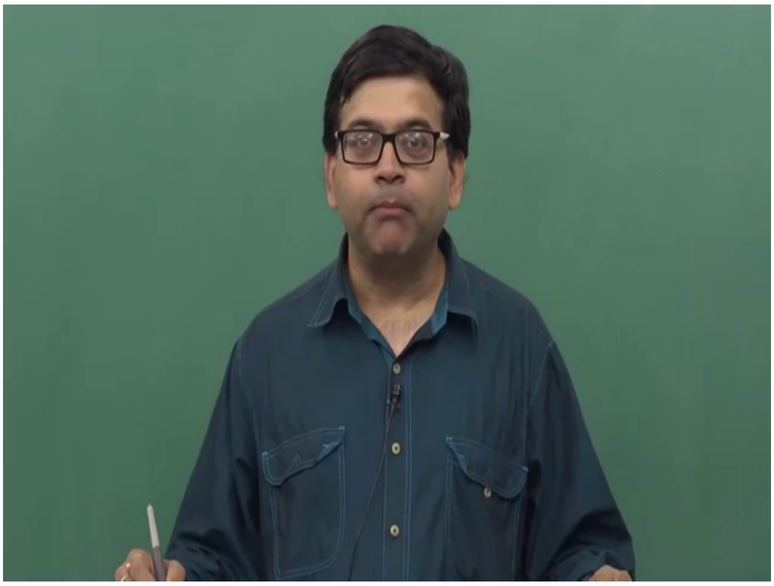
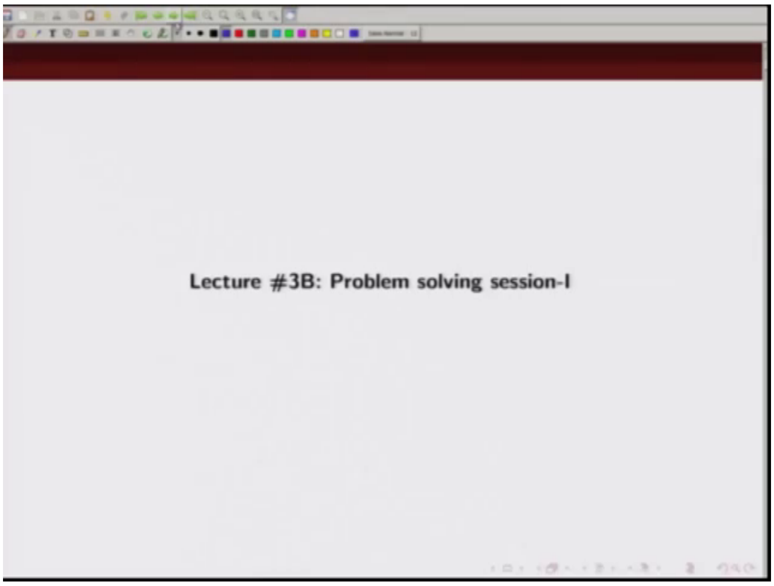
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 01
Lecture Number 06
Problem Solving Session-I

(Refer Slide Time 00:13)

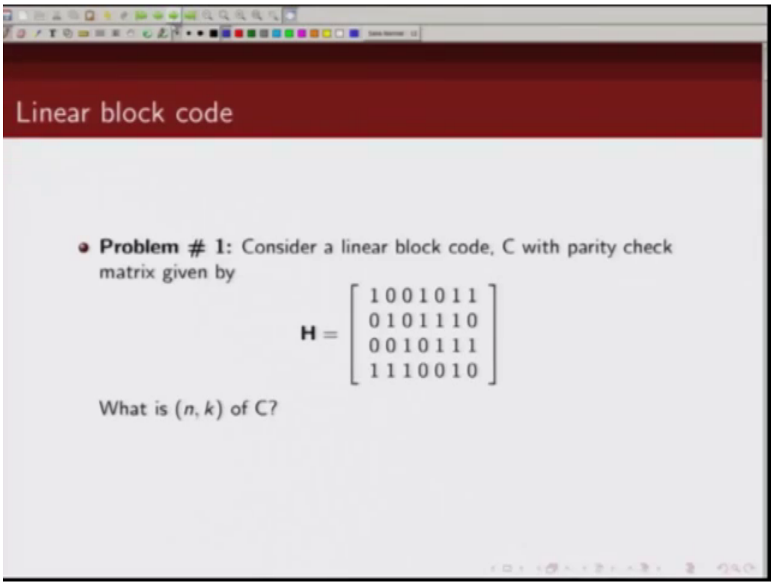


Before we discuss decoding of linear block codes let us solve some problems today.

(Refer Slide Time 00:19)

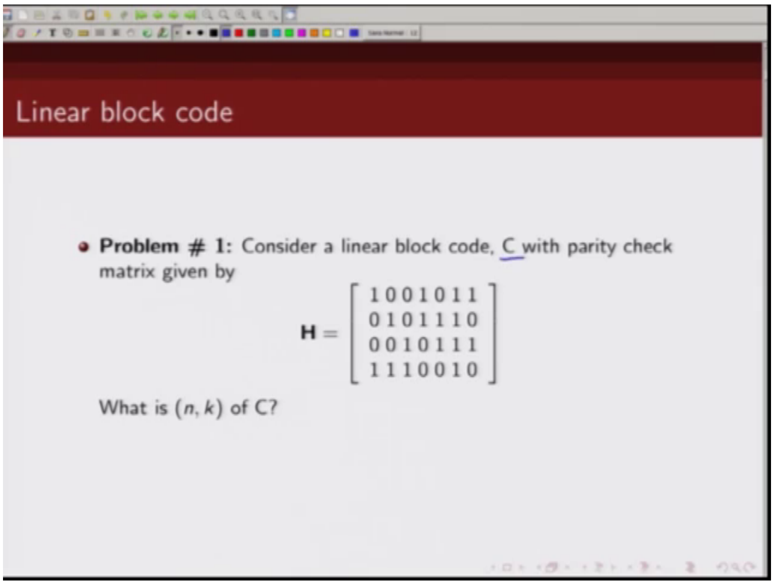


(Refer Slide Time 00:21)



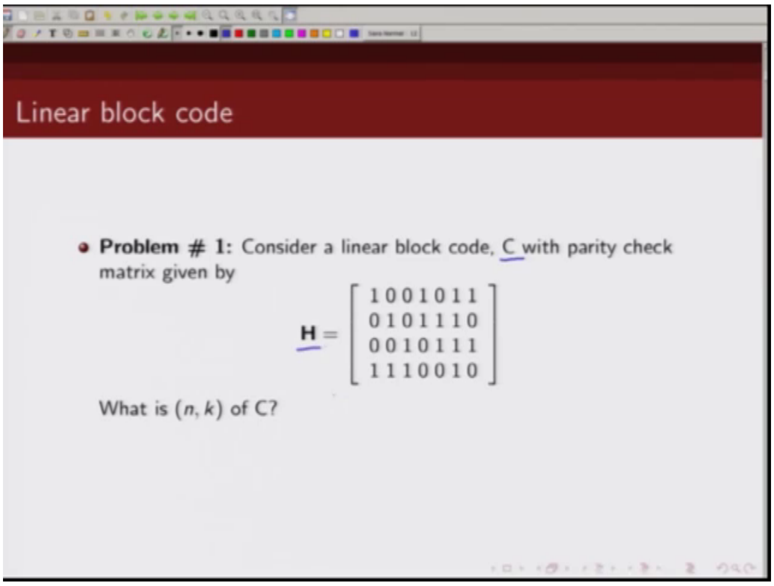
So first question we are going to look at is consider a linear block code C whose

(Refer Slide Time 00:28)

