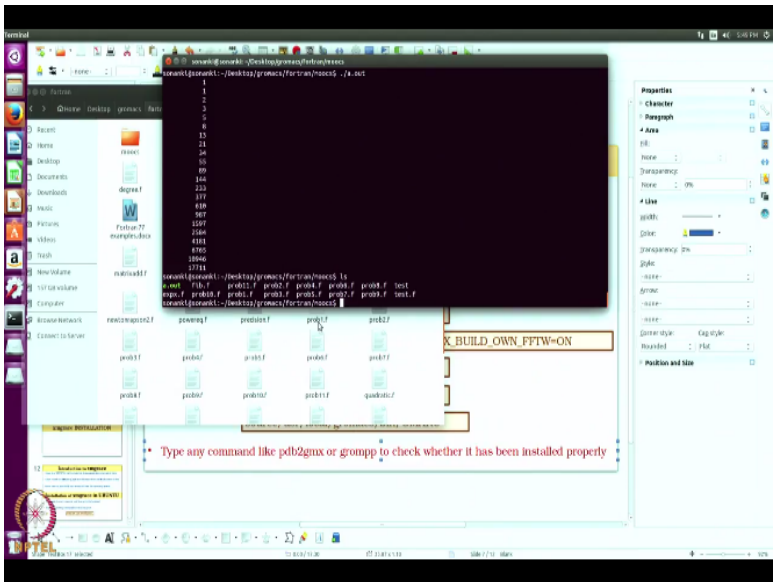
Computational Chemistry & Classical Molecular Dynamics
Prof. B. L. Tembe
Department of Chemistry
Indian Institute of Technology – Bombay

Lecture - 07
Practical Session on Programming - 1

Hello. So far what we have done is that we have discussed the main aim of our course. The main aim is learn some fortran programming and apply it to several problems in chemistry, then go to some public domain software such as Scilab and Gromacs and learn to do classical molecular dynamics calculations. So we have been using a Linux operating system for this, so today will be the first practical session on programming.

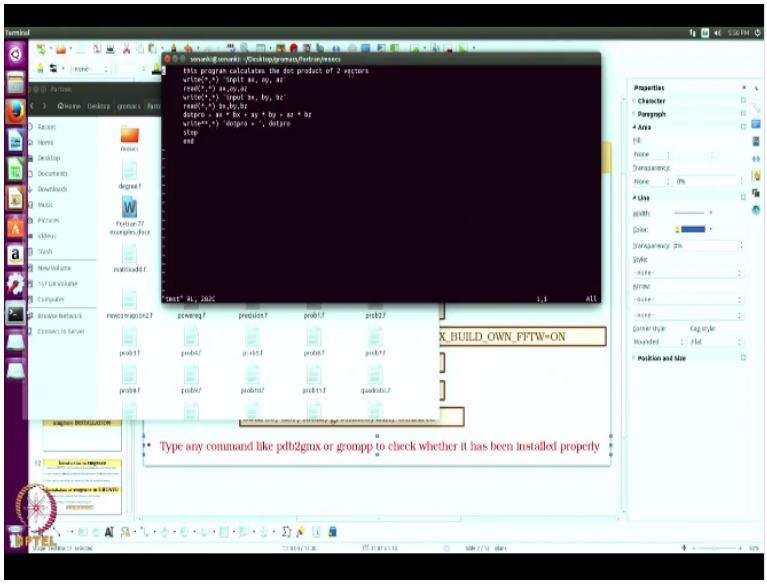
What I have done, I have created a directory MOOCS under directory fortran.

(Refer Slide Time: 00:56)



So my path you will see on the screen my path is desktop Gromacs fortran MOOCS. Everything that I have is in the MOOCS directory. So what I will try to do, I will try to execute a very simple program and then go along with this step by step to other programs. So the first program I want to execute is called a test program. So what this test program does is to calculate the dot product of two vectors.

(Refer Slide Time: 01:28)



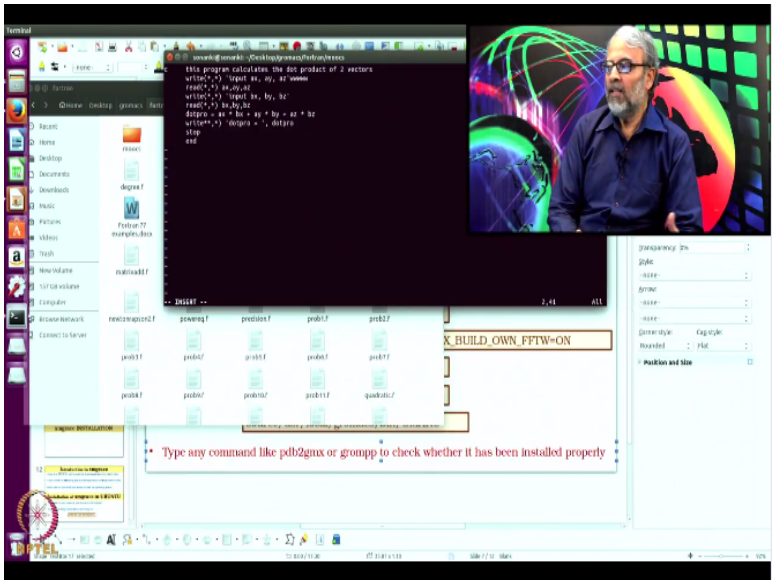
So I will use a vi editor, vi test okay, so there is nothing here because I typed vi text, actually it was vi test. So I come out, so to come out okay so I have come back to my original directory. So instead of test I have typed text. So let me type vi see my command vi test okay, so it shows my program. So look at the program very carefully okay. The first line is a comment card.

