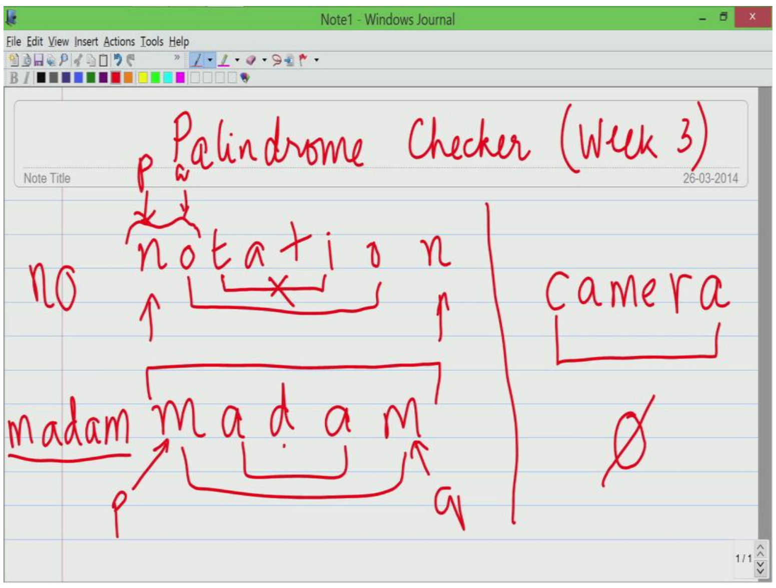
Programming, Data Structures and Algorithms

Prof. Shankar Balachandran
Department of Computer Science and Engineering
Indian Institute Technology, Madras

Lecture – 27
Solution: Third Week Programming Assignment

Welcome to the solution section for week 3. So, in this tutorial section what we will do is we look at this problem of palindrome checker. So, for the first time we gave a piece of code that gets automatically appended, and that piece of code was shown to you as commented section, and we said fill in the code to do the work. So, let us look at the problem statement first.

(Refer Slide Time: 00:36)



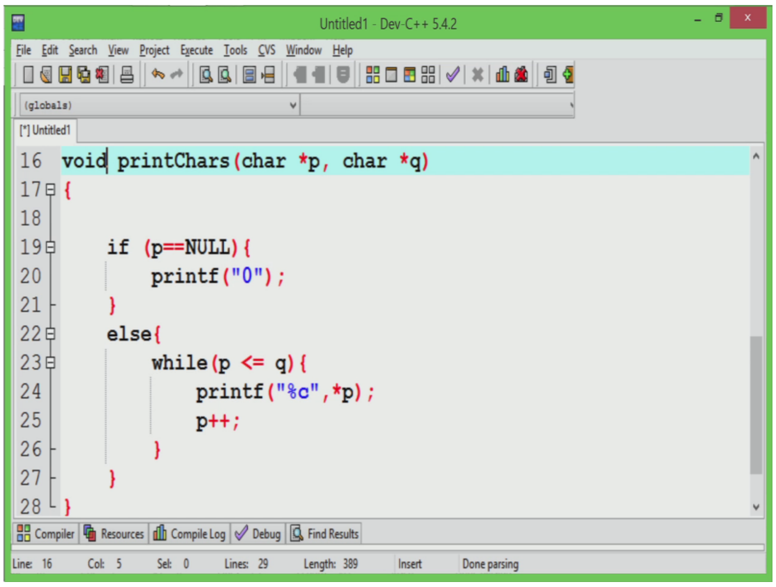
So, let us look at a few examples and let us understand what the problem is actually asking for. So, I am taking some of the things from the public test cases. So, let us take this input called notation. And what we are asked to do is we are supposed to check if, is a given this input notation, is supposed to check matching characters at both ends.

So, for example, in this n has a matching n on that side, and o has a matching o on that side. And the moment you go to t, this does not match any more. And what we are asked to do is, print the prefix, the longest prefix in the input string itself which has a matching thing on the other side. So, in this case, the prefix n o has a match on the other side namely o, n.

So, let me take another example which is actually a palindrome. If I have m a d a m, madam, then m has a matching m, a has a matching a, and d is sitting in the middle. In this case, the whole string is the prefix itself. So, the whole string has a matching reverse, matching string at the reverse. So, let us take a few other examples. Let us take examples where they are not even palindromes. So, for example, there is a public test case says camera. So, even the very first letter does not match. So, the longest prefix is actually nothing.

So, what we wanted you to do in this problem was take these cases and print the longest prefix. So, for this input notation the longest prefix would have been n, o; for this input madam, the longest prefix would have been m a d a m. And if you look at this input camera there is no prefix at all, where there is a matching suffix in the reverse. So, in these cases we ask you to with print number 0. So, this is the deal. And how do we ensure this?

(Refer Slide Time: 02:51)



```
void printChars(char *p, char *q)
{

    if (p==NULL){
        printf("0");
    }
    else{
        while(p <= q){
            printf("%c",*p);
            p++;
        }
    }
}
```

Untitled1 - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
(globals)
[*] Untitled1
Compiler Resources Compile Log Debug Find Results
Line: 16 Col: 5 Sel: 0 Lines: 29 Length: 389 Insert Done parsing

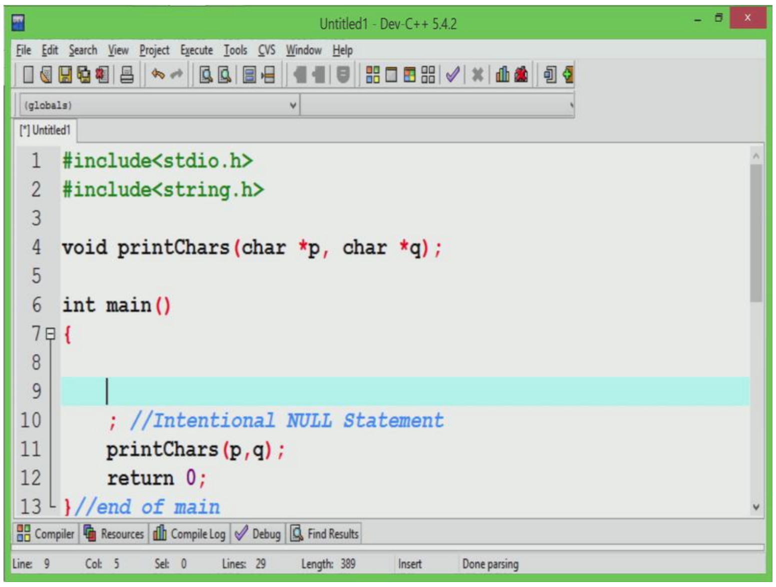
We gave you this function called print characters. Print characters is supposed to take 2 pointers, 2 character pointers p and q. And if you make p equals null, it will printf 0, right. So, what this is supposed to do? What you are supposed to do with this is, if you see that as given string is not a palindrome, you should somehow ensure that p is passed as null because that is what we are expecting in the output. If the input is not a palindrome which means not even a single character from the beginning and end is matching then you just somehow you should ensure that p is passed as null and this function will print 0, otherwise let us see what this function is doing.

So, else while p is less than or equal to q. So, we are assuming that both p and q are pointers to an array because a string is an array of characters, and p is supposed to give the starting location and q is supposed to give the ending location for the prefix that you want to print. So, for example, in if you process notation at the end if you somehow ensure that p is n and q is o, you will end up printing n o which is the longest prefix.

However, for an example like madam, we should ensure that p should point to this m and q should point to this m. Somehow you should process and finally ensure that p and q are these because if you give p as this and q as this and call print characters, then you will get the whole string m a d a m printed. So, what this does is, if goes p less than or equal to q print star p and p plus plus.

So, what it is doing is, it is starting from whatever is pointed to by p, it prints that character. And p plus plus, remember it is a character pointer. So, when you do p plus plus it goes 1 character at a time till it reaches q. So, it also prints what is pointer to by q and it stops. So, this is what the print characters is doing. So, I am staring with the template that we gave you.