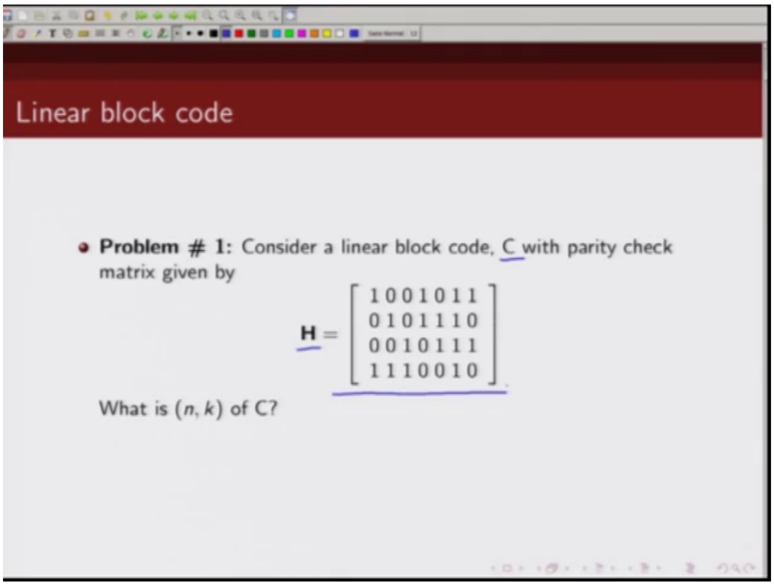


parity check matrix is given by this.

(Refer Slide Time 00:30)

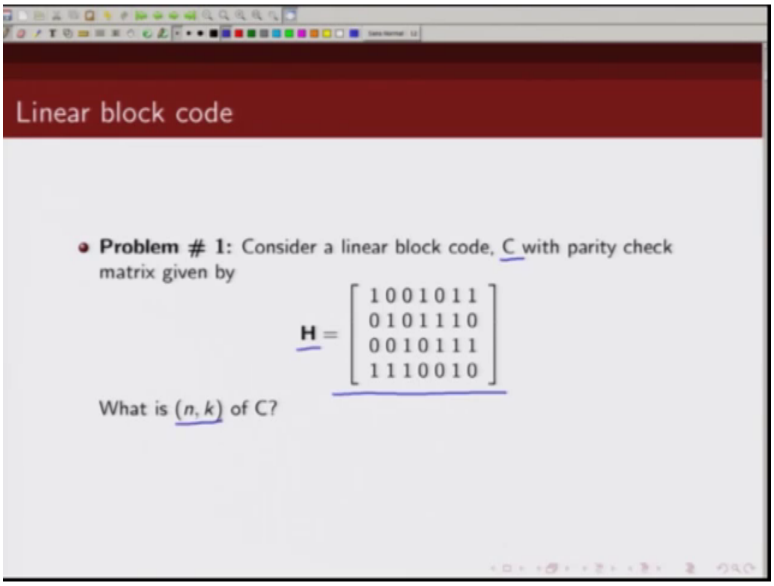


(Refer Slide Time 00:31)



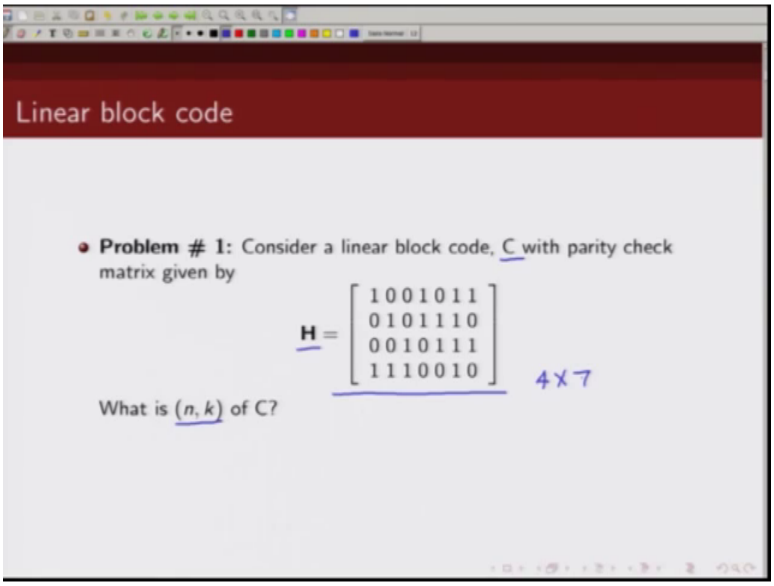
And you are asked what are the code parameters, n and k;

(Refer Slide Time 00:36)



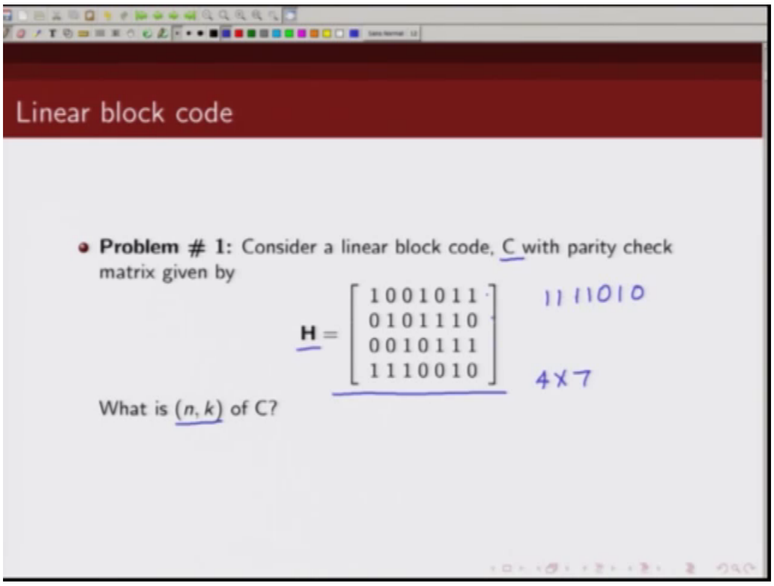
n which is a block length, codeword length and k is the size, it's the dimension of the, basically information sequence length s k. Now how do we solve it? We know, we will first find out what is the rank of this matrix H. Now you can see this is a 4 cross 7 matrix right? So the maximum rank

(Refer Slide Time 01:02)



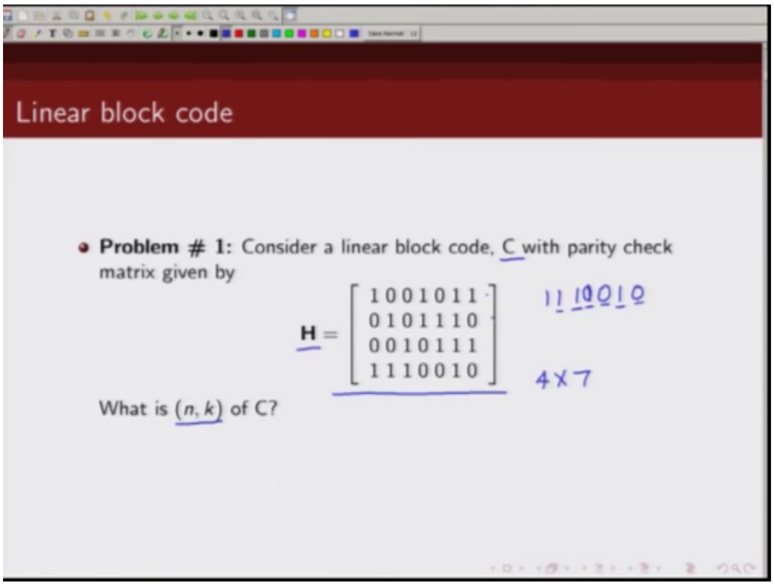
possible is 4. Let's see whether it has rank 4. Now if you add row 1, 2 and 3 what do you get? 1 1 1 1 0 1 0, sorry 1 1 1 0

(Refer Slide Time 01:21)