So the first line says this program calculates the dot product of two vectors. So this is a simple program to calculate dot product of two vectors. So when I want to calculate the dot product of two vectors, first I have to input those vectors. So I will call it vector a and vector b and the components are ax, ay, az for vector a and bx, by, bz for vector b. So my first line would be write star, star)'input ax ay az.

Now remember instead of input I have made it inpit, so what I will do with the cursor I go back to that i, so the command to replace i by u is type r, in Linux if you type r and type u that i has become u. So to replace a character type press r and type u in that place. Now suppose you want to insert a line, so now there are two modes when you do vi, one is called the edit mode.

In the edit mode, you can use the cursor to go anywhere and change things and one is called the input mode. To have an input mode, let me go to this.

(Refer Slide Time: 03:33)



So since I have done up to the first line, I go to the end of this first line and then type i here on my screen, when I type i it allows me to insert. So in the insert mode, I was at the location of the colon, I go to the right and let me say www and typing something at random. Now I know that some unnecessary things have been typed, I want to delete it, so to delete I type escape then I start deleting, x means delete, x delete, delete, delete, delete.

So I am in the edit mode. In the edit mode, I can move and change things. In the insert mode, I want to insert things. To again to insert type i okay then move to the right type anything you want. To delete whatever, I was in go to escape, now I am in the edit mode just type okay xxxx, it will delete all these things okay. So with some practice you will know how to go to different places, how to insert and how to edit.

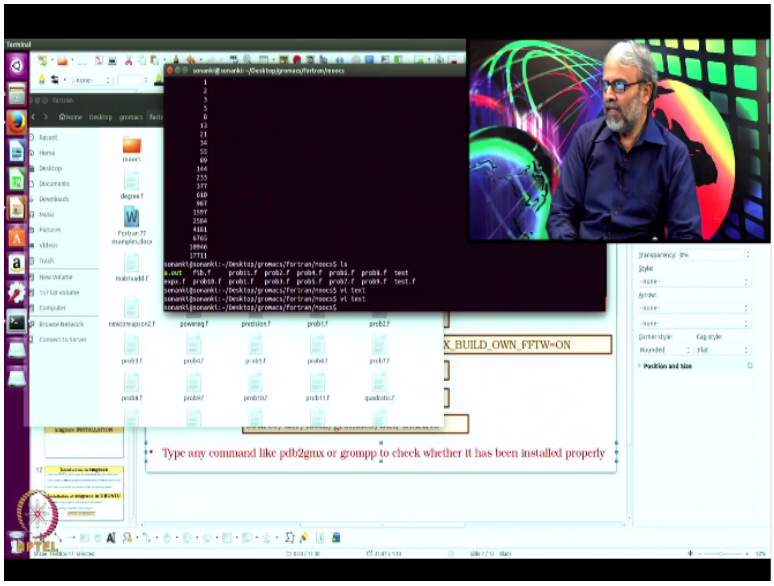
These are two modes, when you type i in the edit mode you go to insert and when type escape in the insert mode you go to the edit mode. So by practice you will get this. So coming back to my program, my first line is it says write star, star. As we said star, star means write on the screen. In ? It says input ax ay az. What this line does for me is it writes from the screen that statement input ax ay az.

So go to the next line, next line says read star, star ax ay az. So once you see on the screen input something and the next line is read, so it says read ax ay az. Through this particular statement, you will read from the screen the values of ax ay az, 3 numbers. So the next line is okay write input bx by bz. After you input ax ay az you will see on the screen write you will see on the screen input bx by bz.

So it prompts you to type the values of bx by bz. The next line is read star, star bx by bz. This will read these 3 values from the screen okay. So once you have read the 6 values that is the 3 component of vector a, 3 component of vector b, your next task is to take the dot product. See the next line the next line says dot product=ax*bx+ay*by+az*bz, so it takes the sum of ax*bx ay*by and az*bz.

So now that is your dot product. So the next line you want to write the dot product on the screen. So you see that there is a small error here in the write, so it is instead of write into bracket it says write star, star.