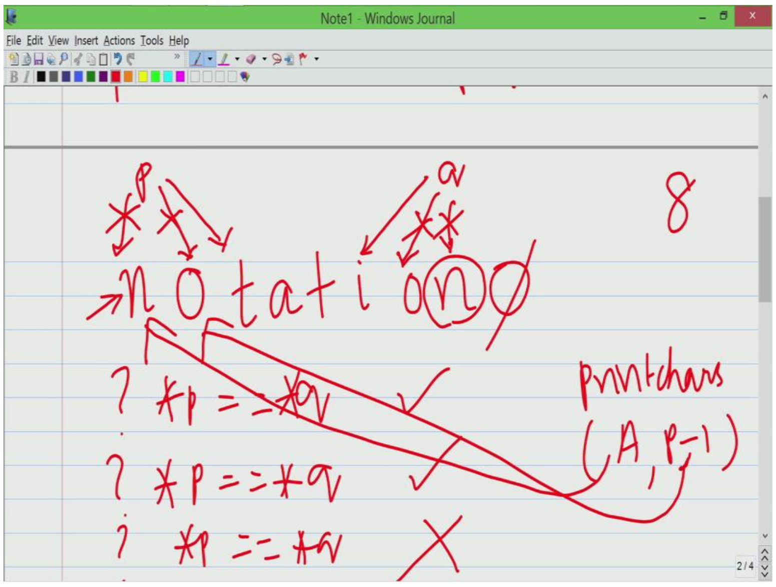
(Refer Slide Time: 04:35)


Untitled1 - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
(globals)
[*] Untitled1
Compiler Resources Compile Log Debug Find Results
Line: 9 Col: 5 Sel: 0 Lines: 29 Length: 389 Insert Done parsing

```
#include<stdio.h>
#include<string.h>

void printChars(char *p, char *q);

int main()
{


    ; //Intentional NULL Statement
    printChars(p,q);
    return 0;
}//end of main
```

So, we gave you this, hash include stdio dot h and hash include string dot h, to do any string operations. We put the prototype called print characters, character star p and character star q because it is a function that we are using later we need to say what is the prototype and we have the int main. We put a intentional semicolon followed by a function called printchars. So, this function call somehow before this null statement we have to ensure that p and q are appropriately chosen. So, let us go back and look at this, these inputs once more.

(Refer Slide Time: 05:44)



So, let us start with notation. How do we write or how do we solve this problem? So, in the worst case, right, the whole string could have been a palindrome. So, what I am going to do is I am going to assume that there is a pointer p that points to the very first location and q which points to the last valid location. So, remember, you are reading as a string you would have a null character at the end, but we are not going to use it. So, the valid characters are 1 step before that, right.

So, if I start with p here and q here, the first thing I will do is I will go and see if there is a match, right. So, is star p equals star q? So, that is a question. I am asking this question, is star p equals star q? So, in this case, star p and star q are both equal which means we have 2 characters at the 2 ends which are matching. So, atleast there is a prefix which is of non zero length. So, this now we can; so we should record this. So, is somewhere we know that star p and star q are the same.

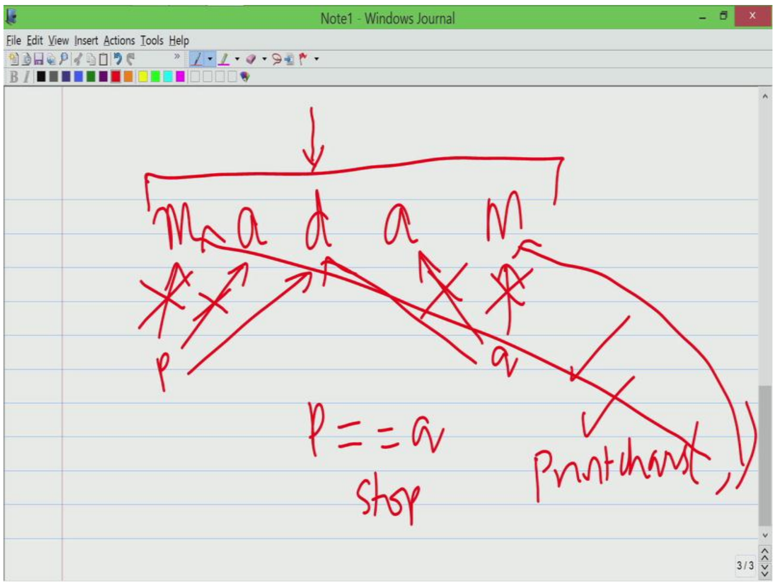
Then what we have to do is we will have to go and see if the prefix can be extended. In this example it can be extended by 1 more character, right. So, if you do p plus plus, p will start pointing to o; and if you do q minus minus, it will start pointing to this o. And again I can, we can ask this question, is star p equals star q? So, at this point p is pointing to this o, and q is pointing to the o at the end. So, at this point, is star p equal star q? Yes indeed. So, this is true and this is true. So, the prefix is extended by 1 more step.

Now, let me extend p to 1 more step further and let q go back 1 more step. So, p and q will now point to t and i. At this point if we ask this question, is star p equals to star q, the answer is incorrect. So, it is not true. So, what we want is we have found one position in which the matching position at the other end does not have a matching character. So, t is matching position is i on that side, but t is not the same as i. So, at this point, you can notice that p is stopping 1 step away from the actual prefix. So, p is stopping at t which is 1 step away from the matching prefix.

And what do we want to do in this example? So, you have n o that has to be printed. So, somehow if I make, I am going to call print characters, right. So, for print characters it needs 2 pointers, and the first pointer should point to the beginning of the prefix which should be this and the second pointer should be pointing to the end of the prefix which is this. However, so beginning of the prefix is easy to get. It is just ampersand of a of 0, or a itself.

But, the end of the prefix we have p which overstep by 1 step. So, what we need is if we call print characters with p minus 1 and a, right. So, a, would be the beginning of the prefix, and p minus 1 would be the end of the prefix. So, somehow if we ensure that if we can call with, a and p minus 1, our job is done. So, this is for notation.

(Refer Slide Time: 09:26)



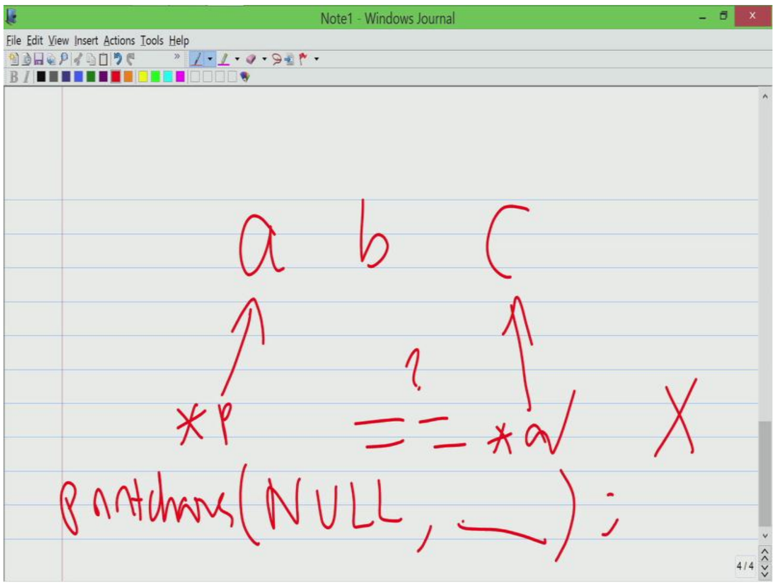
Let us look at another example. So, let us look at m a d a m. Again, will start with these 2 locations; at this point is star p equal to star q? Yes. Then p will, so these are for p and q; then p moves 1 step and q moves 1 step back; at this point, is star p equal star q? Again, yes. Now, p points to this location and q would point to this location, actually we have reached the middle of this string. When we reach the middle of the string, so this characters would actually be the same, right.