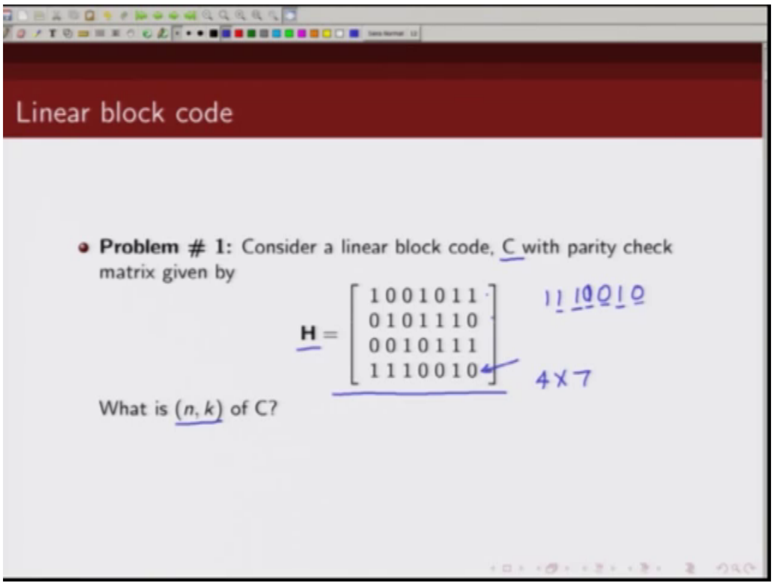
this is 0, 0 1 0. This is what you get, you can see this is 1, this is 1, this is 1, this is 0, this is 0 this is 1and this is 0.

(Refer Slide Time 01:40)



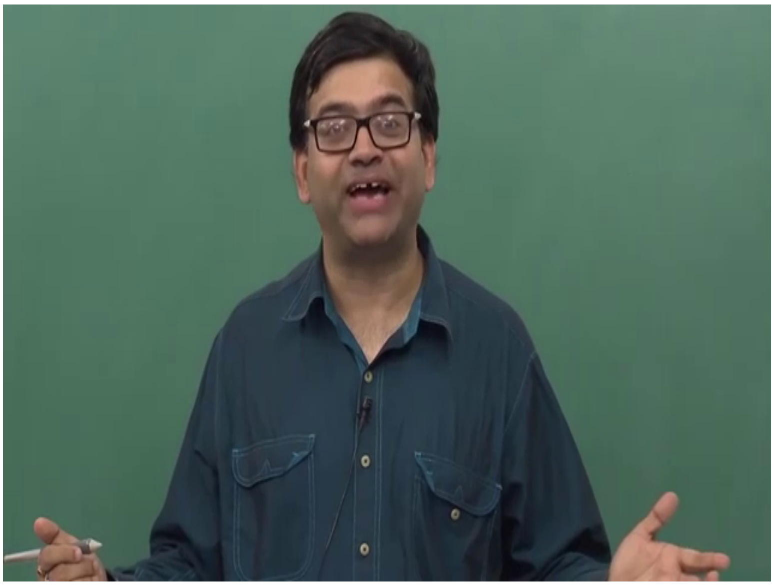
And what is row number 4? It's exactly same

(Refer Slide Time 01:45)



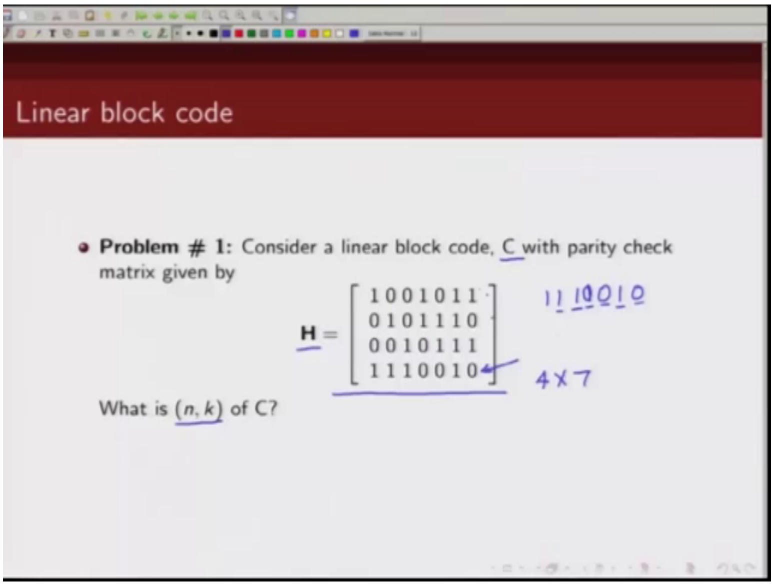
same as this. So you can see row 1, row 2, row 3 and row 4 add up to 0. That means it does not have

(Refer Slide Time 01:55)



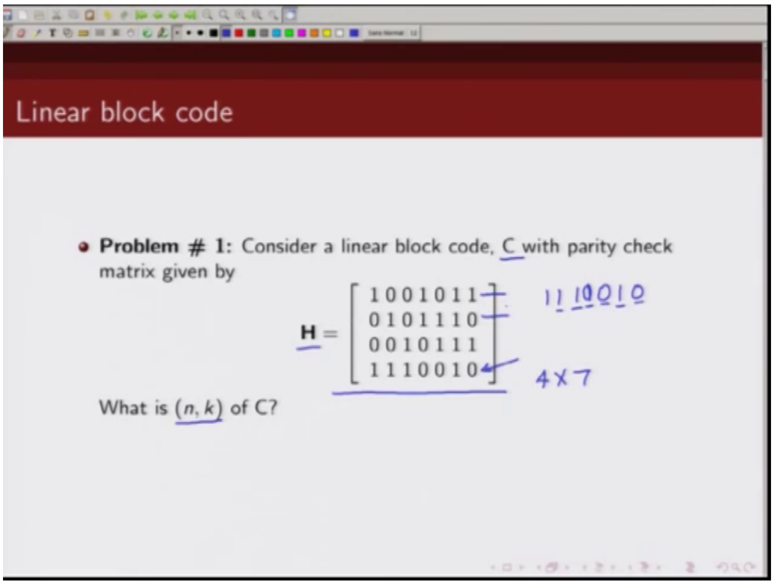
rank 4.So maximum rank possible is 3. So let's see

(Refer Slide Time 02:00)



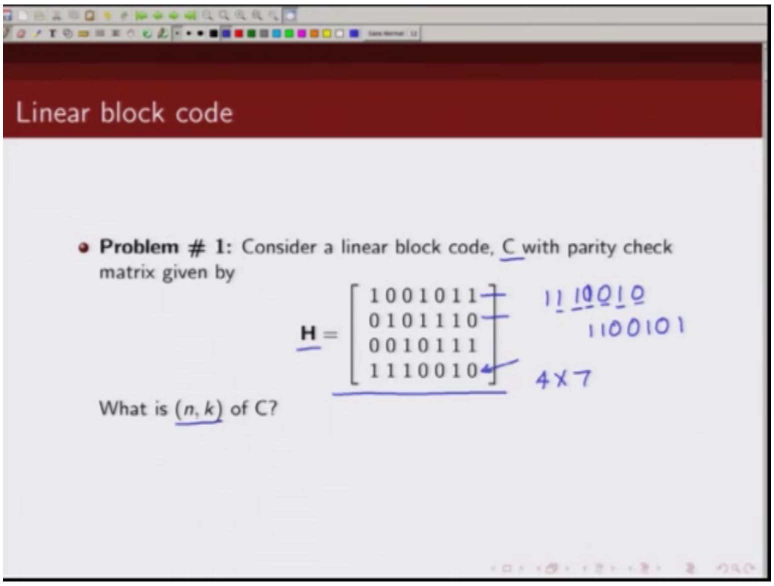
rank 4.So maximum rank possible is 3. So let's see

(Refer Slide Time 02:11)