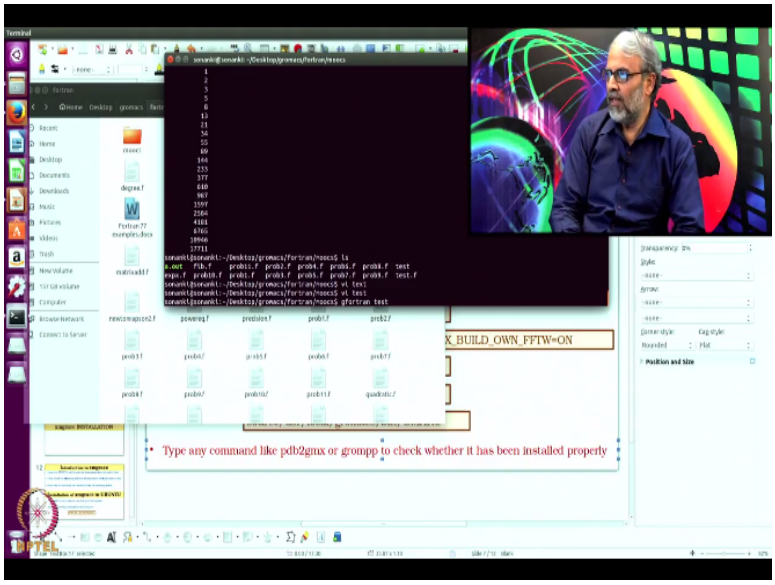
(Refer Slide Time: 06:38)



So I would replace one star by bracket, so that is again r and I type the left parenthesis. So now this my program is right. So I want to come out of this program I type shift colon, so when I type shift colon that colon appears at the bottom okay. So that is the place where you can put your command. So I have entered my program and again remember the first line is a comment card, first line is a comment card.

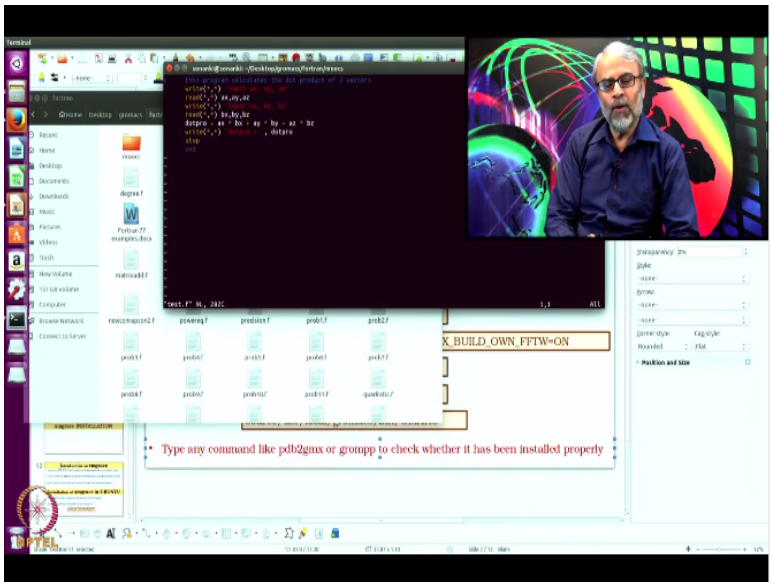
So I come to this cursor shift colon I typed w or x, x means exit after saving. So what it has done? It has changed the program the way we change it and it is ready. Now I want to compile it.

(Refer Slide Time: 07:33)



So to compile what was the command gfortran. The program name was test, so I will give test okay.

(Refer Slide Time: 07:45)



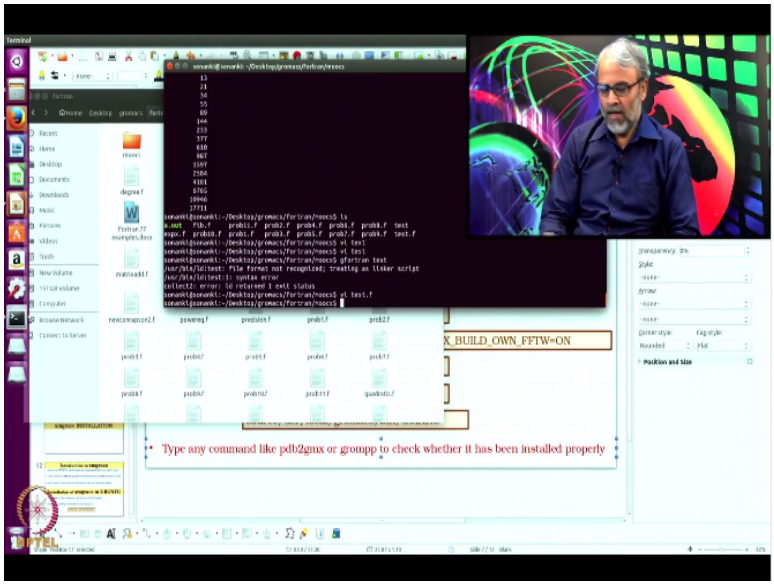
I have said gfortran test and I enter now it says all kinds of error it has given, so this was my command gfortran test, now it says user bin test file format not recognized treating as linker script, all kinds of things, syntax error okay, it exited. So now what was the problem? As far as I was concerned the program was completely correct. So the problem was the program that we wrote did not have an extension of dot f.