So, in this example, it is actually a palindrome. So, if we want to check if something is a palindrome or not, you do not have to go beyond the middle of the string because once you go beyond the middle of the string, we have already matched the left side with the right side, now we will match the right side to the left side; we do not really need to do it. So, we can stop when we are in the middle of the string.

So, in this case, when p equals to q, we can actually stop. And at this point we still need to say what is the whole prefix, right. So, in this case the prefix is expected to be the whole thing. So, what we want is we want the first pointer that we are calling to print characters, should be here, this m; and the second pointer that you pass should be to the end one, right. So, somehow you should ensure that these 2 pointers to the beginning and end. So, this is something that we need to do.

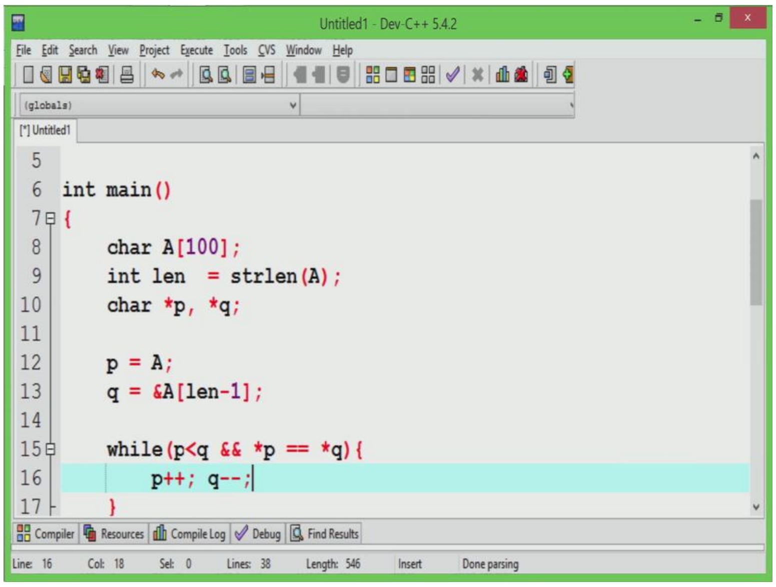
So, if something is a full palindrome we need to ensure that the pointers that we pass to print characters are to the first and last character. If we have something like a prefix, like in this example n o and o n, we should ensure that the first pointer is to the first location and the second pointer is to the prefix; in this case, it is o.

(Refer Slide Time: 11:28)



So, finally, let us look at one example. Let us say, a b c. If a b c is given as input you will start with p as this and q as that, and star p is actually not equal to. So, star p equal to star q, if we ask that question, this is not true. So, you can as well say that there is no prefix. And for this case, we wanted print characters of. So, when we call print characters we should somehow ensure that we pass a null pointer here and anything that you want here, right. It does not matter what you pass because print characters we can see is written in such a way that if p is null, it is already printing 0.

(Refer Slide Time: 12:04


Untitled1 - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
(globals)
[*] Untitled1
Compiler Resources Compile Log Debug Find Results

```
5
6  int main()
7  {
8      char A[100];
9      int len  = strlen(A);
10     char *p, *q;
11
12     p = A;
13     q = &A[len-1];
14
15     while(p<q && *p == *q){
16         p++; q--;
17     }
```

Line: 16 Col: 18 Sel: 0 Lines: 38 Length: 546 Insert Done parsing

So, now, let us take this and see how we want to write a program for this. So, the first thing we want is we want to be able to read a and the input specification said at most 99 characters will be in there. So, I will declare an array, A, of size 100 because I want to accommodate back slash 0 which is the end of the string. So, I want to accommodate that. And I am going to track, where the end of the string is. So, the length of the string is, you can use the function called strlen of A, this is a string function available in string dot h. So, I am going to declare an integer called length which tracks the length of A.

So, if you go and look at this input, let us say notation, so what is the length of the string? So, let us count it. So, 1, 2, 3, 4, 5, 6, 7, 8, there actually 8 characters which are valid characters. So, the length of this string is 8. However, it uses 9 locations. This location would be a of 0, and this location would be a of 7, this null character would be a of 8.

So, what we are tracking here is just the length of the string. It is not the last valid position though. So, if we subtract one from length it will give you the last valid position, right. So, now, it is passing p and q. So, if we notice print characters here, at line number 13 currently, you need to set up p and q. So, we need 2 pointers, character star p and star q. So, somehow we should ensure that p and q follow the appropriate ones, right.

So, what we, what did we do in the algorithm? We said we will set up p to start at A which is the beginning of the array. And q should point to the last valid character; so where is the last valid character? It is at length minus 1 position, but it is not the character A of length minus 1, it should be the ampersand of A of length minus 1. We want the address of the last valid character. So, I put a ampersand of A of length minus 1. So, so far, so good.

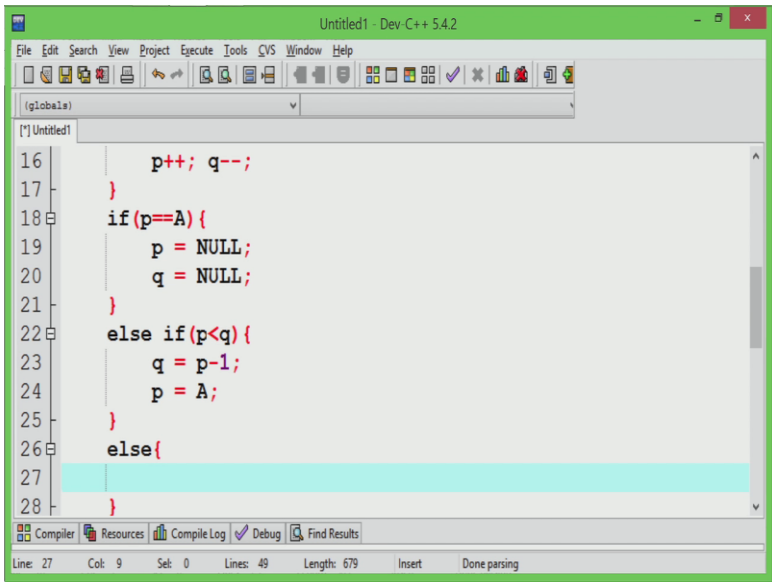
Now, I want to go and write this program which will ensure that this whole thing is going to run in a loop. So, as I said before there are 2 conditions that can happen, I am going to start. So, again if we go and look at this. So, we start with p and q. As long as p and q, they do not cross each other. So, p is to the left of q and we can keep checking if star p is the same as star q.

If you look at madam, they did not cross each other, but they touch the same location. So, we do not want that. So, if it touches we know that it is already a palindrome, right. So, I am going to write this condition. While p is less than q which means p is still to the left of q, p has not collided with q nor has crossed over. So, this loop is trying to maintain or this loop will run only when p is to the left of q, not when p is to the right, not when p is the same as q.

So, now, if p is to the left of q and star p equals to star q, I said we can move p by 1 step to the right and q by 1 step to the left, right. Now, this while loop will keep running till one of the conditions fail; so either p less than q can fail, or star p equals to star q can fail. If p is less than q, so let us say, if star p is equal to star q failed, it means that at this point your palindrome is not matching any more. You had a prefix till now, but at this point you had a character.

For example, in notation, star p would have been t and star q would have been i, and p is still to left of q, you would have failed that condition, right. So, whenever you have some prefix, but something else fails at some other location, this star p equal to star q would fail. However, this p less than q will fail when p is either equal to q or when p becomes either greater than q. So, we will have to check various conditions. So, let us start with the first one.