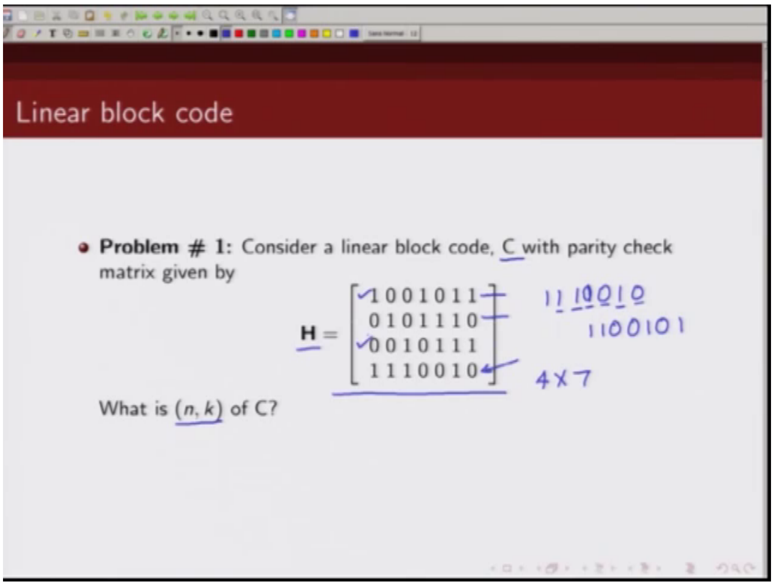
if any 3 rows combination add up to 0. So let's see, let's see if we can consider sum of these two rows. This is what, 1 1 0 0 1 0 1.

(Refer Slide Time 02:20)



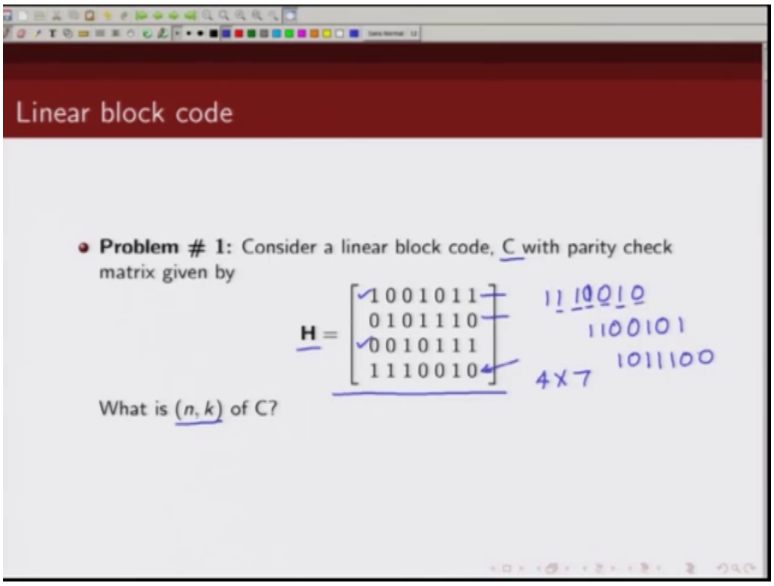
Now none of the rows are equal to this, you can see. If we consider this row and this row, we add these two rows.

(Refer Slide Time 02:28)



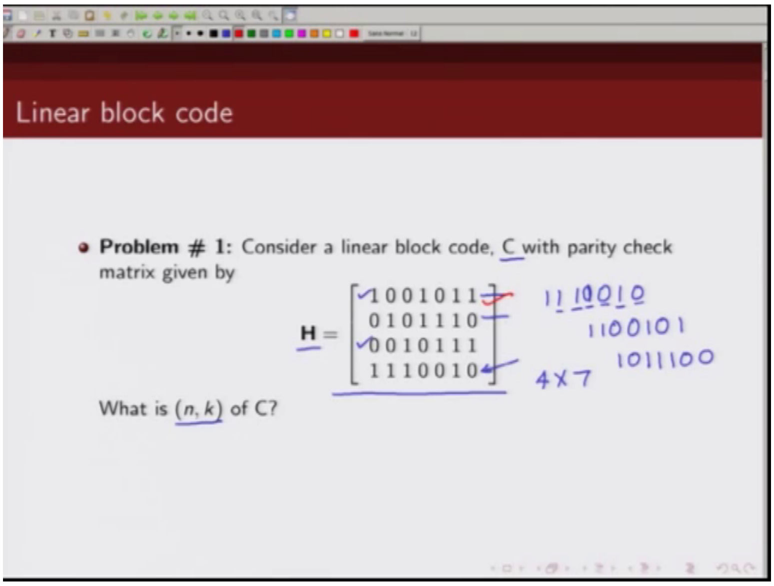
Let's see. What do we get is 1 0 1 1 1 0 0. Now note

(Refer Slide Time 02:41)



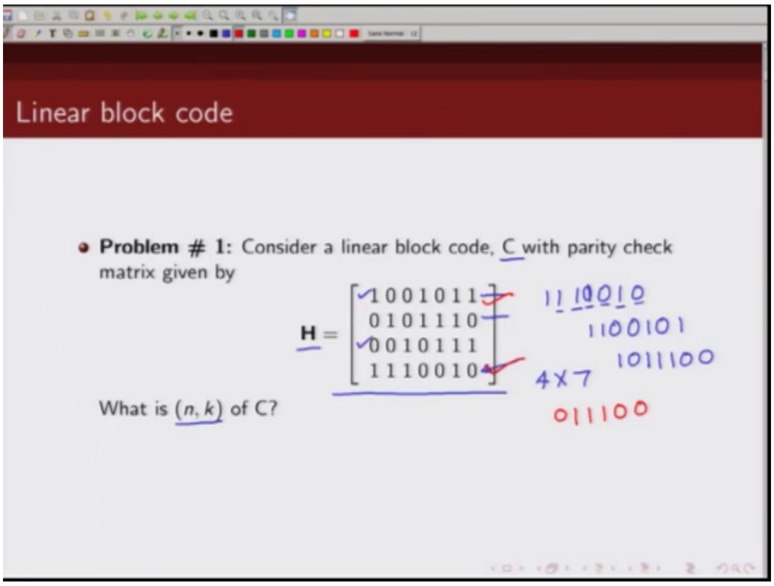
none of these rows, r 2 and r 4 is equal to this. So these set of 3 rows, basically they are independent. Let's try adding up this and

(Refer Slide Time 02:54)



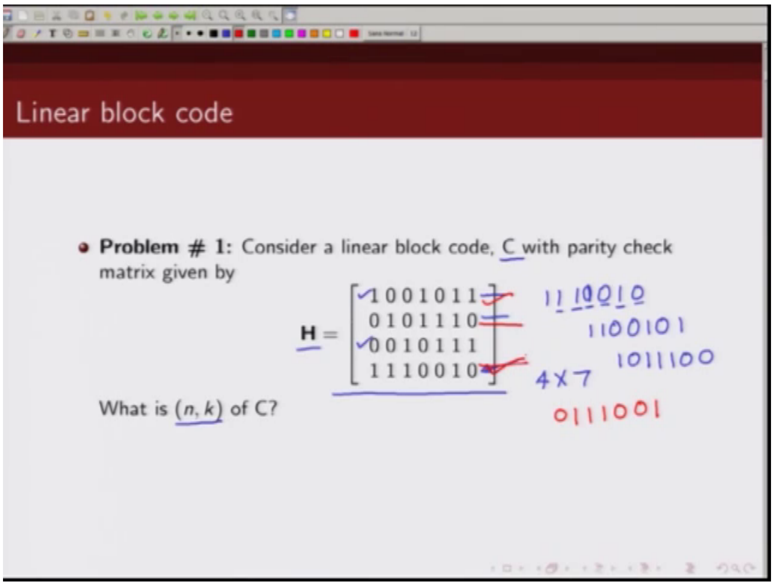
and this. If we add first row and fourth row, what do we get? 0 1 1 1 0 0 and

(Refer Slide Time 03:12)