Remember whenever you want to execute the fortran and command gfortran, the file should be in the dot f format okay. So I purposely wrote it as a test and try to compile because I know that it will not work. So in the future whenever you use gfortran, the file that you use

should be test.f. So I have already created the test.f. Let me edit that vi test.f. So it is the same program that we wrote earlier except the name is test.f. See the last line here.

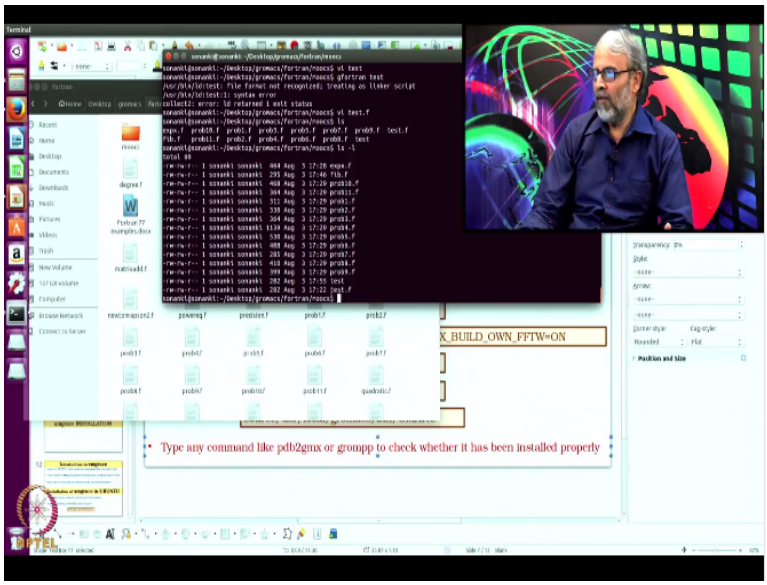
It says the program is test.f okay, 9 lines, 282 characters. Look at the top 1, 2, 3, 4, 5, 6, 7, 8, 9 there are 9 lines. So in this case it is not just test but test.f. Now I will come out. How do I come out of it? Shift colon.

(Refer Slide Time: 09:09)



I will type x, I come out. So now let me do ls, ls command in Linux is list of directories. So these are all my programs.

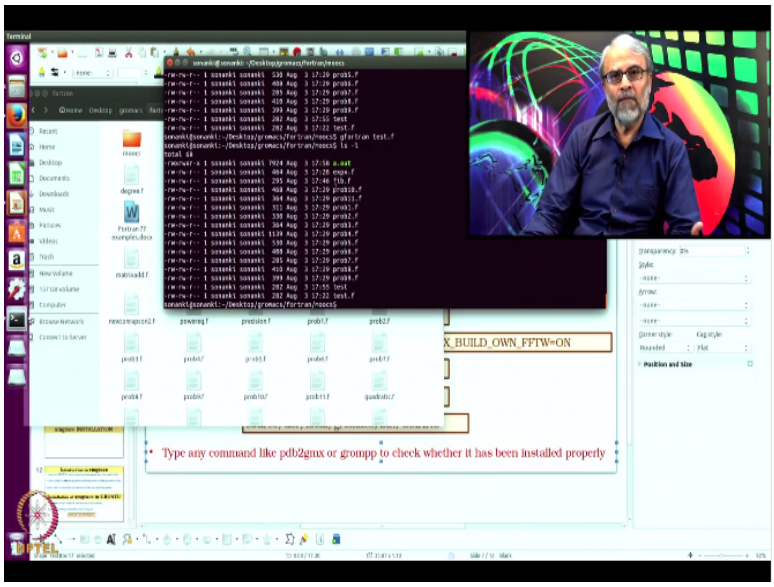
(Refer Slide Time: 09:21)



So since it is very cluttered, I will type ls-l. So ls-l gives me a longer list, so these are all my programs. See from top to bottom these are my programs. You will see that there is a test and

there is a test.f. Actually, test is same as test.f but since in fortran I need test.f. I have made a copy of test*test.f, now I am ready to execute. So let me execute.

(Refer Slide Time: 09:45)