(Refer Slide Time: 16:53)


Untitled1 - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
(globals)
[*] Untitled1
Compiler Resources Compile Log Debug Find Results
Line: 27 Col: 9 Sel: 0 Lines: 49 Length: 679 Insert Done parsing

```
        p++; q--;
    }
    if(p==A){
        p = NULL;
        q = NULL;
    }
    else if(p<q){
        q = p-1;
        p = A;
    }
    else{

    }
```

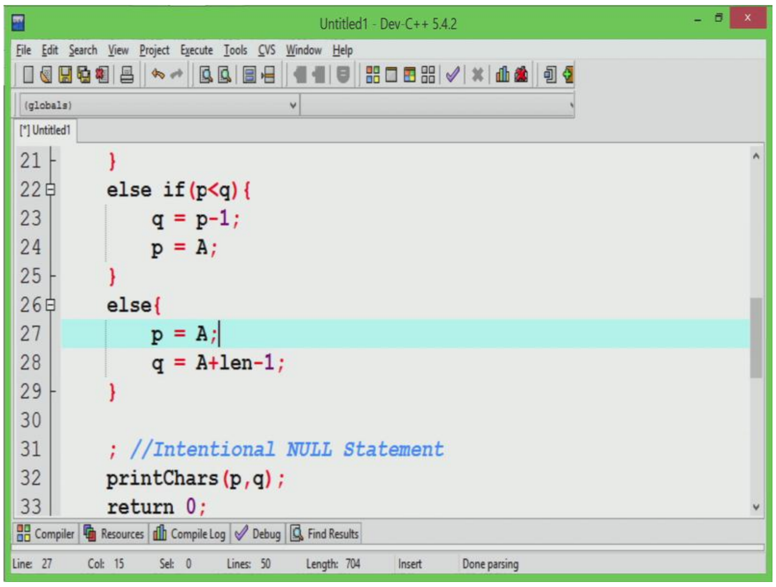
It is possible that if I gave you an input like camera, right, it never goes anywhere. So, p does not even increment and q does not decrement even 1 step. So, that is the first thing I am going to check. If p is actually the same as A itself, it means it is not a valid palindrome. So, even the first characters do not match. In this case, we wanted to print 0, and we are going to achieve that by the making p equal to null, right. So, once you pass p equal to null, print characters will take care of printing 0.

So, just to be careful I am also going to make q equals null. So, this is the first check. If, for in, so the example I gave, right, camera. For camera p would still remain at A, therefore, we want to print 0. Now, what are the other things that can happen? We can have p less than q, right. So, else if p is less than q, we know that the while loop failed because of star p not being equal to star q because; see, look at this point, right; so at this point p is less than q and star p is equal to star q. If you came out of the while loop and if p is less than q is still true then star p should not have been equal to star q that is the only thing that is possible.

So, if that is the case then we know that p over stepped by 1 location like in notation, right. So, p over stepped, so I need to go till p minus 1th location and print that, but I have to start from the beginning, right. So, that is what I am doing here. So, when I put q equals p minus 1, I take 1 step back and say, print only till this matching prefix, but where should I start it? I should start it at p. So, I should start it at the beginning of the string. So, that is A.

If both these conditions are not true; so p is not equal to A which means p is actually proceeded somewhat, and star p is still equal to star q. So, which means p is less than equal to q is not true then what could have happened is you have mechanism by which you have a complete palindrome and the whole thing collided at, let us say, in madam, p and q collided at p, right.

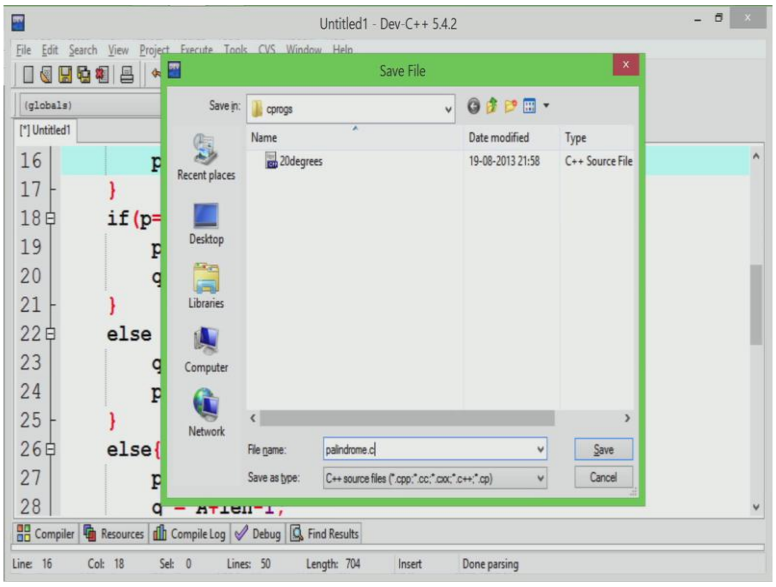
(Refer Slide Time: 19:12)


Untitled1 - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
(globals)
[*] Untitled1
Compiler Resources Compile Log Debug Find Results
Line: 27 Col: 15 Sel: 0 Lines: 50 Length: 704 Insert Done parsing

```
    }
    else if(p<q){
        q = p-1;
        p = A;
    }
    else{
        p = A;
        q = A+len-1;
    }

    ; //Intentional NULL Statement
    printChars(p,q);
    return 0;
```

So, this could have happened, right. So, if p and q collide then we know that the whole string is a palindrome which means I should start from the beginning of the string and go till the end of the string, right. So, if I do that, this would be a palindrome, right. So, now, I am going to take this.

(Refer Slide Time: 19:33)



Untitled1 - Dev-C++ 5.4.2

Save File

Save in: cprogs

Name	Date modified	Type
20degrees	19-08-2013 21:58	C++ Source File

File name: palindrome.c

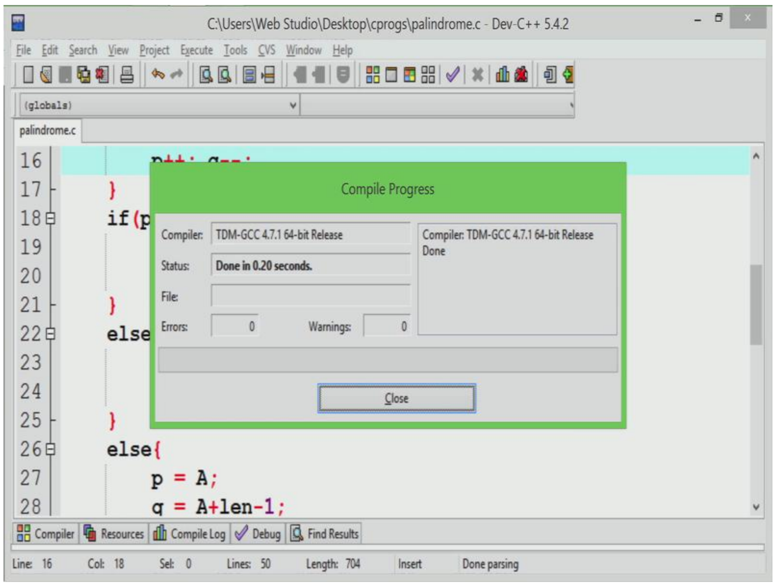
Save as type: C++ source files (*.cpp;*.cc;*.cxx;*.c++;*.cp)

```
        p
    }
    if(p=
        p
        q
    }
    else
        q
        p
    }
    else{
        p
        q = A+len-1;
```

Line: 16 Col: 18 Sel: 0 Lines: 50 Length: 704 Insert Done parsing

So, I save this as palindrome dot c. I have saved it.

(Refer Slide Time: 19:41)



C:\Users\Web Studio\Desktop\cprogs\palindrome.c - Dev-C++ 5.4.2

Compile Progress

Compiler: TDM-GCC 4.7.1 64-bit Release

Status: Done in 0.20 seconds.