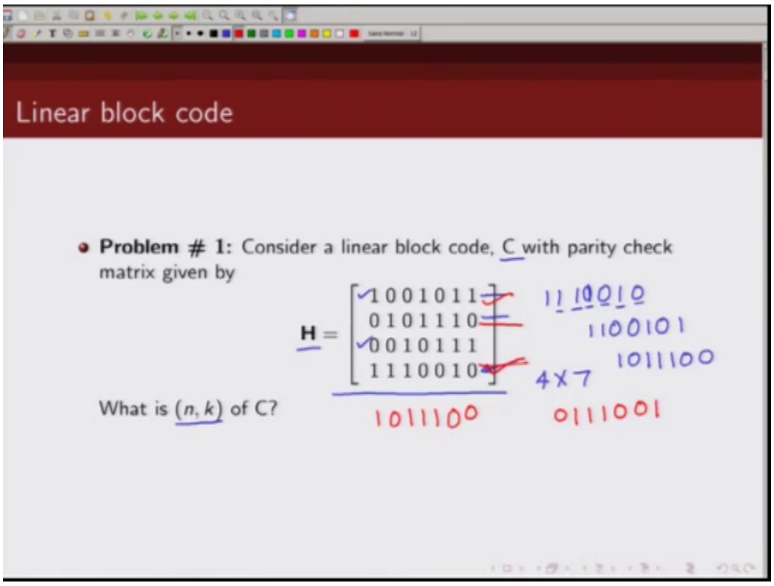
1, and note row number 3 and 2 are not same as this. Like that we can check, we can check for example row 2 and 4, we add up row 2 and 4,

(Refer Slide Time 03:23)



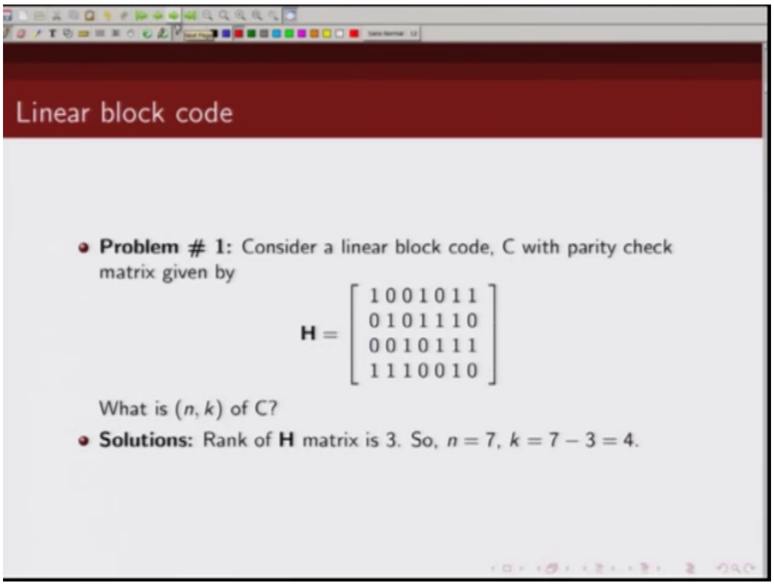
what do we get? 1 0 1 1 1 0 0, now note row number 3

(Refer Slide Time 03:36)



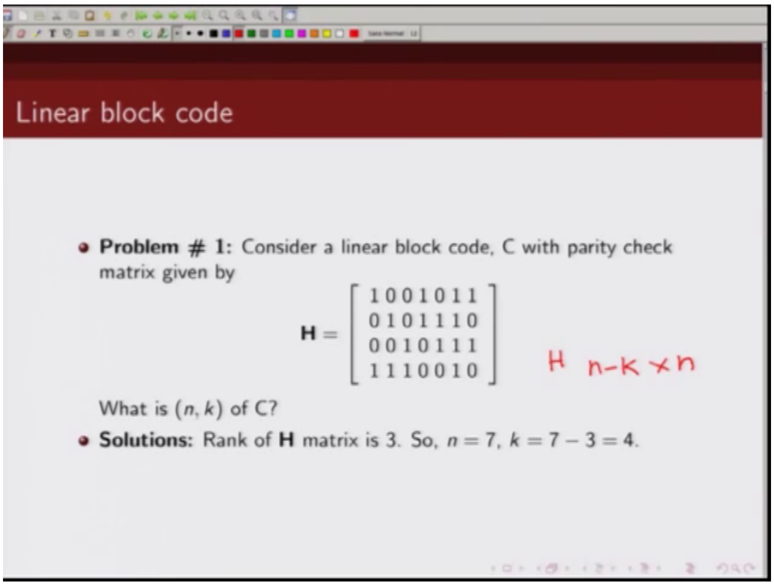
and row number 1 are not same as this. So we can see that any 3 rows do not add up to 0. So the rank of this matrix H is 3. So

(Refer Slide Time 03:49)



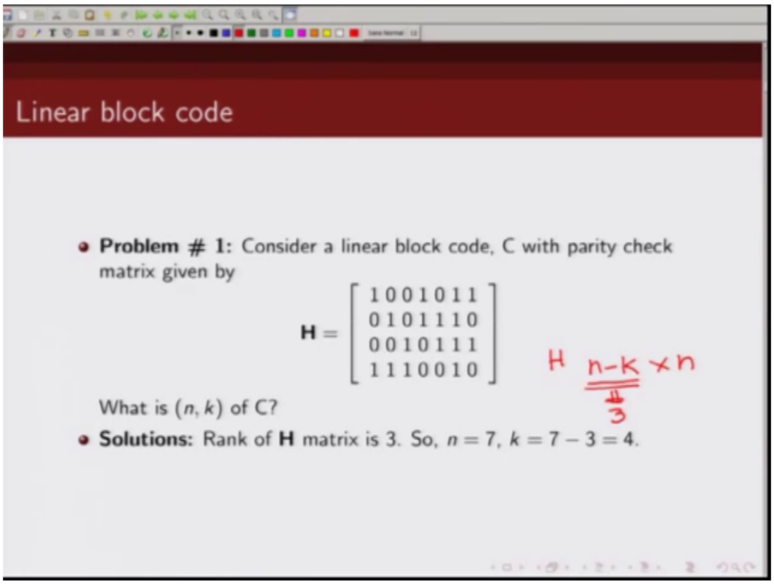
rank of this matrix is 3. Now we know parity check matrix is n minus k cross n. So

(Refer Slide Time 03:58)



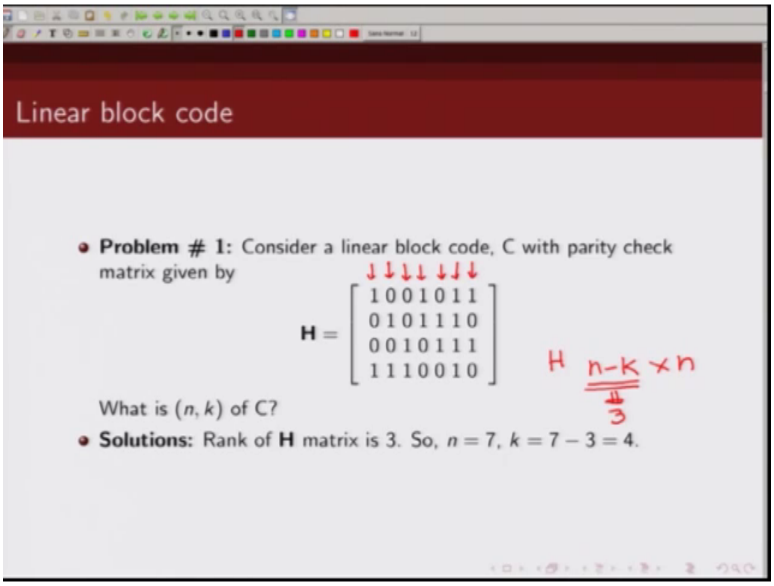
n minus k is in our case, equal to 3 and what is

