 02:59)). So, the longest increasing subsequence is 7 8 11 12 15 and its length is 5. So, this the longest contiguous increasing subsequence.

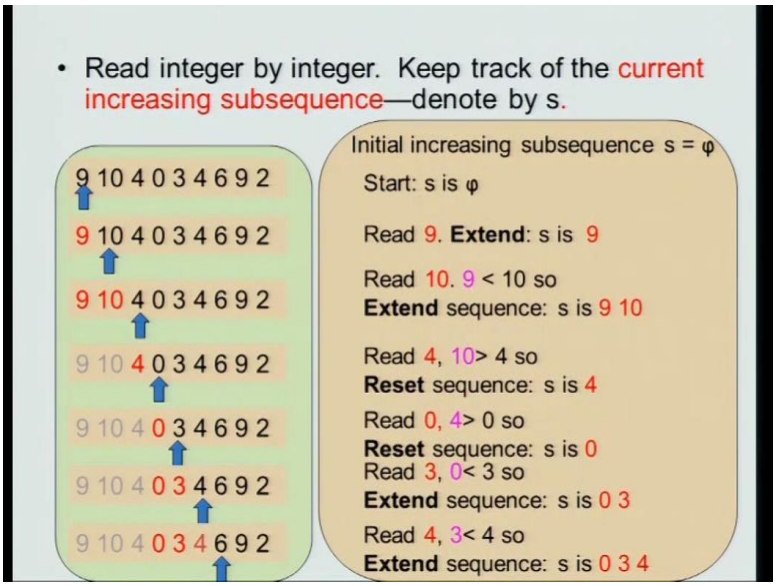
(Refer Slide Time: 03:23)



So, here is a basic property given any sequence of numbers, we can break it into a collection of increasing contiguous subsequences. For example the numbers that the sequence that we have seen. So, 9 2 4 0 3 4 6 4 6 9 2, and that the length of the increasing, the longest increasing contiguous subsequence is 5. So, we have to write a program to do this, given a sequence of numbers find the length of the longest increasing subsequence.

So, how do we do it? We do it in the way that we have been writing loops so far, like adding n numbers and finding this sum and so on. The idea was that you start from the first number and keep on adding the numbers until you hit -1 at which point you have this sum. So, the idea of this algorithm was that you start from the first, and you keep reading, until certain condition happens. We will adopt that idea to solve our current problem.

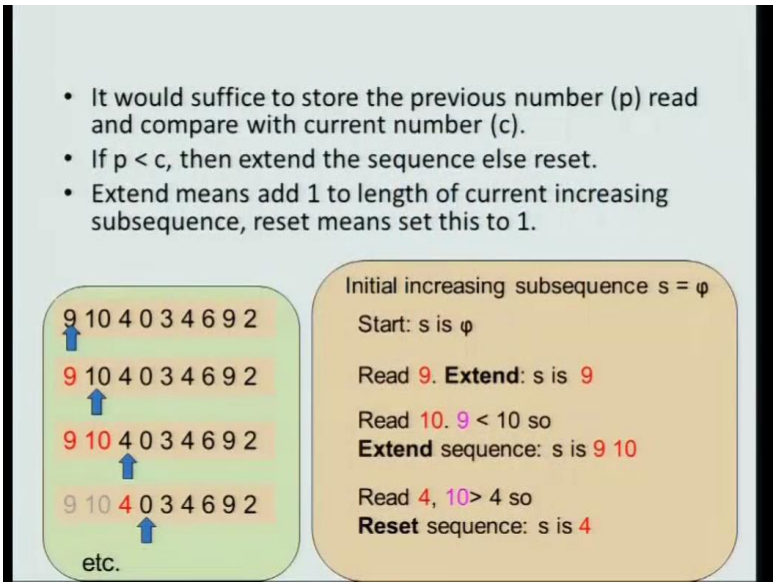
(Refer Slide Time: 04:28)



So, what we need to do is to keep track of the current increasing subsequence. Let us say that it is denoted by s. So, before we get into the code, let see how we will do it by hand. So, initially the increasing subsequence is s, and let say that it is empty. After you read 9, you have an increasing subsequence which consist of exactly 1 number. So, s is 9. Now, the next number is 10; 10 is greater than 9. So, you extend this. Read the next number 4; 4 is less than 10. So, 9 10 4 cannot be an increasing subsequence, therefore you say that you break the subsequence there, so 9 10 is a different subsequence.