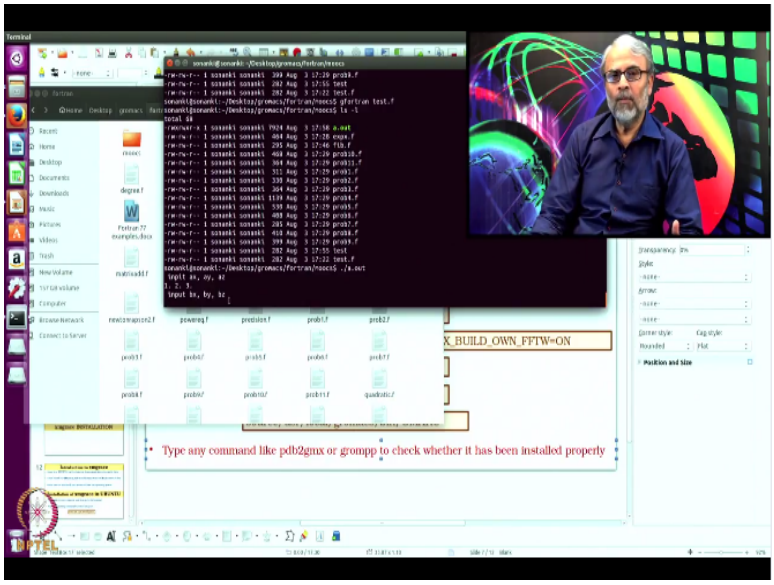


What is the command? gfortran test.f, when I do that let us see what happens. So this time, it gave no error, it has executed okay. So how do I know it has executed. Let me do ls-l okay, you see at the top, it has created this file called a.out. This a.out is an executable file. So in Linux there are different types of files, there are normal program files, executable files. In all the executable files, look at these first 9 lines okay.

The first 9 lines in the list tell you about the nature of the file okay, read, write, execute, read write execute, read write, execute. So these are 3 situations okay. The last x says that it is in execute and in Linux there are different kind of permissions you give, one is the user, then there are other users and so on. So this read, write, x, read write, x, read write x differ to the present user, other users and the overall system okay.

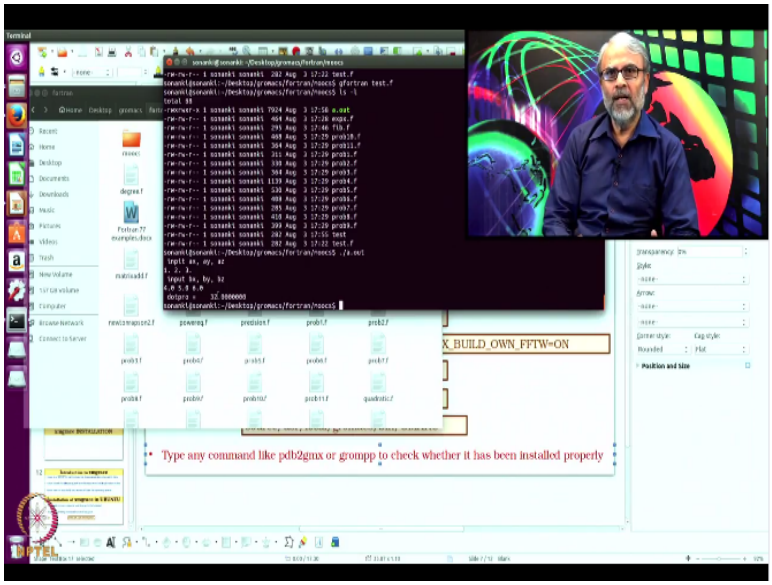
So we will not worry too much about it. The main reason for showing is that this a.out is an executable file. So we will see that that x remaining at the last line out of these 9 things okay. So now I will execute the program.

(Refer Slide Time: 11:12)



To execute the program, ./a.out okay, so what does it ask me inpit ax ay az. I had changed inpit to input in the test file but not in the test.f but that does not matter because it is a syntax error. Our job is to type ax ay az on the screen. I will type 3 numbers, one point space, two point space, three point. So these are my data. The vector has 3 components, 1, 2, 3 enter. So when it enters, the next thing written on the screen was bx by bz.

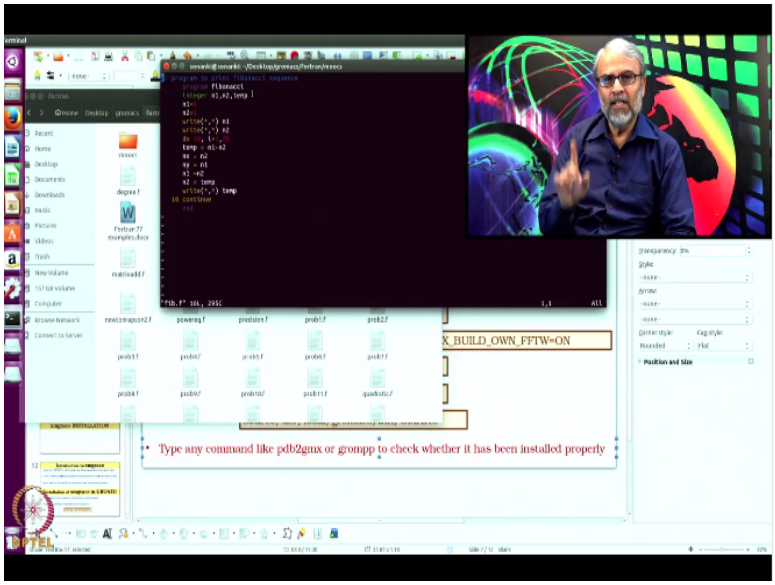
(Refer Slide Time: 11:56)



The program tells me to input the values of the vector b bx by bz, so let us give it 4.0, 5.0 and 6.0. These are my component of the second vector okay. When I enter, so it has set the dot product is 32. So what is 32? 1*4 that is 4, 2*5 that is 10 okay, 4+10 is 14, 3*6 is 18 okay, 14+18 is 32, so that is the value of my dot product. So this was the execution of your first program.