(Refer Slide Time 04:04)



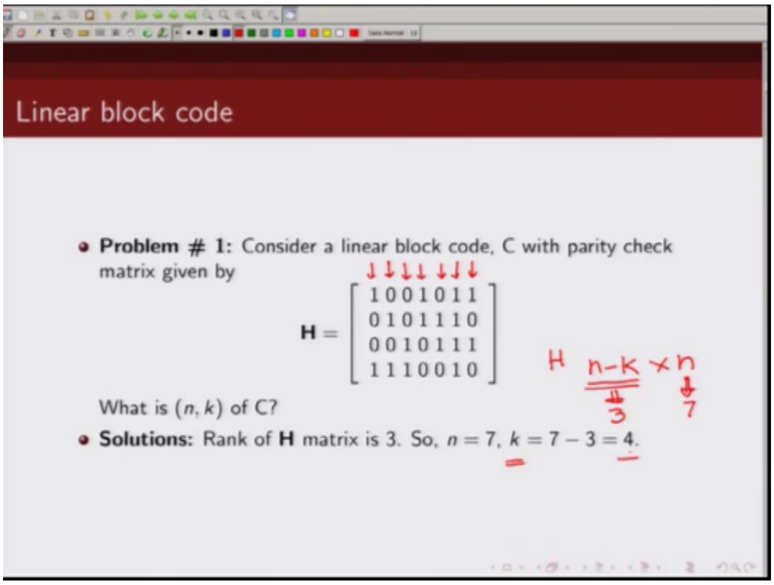
n, number of columns of this. So that's 1, 2, 3, 4, 5, 6, 7. So

(Refer Slide Time 04:14)



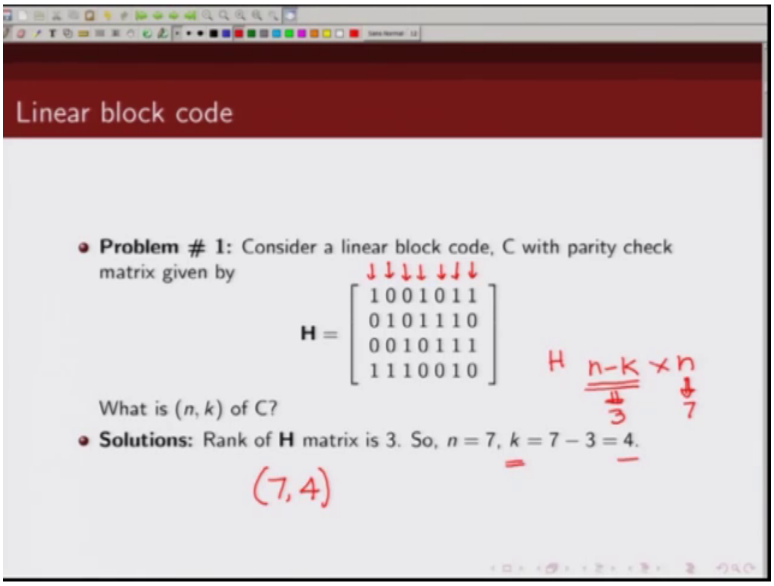
n is 7. So that would then give us k equal to 4. So this

(Refer Slide Time 04:23)



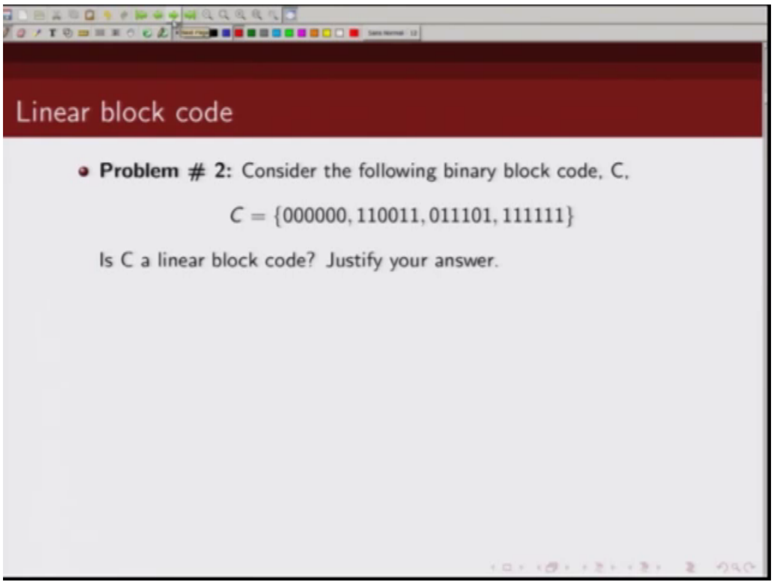
this is an example parity check matrix for a 7 4 linear block code

(Refer Slide Time 04:32)



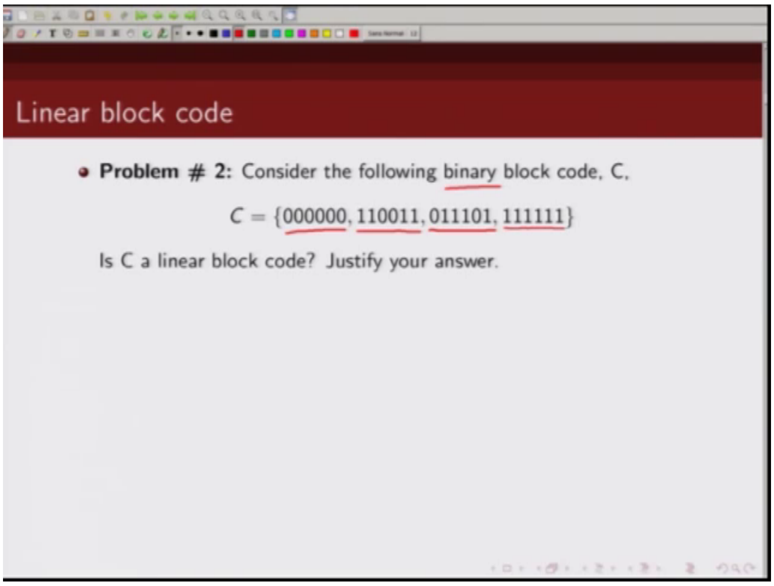
Ok.

(Refer Slide Time 04:35)



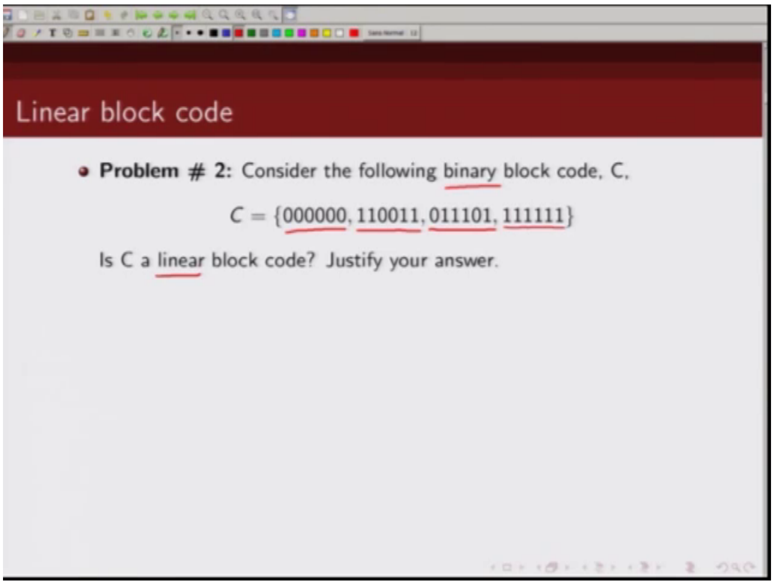
Now let's look at another problem. You are given a set of codewords. And what are these codewords? These are binary codewords. So this is all zero, 1 1 0 0 1 1, 0 1 1 1 0 1 and 1 1 all 1