File:

Errors: 0 Warnings: 0

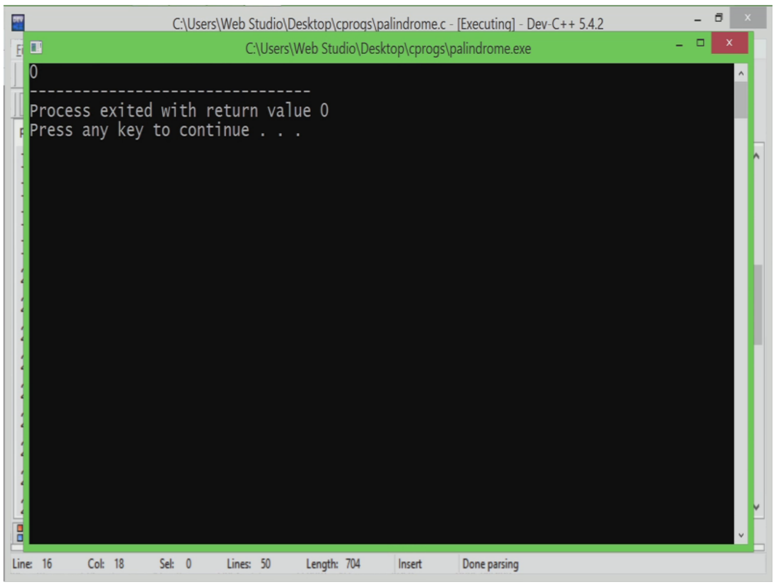
Compiler: TDM-GCC 4.7.1 64-bit Release
Done

```
        p++; q--;
    }
    if(p
    
    
    }
    else
    
    
    }
    else{
        p = A;
        q = A+len-1;
```

Line: 16 Col: 18 Sel: 0 Lines: 50 Length: 704 Insert Done parsing

Let me compile it. So, there is no compilation error.

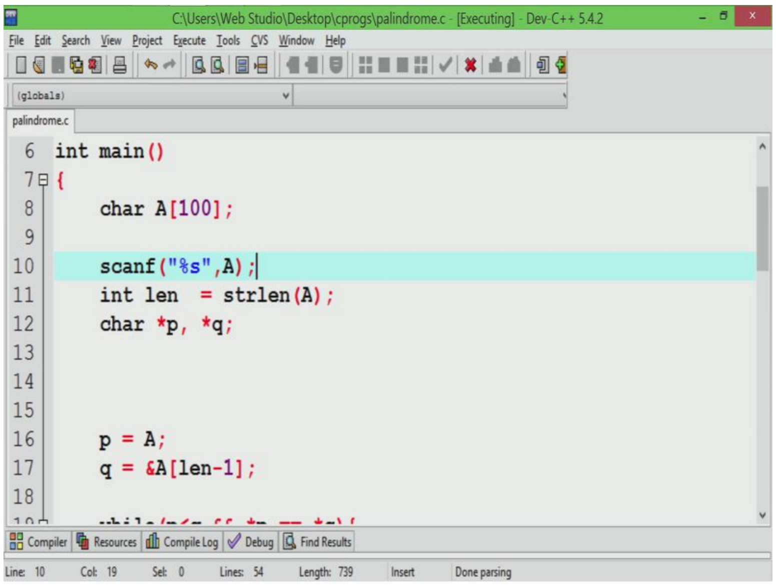
(Refer Slide Time: 19:44)



```
0
--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Let me run it, right.

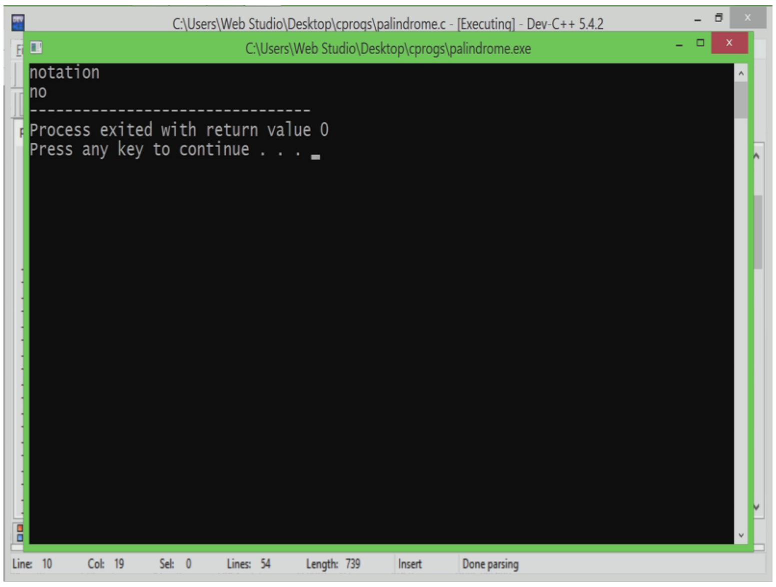
(Refer Slide Time: 19:46



```
int main()
{
    char A[100];

    scanf("%s",A);
    int len  = strlen(A);
    char *p, *q;



    p = A;
    q = &A[len-1];

```

So, it looks like I did not scan the input at all. So, let me scan the input also. So, I just declared it and I forgot to scan it. So, scanf percentage s A. Also, it is going to take input from the user. So, let me compile and run it.

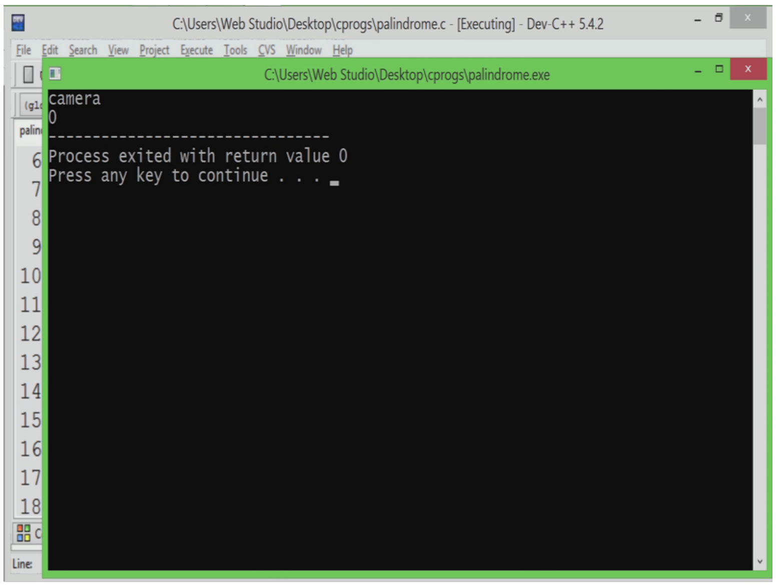
(Refer Slide Time: 20:07)


C:\Users\Web Studio\Desktop\cprogs\palindrome.c - [Executing] - Dev-C++ 5.4.2
C:\Users\Web Studio\Desktop\cprogs\palindrome.exe
Line: 10 Col: 19 Sel: 0 Lines: 54 Length: 739 Insert Done parsing

```
notation
no
--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

So, at this point I am going to give notation. So, n o t a t i o n, I am sorry. So, I will run once more, not that it matters, but. So, for notation it prints no which is good.