So you got the value of your dot product. If you want to practice, give other values and see what the value of dot product is. Next I will go to another program. This is called a Fibonacci series okay.

(Refer Slide Time: 12:38)



Let us see the program vi Fib.f, that is the name of my file, Fib.f is the file, I have already entered this line statements in the program. So let us view what that Fib.f is. Fib.f is a program that calculates Fibonacci sequence. Fibonacci sequence is a sequence of numbers okay. The numbers are 1, 1, 2, 3, 5, 8 and so on. Each number is the sum of two previous numbers. The first two numbers are 1 and 1.

So the third number is 1+1 that is 2, the next number is 1+2 that is 3, the next number is 2+3 that is 5, the next number is 3+5 that is 8, it adds the previous numbers. It is a sequence each term is a sequence is a sum of two previous numbers and the two starting numbers and 1 and 1. So that is what this program is doing. So let us see this program from the beginning. So the first line is a comment card program to print Fibonacci sequence okay.

Then, again next line program Fibonacci, this is not a compulsory thing, but not a problem, we have given the name Fibonacci. The third line this is the declaration statement, remember so far what we have done, we have only considered executable statements and do statements. So these are all executable statements but the statement integer n1, n2 temp, so it declares that the variables n1, n2, temp are integers.

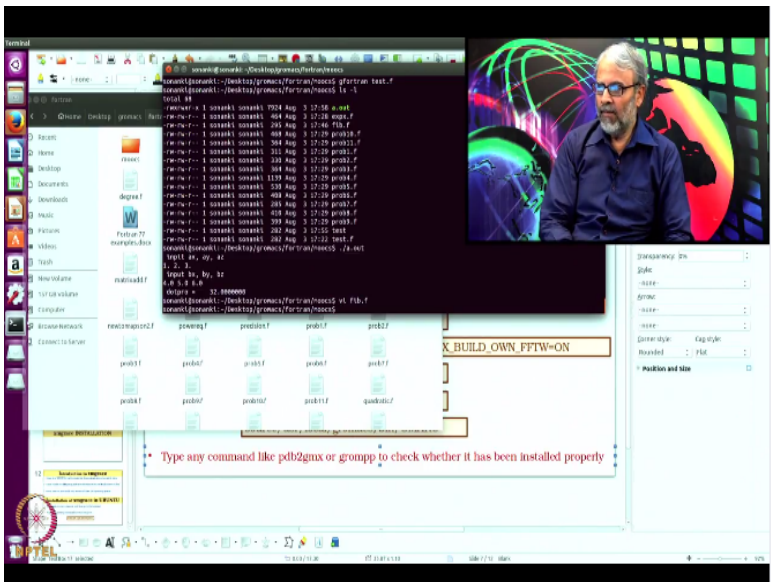
So the beginning of a program always has a set of all the declaration statements. So this declares that these are all integers, n1, n2 there is no need but temp you have to declare because we said earlier that all variables that begin with i, j, k, l, m, n are by default integers. So if I want to make temp as integer, I have declared it as an integer. So declaration statement overwrites the default things in the fortran.

So I have done that. So in the program n1=1, n2=1, on the screen it will write n1, on the screen it will write n2 and now we start the program. It says do temp i going from 1 to 20. So what is the meaning of this do statement? 10 is the line number, see this is a line number here. So all the statements which are below this up to line 10, it will keep on doing it starting with i=1 up to i=20. So it is an iterative loop.

So what does the loop does? What does the loop do? It starts with the loop temp=n1+n2 that is my algorithm. The new number is the sum of two old numbers okay. So the first number is temp n1+n2. So next time what I have to do? Next time what I have to do is that the next number will be the sum of temp and n2. The first number as n1, second number of n2, the third number is temp, so the next number will be n2 and temp.

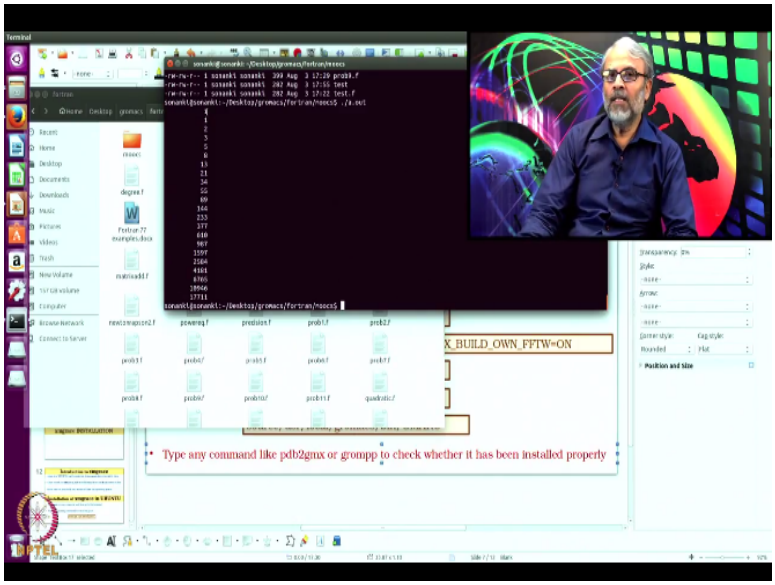
That is the sum, so to do that I am redefining n1=n2 that is n1 is the same as the earlier number. Then, n2=temp okay, so the next time when I add n1+n2 it will be n2+temp, so that will be the next number. So this is an iterative loop. Let us execute it okay.