(Refer Slide Time 04:54)



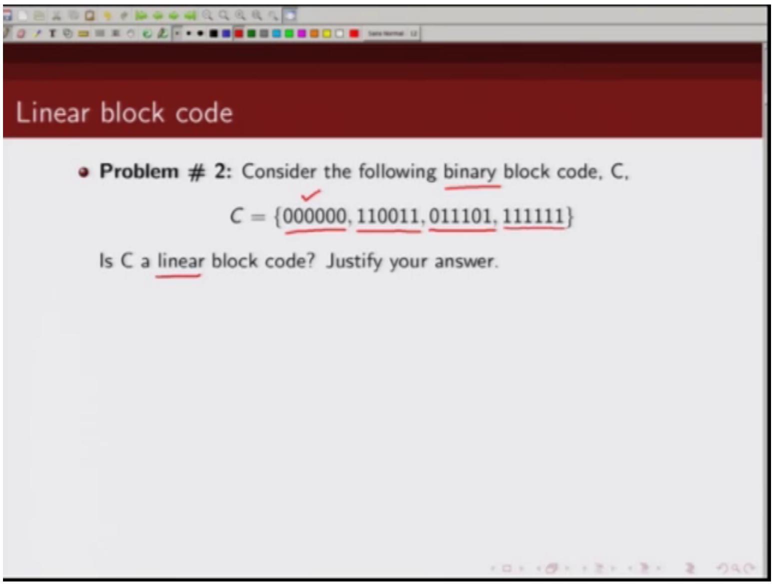
And the question that has been asked is, is this a linear code? Is this a linear code?

(Refer Slide Time 05:04)



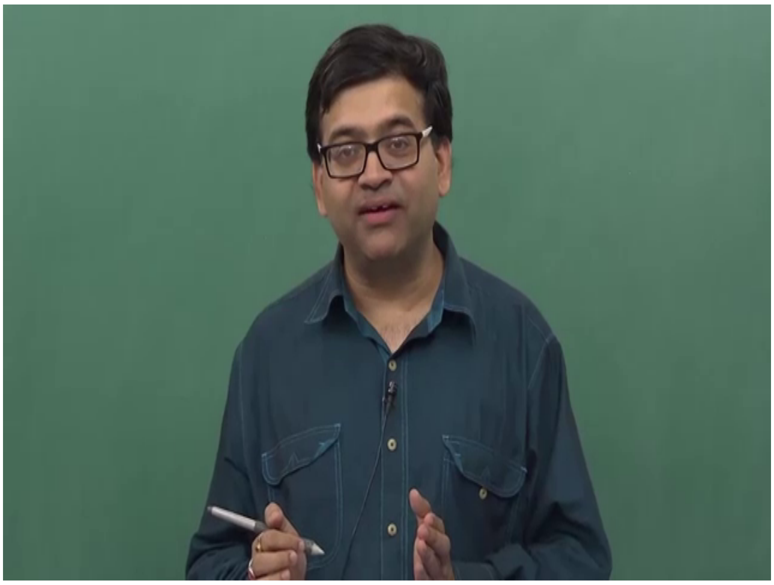
Now what do we know about linear code? A linear code should have all 0 codewords which this codeword has. And sum

(Refer Slide Time 05:13)



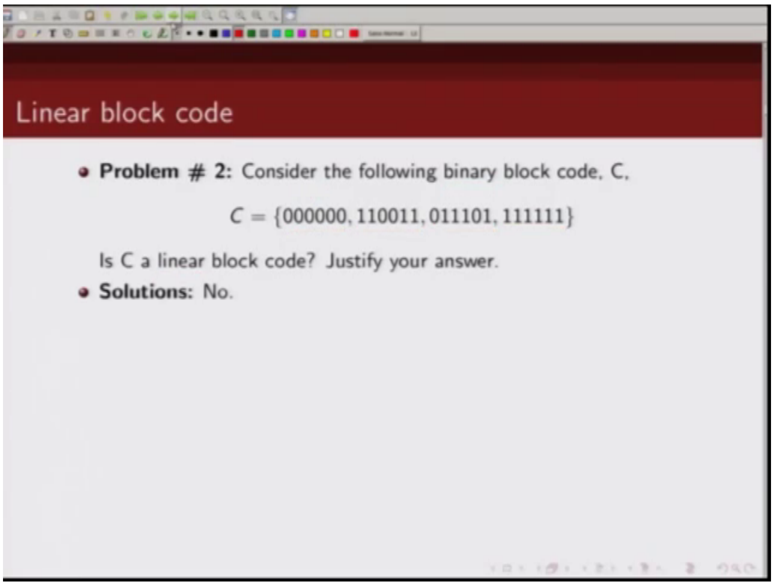
of any two codewords

(Refer Slide Time 05:15)



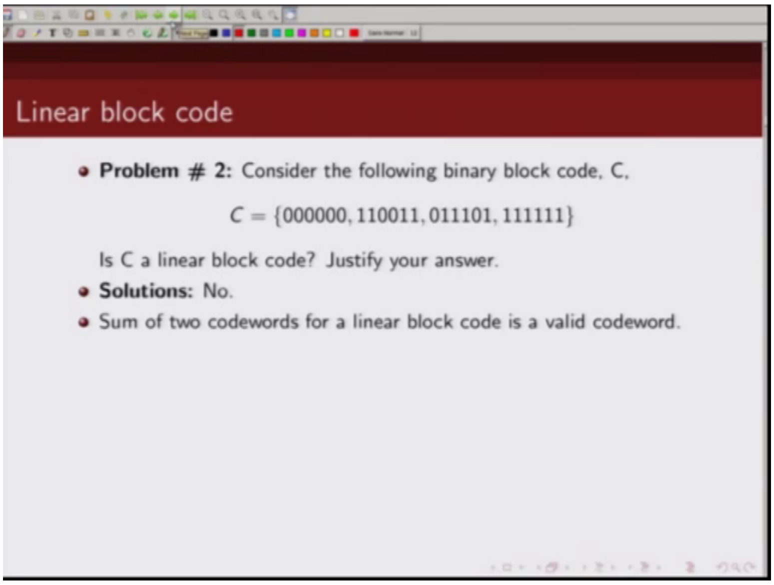
is also a valid codeword. So let's see.

(Refer Slide Time 05:19)



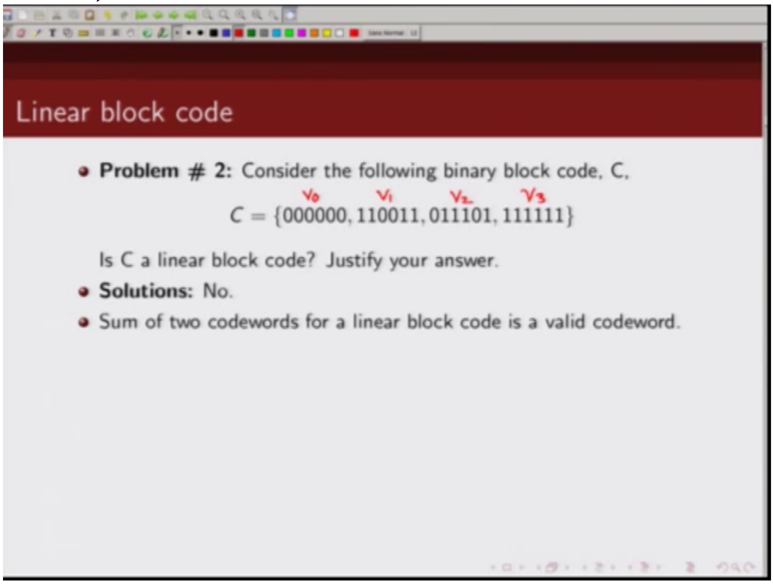
So let's see if sum of all codewords is already a valid codeword.