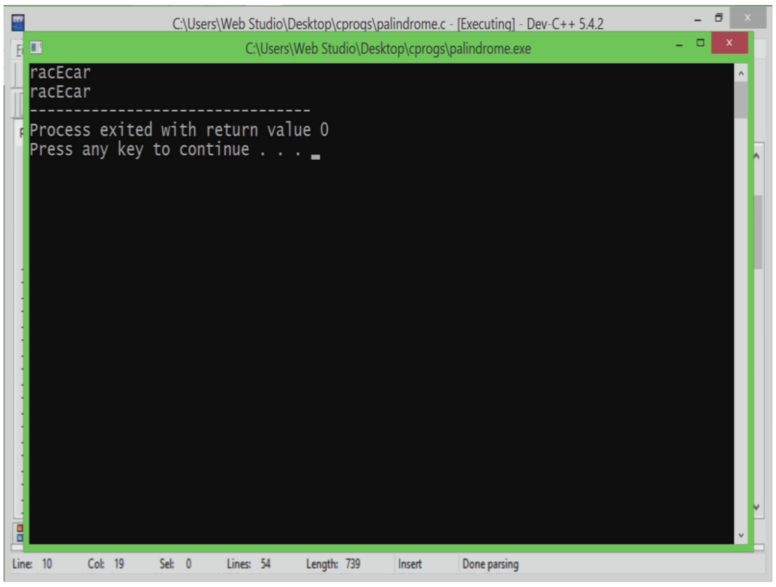
(Refer Slide Time: 20:21)


C:\Users\Web Studio\Desktop\cprogs\palindrome.c - [Executing] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
C:\Users\Web Studio\Desktop\cprogs\palindrome.exe

```
camera
0
--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Camera, it prints 0 which is good. So, let us look at the other public test cases.

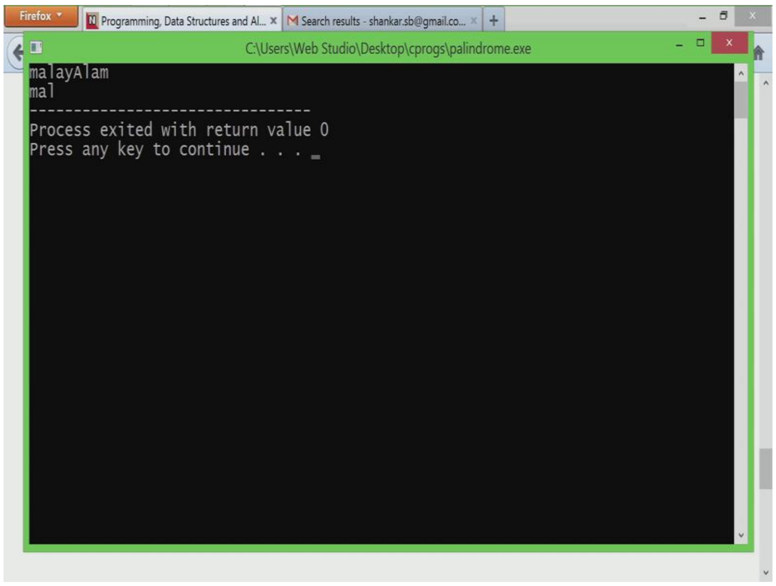
(Refer Slide Time: 20:37)



```
racEcar
racEcar
--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

So, let us look at race car, r a c E C A R. So, this is ok.

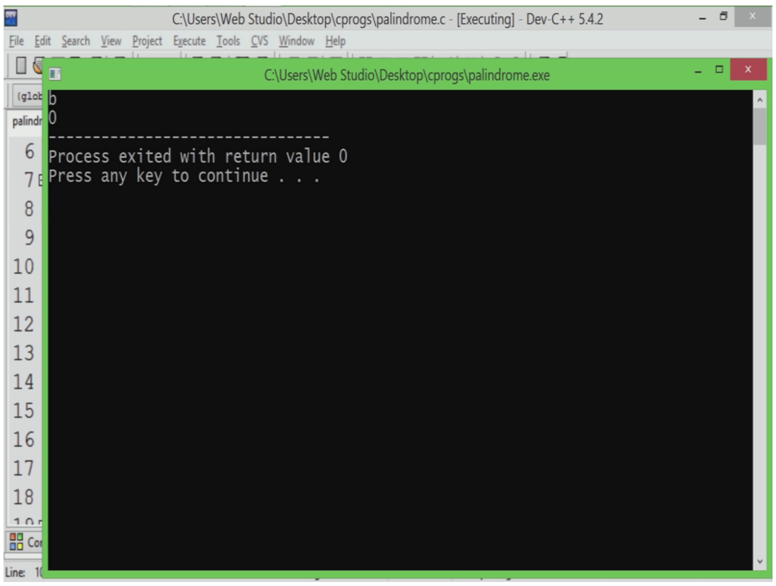
(Refer Slide Time: 20:43)



```
malayAlam
mal
--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

And the other input that was given was Malayalam capital. So, let we see what it is. So, there is m a l a y capital A. So, I am going to give that as an input, A, and it prints till m a l. So, now, it seems to have taken care of 4 cases. Let me check this test case 2.

(Refer Slide Time: 21:08)

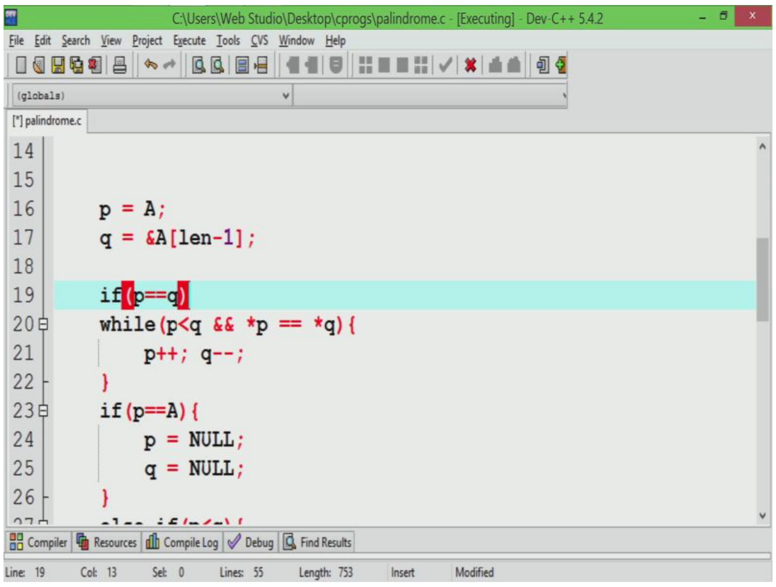


```
b
0
--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

So, I will run that now. I never talked about it so far. I put b; it is a single character and it gives me 0. There is a small problem. So, let us go and see what this problem is. So, if it is a single character it gives me 0. So, b is a single character and it is a valid palindrome, right. Let us now run through the code and see what happens. We start with p equals A which means p will point to b, the letter b; q is ampersand A of length minus 1; so q will also point to the letter b. And while p is less than q and star p equals to star q, this condition would have failed because both p and q are the same, right. So, because p is pointing to the letter b, q is also pointing to the letter b, and this would have failed.

Now, we have this check; if p equals A, in this case p equals A, we put p equal to null and q equal to null. So, this is a problem because we started with p equal to null and q equal to null which means it will always declare all the single letter things as not a palindrome. So, this is a very special corner case that has to be handled. So, it is quite likely that the if we submitted a program like this, it had passed the 3 or 4 test cases that are there, but not this test case for the single character. So, I am going to deal with the single character as a special case.