Now you start a new subsequence which is 4. So, the current subsequence is just 4. So, 0 is less than 4, so you break it there, the current increasing subsequence become 0. Read the next number 3, 3 is greater than 0. See you extend the subsequence s is now 0 3 4, 4 is greater than 3. So, the sequence becomes 0 3 4 and so on. So, what are we doing here? We are reading the read, we are reading the numbers integer by integer, and we are keeping track of the current increasing subsequence. So, this is part of what we want to do?

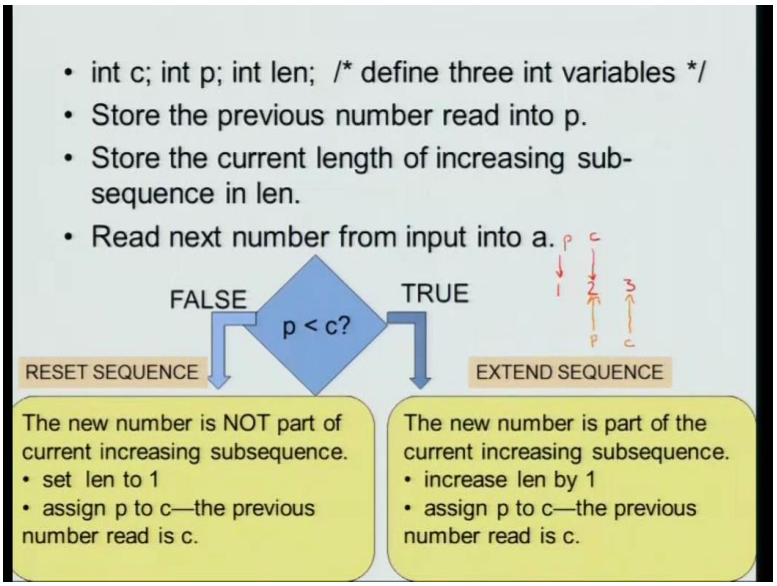
(Refer Slide Time: 06:16)



Now, in order to decided whether we should increase the subsequence extend it or break the subsequence, and start a new sequence what we need to do is we should remember what was the previous number we saw, and what is the current number that we have seen. So, with two variables we can say whether the subsequence should be extended or should be broken at this point.

So, if the previous number is less than the current number, then we should increase the we should extend the subsequence. If the previous number is greater than or equal to the current number then we should break the subsequence and start a new sequence. So, extend the sequence means add 1 to the length of the current increasing subsequence, and add that number to the subsequence. Reset means you start a new sequence of length 1.

(Refer Slide Time: 07:15)



So, from the current from the description that we have seen so far, we need the following variables, we need C which is for the current rate c number, p which is the previous number that we have seen, and length which is the length of the current increasing sequence. So, we store the previous number into p, store the current length of the increasing subsequence into length, and read the next number into c.

So, if the previous number is less than the current number. So, we take the true branch in which case we extend the sequence. So, the new number that we have read is part of the of the currently increasing subsequence. So, increase the length of the sequence by 1 and now we move lockstep. So, what we do is? So, we are a stage where suppose we have numbers 1 2 and 3, suppose p was pointing to 1, c was pointing to 2. So, since 2 was greater than 1, we extend the sequence. After extending the sequence, we have to proceed and see what will happen with the next number. So, when you do that you can do the following, I will extend the current sequence by doing the following, I will now set p equal to 2, and c equal to 3.

So, this is the idea that we will advance both the variables by 1 number each. So, that it is always true that previous is 1 number behind current. So, I hope this idea is clear that in order to ensure that p is 1 number behind current, you have to advance both p and c. So, assign p to c, this will advance p and then read the next number. So, that will become c. Now what happens if c >= p, then the new number is not part of the current increasing

subsequence. So, you start a new sequence which is of length 1 and again do the same assign p to c, which is advancing the pointer and read the next number. So, here is the method that we will follow in order to keep track of the current increasing subsequence. Now what is left is to find the longest of all the increasing subsequences that we find.