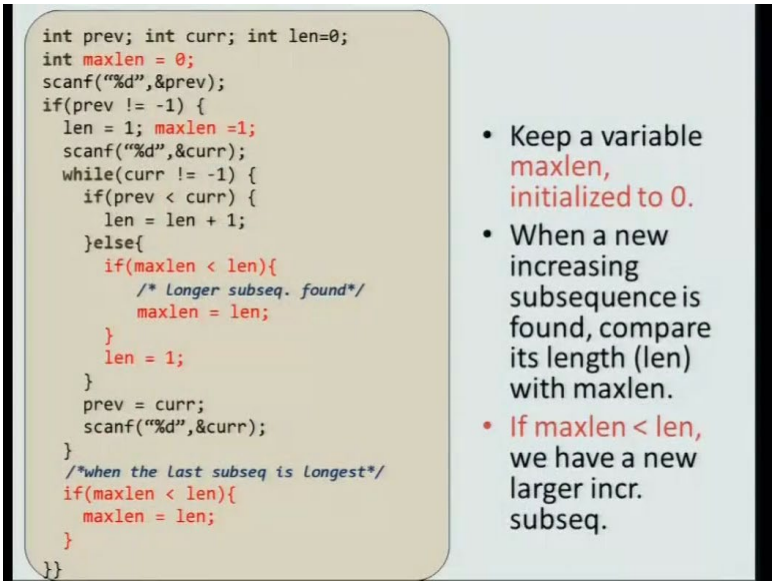
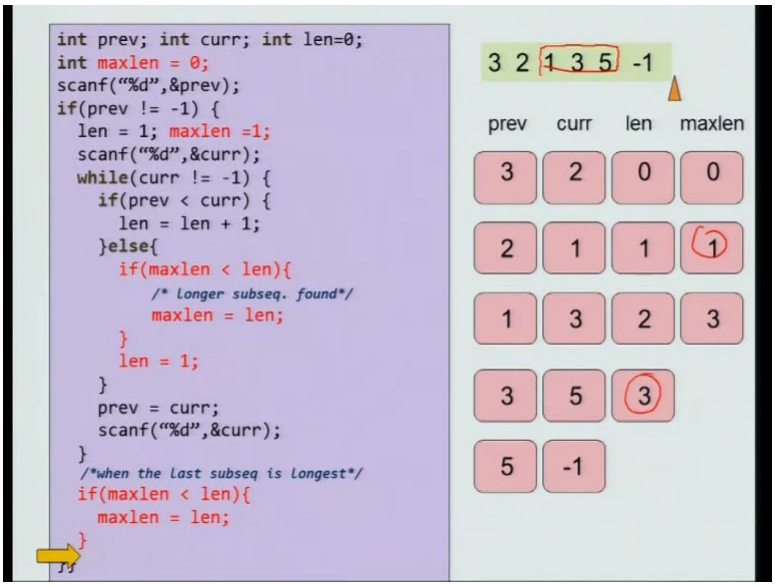
(Refer Slide Time: 00:27)



So, when we extent the sequence we don't have to do anything special when we break a sequence, and we start a new sequence, then all we have to do is you check whether the currently say the sequence that you just saw was longer than the previously known longest sequence. If that is the case then the sequence that just ended is becoming the longer sequence we have seen so far. Otherwise you maintain the max length. So, just forget about the currently stop sequence. Now there is a... So, that this loop, and at the end we have to do slight tricky logic, it could so happen that the sequence ends with a longest sequence increasing sub sequence. In that case, we will never reset the max length. So, if the last sequence is the longest, you also have to handle the case separately. So, we will see an example where, if you exit out of the loop that is you have already seen a -1, you just have to check whether the last increasing sequence that you saw was in fact the longest. So, there is a small if block at the end to do that.

(Refer Slide Time: 01:25)



In this part we will just see, small tracing of this program on a sample input. So, that the logic of the program become slightly more clear. So, I have picked a particular input 3 2 1 3 5 -1, and you will see that the longest increasing sequences are 3. So, the increasing sequences are 3, then 2, then 1 3 5. So, 1 3 5 is going to with a longest increasing sub sequence, and let us see how our program will find that out. So, initially you have a bunch of variables which should be declare. So, len = 0, maxlen = 0, and previous and current are undefined. Then you first read previous. So, previous becomes 3, it is not -1, so you enter the if condition, at which point you set length and max length to 1.

Now you scan the current number. So, current becomes 2. So, remember that previous is now 3, and current is 2. So, current is not -1, therefore you enter the while loop. Prev < curr is false, because previous is 3 and current is 2. Therefore, you enter the else part, maxlen < len is false; both are 1. Therefore, you start a new sequence with length equal to 1. Now you continue the loop with previous becoming current.

So, previous is now 2 and current you read the next number which is 1. So, previous and current have both more 1 step. So, current is not -1, prev < curr is again false, because 2 is greater than 1. So, you enter the else part. Max length and length there is no change. So, you reset the length to 1, previous is current. So, current previous becomes 1, and you scan the next number which is 3. Now at this point previous is 1, and current is 3. So, the if condition is true. So, you extend the length; length increases by 1. Again you advance

previous and current. So, previous becomes 3, current becomes 5. Again 3 is less than 5, so increase the length we are extending the sequence. So, the length becomes 3. Advance, so previous becomes 5, and current becomes -1 at this point you exit the loop.

And now you encounter the situation that max length, which is the length that we have seen so far, recall that it is one, but the length of the sequence that we just stop the see the input with is 3; that is that happen, because the longest increasing ((Refer Time: 05:09)) contiguous sub sequence, well was at the end of the input. So, it happen right at the end. So, when we exit the loop we have to do 1 additional check, we cannot simply say that the maximum length that we have seen in the sequence is 1, because max length is the length of the longest sequence we have seen before the current 1. The current 1 was the 1 that we just stop to with it had a length of 3. So, we just check, if max length equal to length is less than the length, then we set max length to be the length. So, once you do that max length becomes 3. This is just to handle the case when the longest increasing sub sequence is the last. Now you can exit out of the exit out of the if condition, and then print that the maximum length that you have seen is 3.