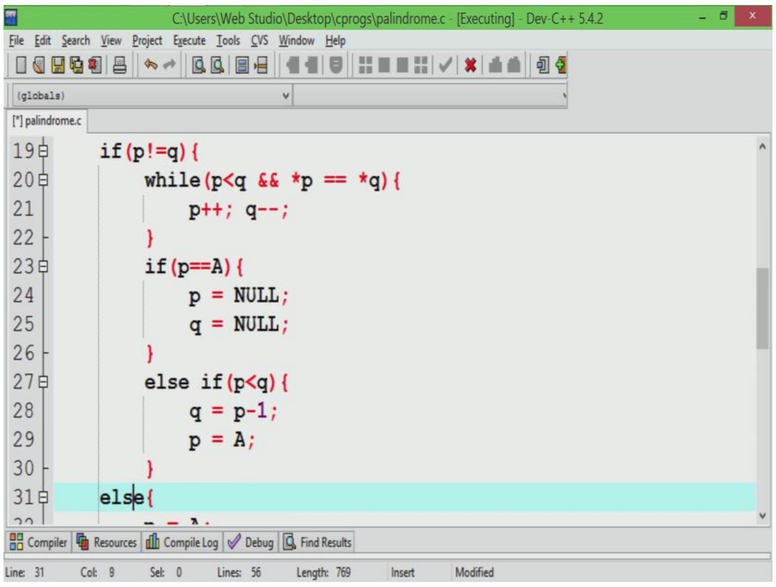
(Refer Slide Time: 22:34)


C:\Users\Web Studio\Desktop\cprogs\palindrome.c - [Executing] - Dev-C++ 5.4.2
[*] palindrome.c

```
    p = A;
    q = &A[len-1];

    if(p==q)
    while(p<q && *p == *q){
        p++; q--;
    }
    if(p==A){
        p = NULL;
        q = NULL;
    }
```

So, if p equals q, right; what that means is, sorry; let me check, if p equals q what does that mean? It means p and q are pointing to the same location; we have not even done any processing so far. If p and q are pointing to the same location it means we have a problem. And what is that problem? The problem is that this will be declared as a not a palindrome.

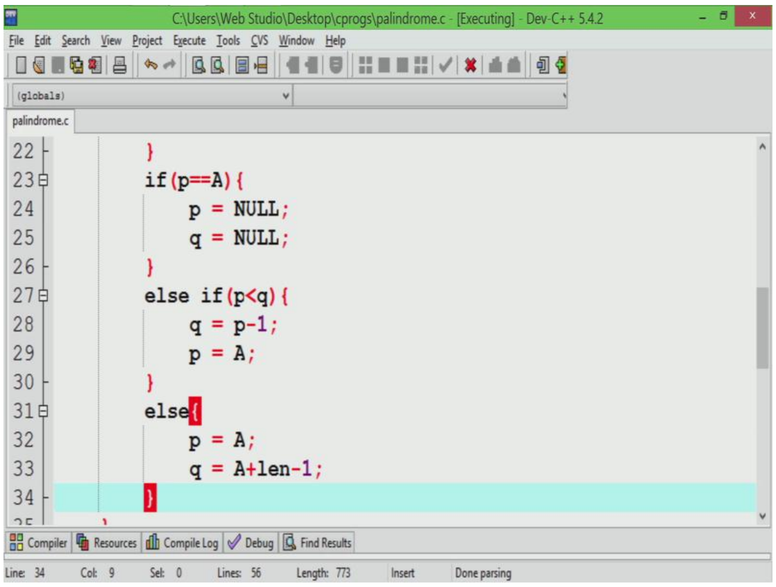
(Refer Slide Time: 23:00)


C:\Users\Web Studio\Desktop\cprogs\palindrome.c - [Executing] - Dev-C++ 5.4.2
[*] palindrome.c

```
    if(p!=q){
        while(p<q && *p == *q){
            p++; q--;
        }
        if(p==A){
            p = NULL;
            q = NULL;
        }
        else if(p<q){
            q = p-1;
            p = A;
        }
    else{
```

So, if p is not equal to q, it means the input string is at least 2 characters, only then I will do all this work. I am going to do all this work only if p is not equal to q. However, if p is equal to q, p is pointing to the first location, q is pointing to the last location, so print characters will automatically take care of it. So, just to be clean I will change the indentation, so that you know what is happening.

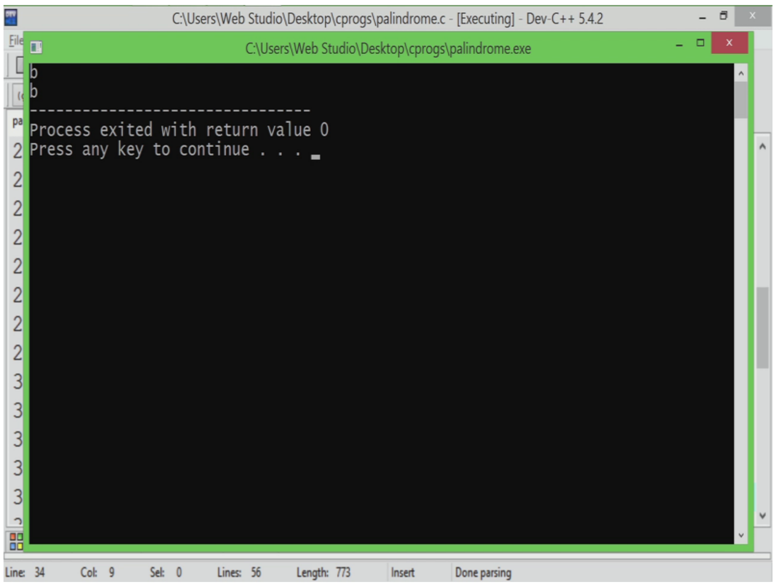
(Refer Slide Time: 23:48)


C:\Users\Web Studio\Desktop\cprogs\palindrome.c - [Executing] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
(globals)
palindrome.c
Compiler Resources Compile Log Debug Find Results
Line: 34 Col: 9 Sel: 0 Lines: 56 Length: 773 Insert Done parsing

```
        }
        if(p==A){
            p = NULL;
            q = NULL;
        }
        else if(p<q){
            q = p-1;
            p = A;
        }
        else{
            p = A;
            q = A+len-1;
        }
```

So, all this work will happen only if you have a palindrome which is greater than 1 in size.

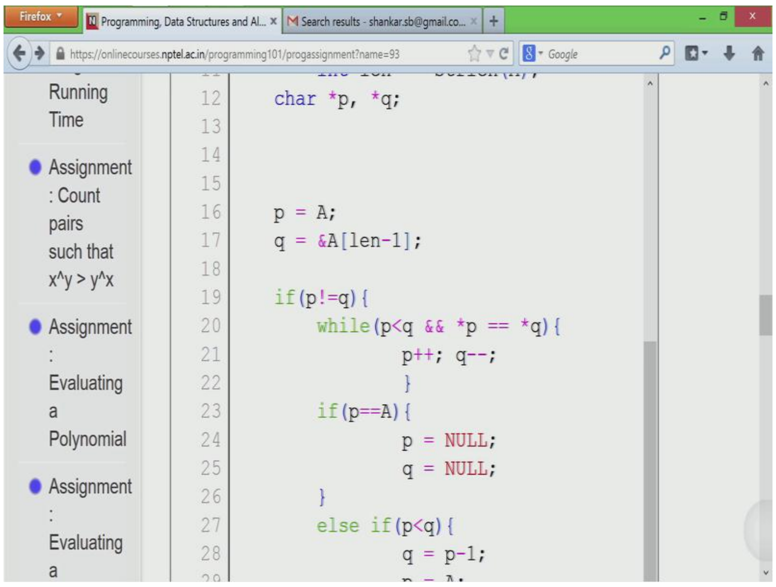
(Refer Slide Time: 23:55)