(Refer Slide Time: 16:10)



So execute again shift colon and coming out, then I type q means quit, so last time we used x there because x means exit after making changes, q means quit without making changes. Now I am not going to make any changes in this program because it is already I had already tried it, so let us just execute okay. So I come out.

(Refer Slide Time: 16:37)



So how do I compile? gfortran, the name of the program was Fib.f okay. So it has already done the compilation. So let us do ls again, ls-l so you see that in this list, there is an a.out again. Earlier when we compiled the test.f program, we had an a.out, when I compile the Fib.f program again there is an a.out. So the present a.out is overwritten on the old a.out. So remember that a.out is always the executable file of the last thing that you have compiled.

So if you want to save the old a.out you can save it with another name. We will do that, so right now I will execute this Fibonacci sequence. What is the command for execution? Just ./a.out okay, so you will see this is my result. See what is the Fibonacci sequence, first number is 1, second number is 1, third number is a sum of the first two numbers. So that is 2, the next number is a sum of 2 and 1 that is 3, the next number is a sum of 3 and 2 that is 5, the next number is the sum of 5 and 3 that is 8.

Then, 5+8 13, 13+8 21, 21+13 is 34, so it is adding the previous two numbers to give you the next number. So this is called a recursion relation. What is the recursion relation? Recursion relation gives you a relation between the new number and all the old numbers. The next option would be suppose instead of a Fibonacci series, I want a sum of the previous 3 numbers.

So instead of 1+1=2, suppose my new series is 1+1+2 that is 4, so that is my new series now okay. So 1+1+2 is 4, the next number will be 1+2+4 that is 7. So you can have a series where you add the previous 3 numbers and not the previous 2 numbers. I will leave it as an exercise, one of our exercises today would be you take a series which is one step beyond the Fibonacci series.

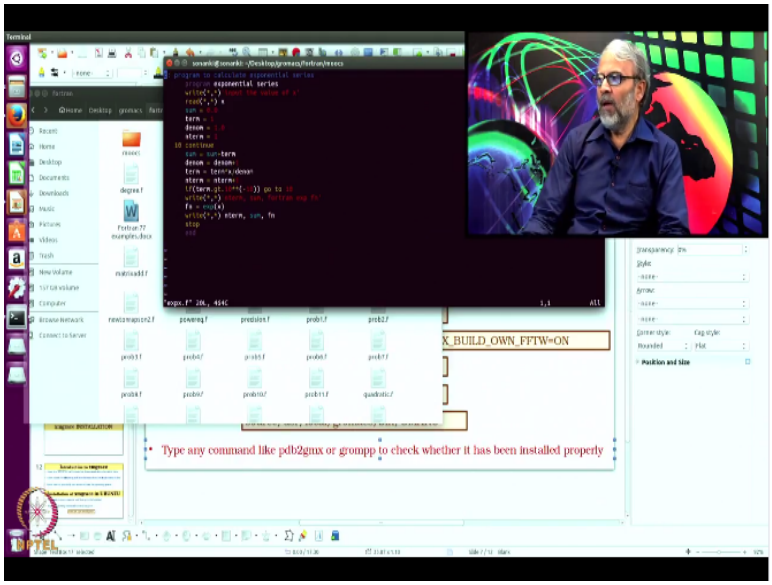
That is in this recursion relation you use the sum of 3 previous numbers, so take the present program and modify it to include the previous 3 previous numbers and do it as an assignment and your teaching associate will correct it for you. If you have any problems, you can get into touch with us. So, so far we have done two programs before we conclude today I will do one more program, third program that is calculation of an exponential of a variable x.

Remember on your calculator, you have this exp of x okay. So how do you calculate these exponential of x, so that is called a Taylor series. So the series is exponential of x=1+x+x square/2 factorial+x cube/3 factorial+x to the 5 x to the 4 by 4 factorial, all are added. It is the sum of x to the n by n factorial, n going from 0 to infinity. It is an infinite series, so what we will write in the program, we will calculate this exponential function.

Then, you are question will be how will I calculate these infinite set of numbers, it is not possible but fortunately once x is very large, x to the power n/n factorial becomes a very, very small number. This n factorial increases very, very rapidly. To make sure that you know why n factorial increases, on your calculator try to calculate the factorial of let us say 100 or factorial of 1000.

The calculator will not be able to calculate because the number is too big to be calculated. So therefore this exponential function is a convergent function, so my program will calculate that for you. Let us edit that function.