C:\Users\Web Studio\Desktop\cprogs\palindrome.c - [Executing] - Dev-C++ 5.4.2
C:\Users\Web Studio\Desktop\cprogs\palindrome.exe
Line: 34 Col: 9 Sel: 0 Lines: 56 Length: 773 Insert Done parsing

```
b
b
--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

So, let me now save it, compile it, and run it. Now, if I do the input b, it is declared as a palindrome and you get this.

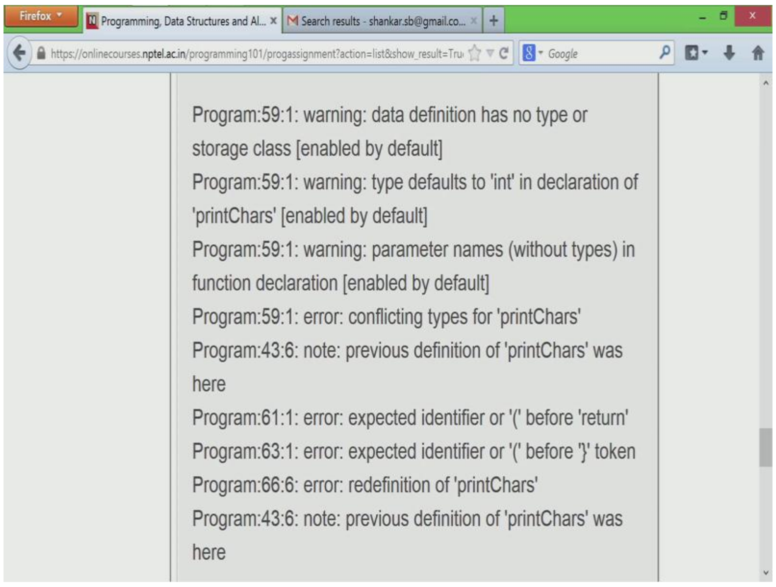
(Refer Slide Time: 24:05)


Running Time
Assignment : Count pairs such that x^y > y^x
Assignment : Evaluating a Polynomial
Assignment : Evaluating a

```
    char *p, *q;



    p = A;
    q = &A[len-1];

    if(p!=q){
        while(p<q && *p == *q){
                p++; q--;
                }
        if(p==A){
                p = NULL;
                q = NULL;
        }
        else if(p<q){
                q = p-1;
```

So, I am going to take this code, put it online, and see if I pass the test cases. So, I will save and compile and run.

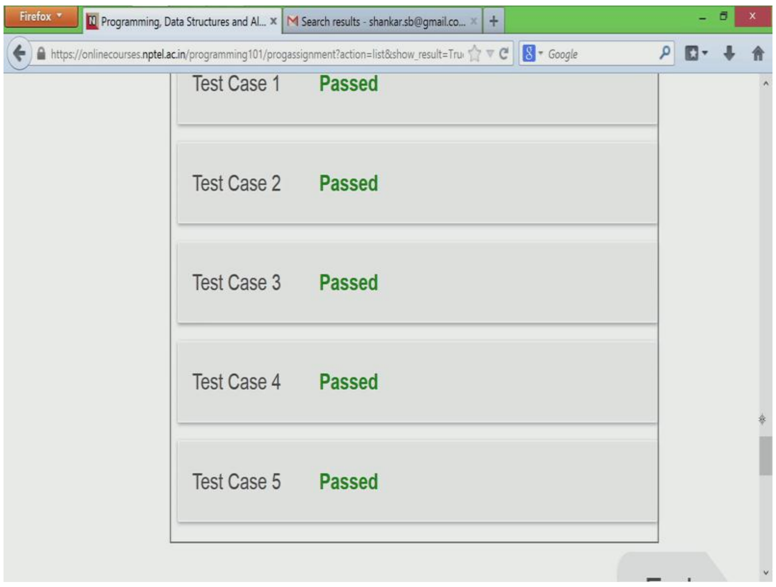
(Refer Slide Time: 24:19)



```
Program:59:1: warning: data definition has no type or
storage class [enabled by default]
Program:59:1: warning: type defaults to 'int' in declaration of
'printChars' [enabled by default]
Program:59:1: warning: parameter names (without types) in
function declaration [enabled by default]
Program:59:1: error: conflicting types for 'printChars'
Program:43:6: note: previous definition of 'printChars' was
here
Program:61:1: error: expected identifier or '(' before 'return'
Program:63:1: error: expected identifier or '(' before '}' token
Program:66:6: error: redefinition of 'printChars'
Program:43:6: note: previous definition of 'printChars' was
here
```

So, I get some warning saying data definition has no type in line number 59. So, all these are supposed to be commented. So, I am not supposed to add any of that; I will remove all of that. So, remember, it supposed to end with here, and the rest of code is automatically added. So, I will save and compile and run once more. So, that is a mistake that you might have done. So, I deleted all the things that where beyond it.

(Refer Slide Time: 24:58)


Firefox
Programming, Data Structures and Al...
Search results - shankar.sb@gmail.co...
https://onlinecourses.nptel.ac.in/programming101/progassignment?action=list&show_result=Tru
Google

Test Case 1 Passed

Test Case 2 Passed

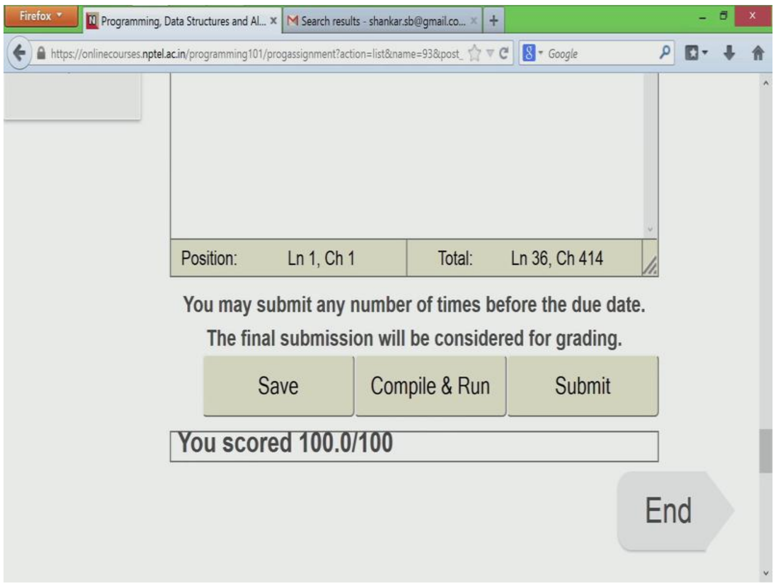
Test Case 3 Passed

Test Case 4 Passed

Test Case 5 Passed

And now, that code got automatically appended, and I seem to have passed all the public test cases. Let me submit this and see if I pass the private test cases also.

(Refer Slide Time: 25:12)


Firefox
Programming, Data Structures and Al...
Search results - shankar.sb@gmail.co...
https://onlinecourses.nptel.ac.in/programming101/progassignment?action=list&name=93&post_
Google
Save
Compile & Run
Submit
End

Position: Ln 1, Ch 1 Total: Ln 36, Ch 414

You may submit any number of times before the due date.
The final submission will be considered for grading.

You scored 100.0/100

So, I seem to get 100 on 100. So, if you go and put the code back you will see that there are issues with it because again there is a piece of code which is the same thing that gets appended to it. For running in the id, I had to uncomment and use it. But, when you put it back in the webs, in the portal, you should ensure that you delete everything that was not supposed to be there. So, the code will look like it is truncated somewhere in the middle, but that is ok, because there is automatically some piece of code that gets added up.

So, I just wanted to show you this to show a few features – 1, we saw how the string operations worked and we used plus plus and minus minus, and A plus length minus 1 and so on. These were operations on the strings; that is one thing that I wanted you to see. The other thing that I wanted you to see is if you putin code from your ide back to the portal, without deleting appropriate things, you will have a problem. So, I deleted those and things were ok now.

And the third thing is, when you do corner cases we have to be careful; we have to really thing through what is happening. So, many times we get these request on the forum that oh all my test cases pass, something is failing, right. So, in this case if I did not check for the test case, the single letter b it may look like everything is ok. We could have worried that inside a private test case and you would have assumed that a public test case has passed. So, these are the 3 things that I wanted you to show. I am hoping that you are able to catch some of these mistakes, just like I did when I started programming these things.

And I am also hopping that you are enjoying the course so far. So, one thing about the certification is we keep seeing requests that oh can we extended deadlines and so on because certification is there. So, one small point that I want to make is do not worry about the certification now; even if you missed 1 or 2 assignments, or 1 or 2 problems in an assignment, keep doing it. So, it is important for you to take as many as you can and finish them, and we will worry about the cutoff and other things later.

So, thank you very much and see you next time, bye.