(Refer Slide Time: 20:37)



So let us see where is the location? So that program is called expx.f, so that calculates you exponential of the function. So to go to that program I say vi expx.f. So this is my program to calculate the exponential of a function. Let us slowly go through every step in this program okay. First line is a comment card okay, no issue, program calculates exponential series.

So this series is for an exponential function. Second line program exponential series, again the same thing, it is not an executable statement. Third one is the one which will start the execution, write star, star input the value of x. Remember, one of the most crucial element of the program is input value. So I have inout the value of x, it reads x from the screen, sum=0, term=0, denominator=1, number of term=1.

So this is my first term okay, denominator is also 1 okay. So up to line number 10, these are all statements were I am initializing these values. So then the next one, remember I have initiated this sum=0, whenever you do calculation always initialize your starting values, sum=0, so I after this line 10, sum=sum+term, term was 1, so I will add to this sum the value of term.

So now the sum is=1, so now I want to calculate the next terms. What do I do? Denominator=denominator+1 okay, earlier my denominator was 1, so next time denominator will be 2. Term is=term*x/denominator okay. So already the denominator is 2, so what I have done term was 1, the next term will be the value of the old term*x/denominator okay. So it will be x square/2 and 2 is the same as 2 factorial.

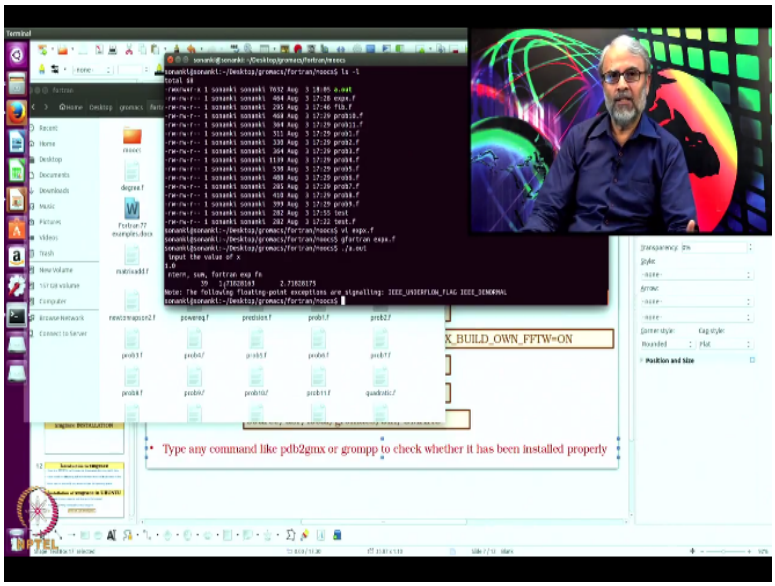
So the next term will be x square/2 and the number of terms I have increased by 1, so I will n term is=n term+1, so I have increased the number of terms by 1 okay. Then, next is the most important line. If the value of term which I have calculated okay is>10-10, go to 10. That means when I calculate a new term, if the term value of the term is large okay then I go back to my statement 10.

If it is not large then I write the value of n term sum fortran exponential function okay and conclude but before I conclude the fortran also has its own library of functions. So in the fortran expx is already calculating that exponential of the function. So I calculate this expx or fn is a result from the fortran compiler, what we have done our sum whichever sum I have calculated is our own program.

So we want to compare our calculation of sum and calculation by the fortran program okay. So the last line write n term sum fn, so coming back to the statement, if the term is>10-10, it will go here calculate. What will be the third term now? Second term is already x square/2 factorial. The third term will be x square/2 factorial*x/denominator which is 3 now, so it will be x cube/3 factorial.

The next term will be x 4/4 factorial, so what we have done again this is a recursion relation, the new value in terms of old value into some product. So again you are calculating the terms in a recursive fashion and adding and then we will execute. So what I will do, I will execute.

(Refer Slide Time: 24:40)



So to come out of this, escape colon x, I come out. Now I will compile it, gfortran okay xx expx.f, I will compile this okay, to execute again ./a.out okay. So now I have to input the value of x, what I will do, I will give just 1 as a value. So most of us know that e to the 1 is around 2.7 okay. So when I enter, so you will see is that you have 39 terms okay, the computer value has given the correct term whereas our program has given 1.7182.

So which means although the program executed properly, when I added 39 terms I did not get 2.78 but I got 1.7. So what we will do? We will go back and check why we did not get exactly the value that we are supposed to get and we will continue with this in the next class okay. So to summarize what we have done, we executed a very simple program of dot product of two vectors, there was no problem, you got the result.

First time, I did gfortran test and it did not compile because it was test and not test.f. So next time we did gfortran test.f, compilation it gave the correct result. Then, we did the Fibonacci series. Again, we compiled, executed, you got Fibonacci numbers. I gave you as an assignment, take the sum of the 3 previous numbers to get your new series. Then, we executed an exponential function program.

So when we executed, our result did not match the fortran compilers results. So next time we will start at this point and check why we did not get the correct result and resume at this point okay. So thank you very much. We will conclude here today.