(Refer Slide Time 05:25)



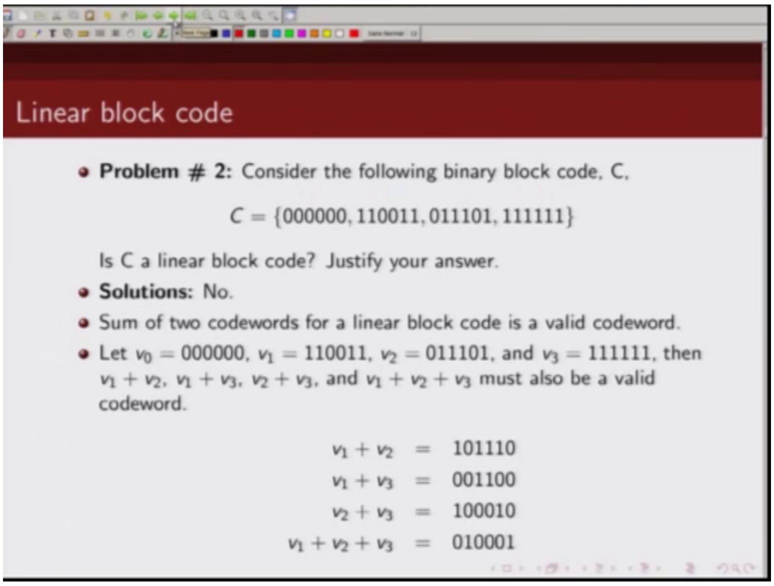
So let's see if sum of all codewords is already a valid codeword.

(Refer Slide Time 05:33)



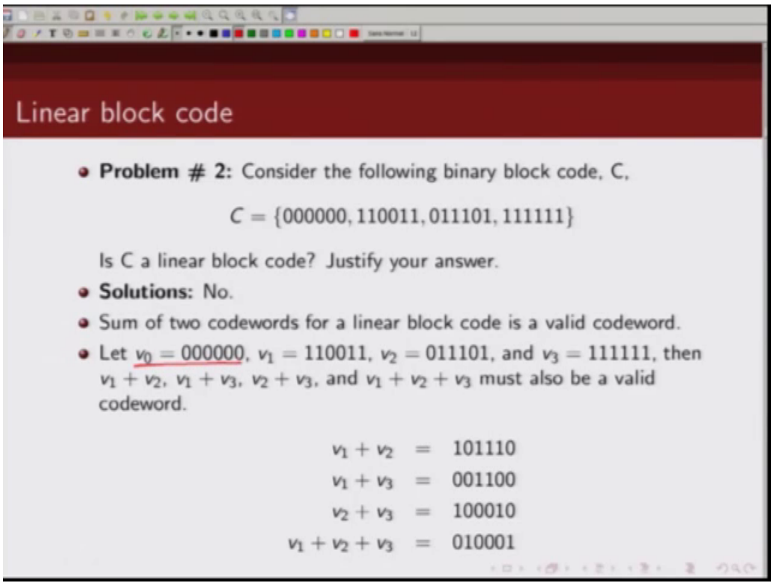
So let's call this v 0, v 1, v 2 and v 3. So what we want is all possible combinations of v 0, v 1, v 2, v 3 should also be a valid codeword. They should be in C. So let's see.

(Refer Slide Time 05:44)



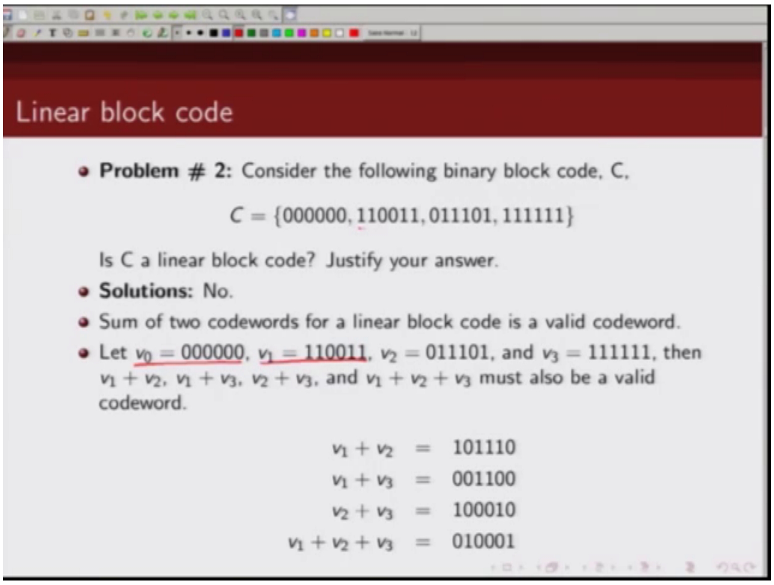
So, as I said we take v 0 to means all zero codewords,

(Refer Slide Time 05:49)



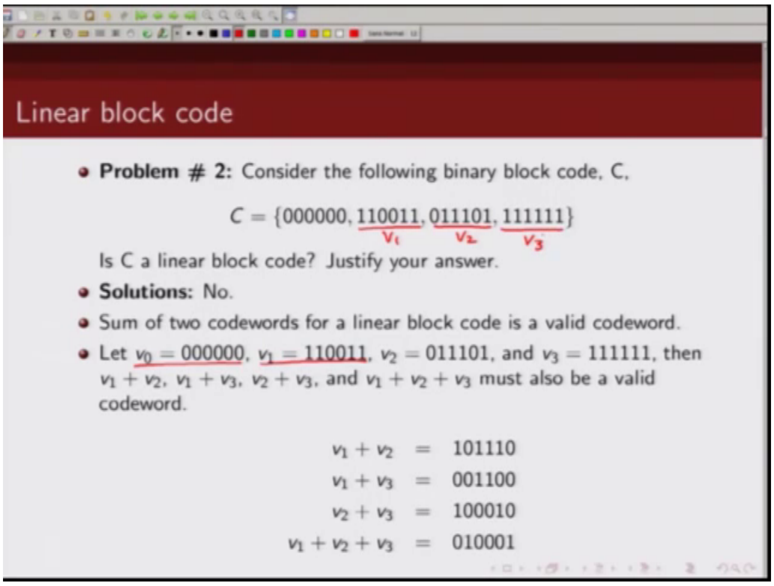
is given by this, this

(Refer Slide Time 05:51)



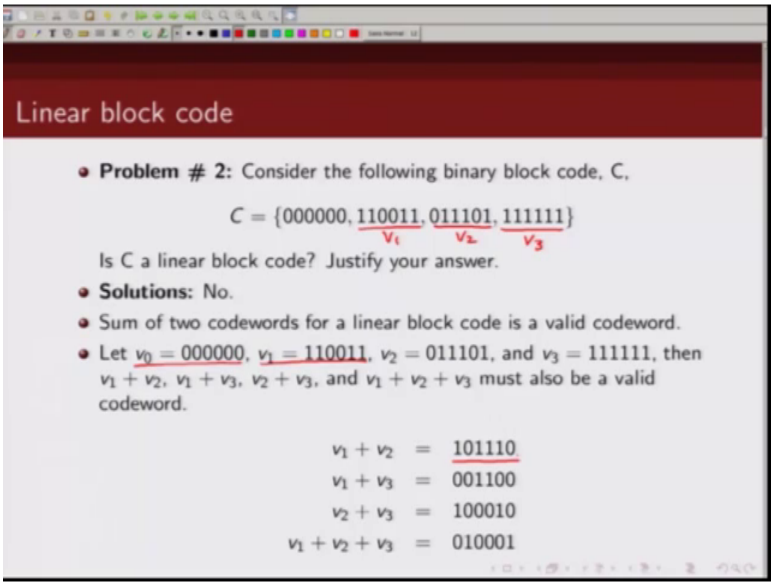
is v 1, this is v 2 and this is v 3. Now let's see all possible

(Refer Slide Time 05:58)



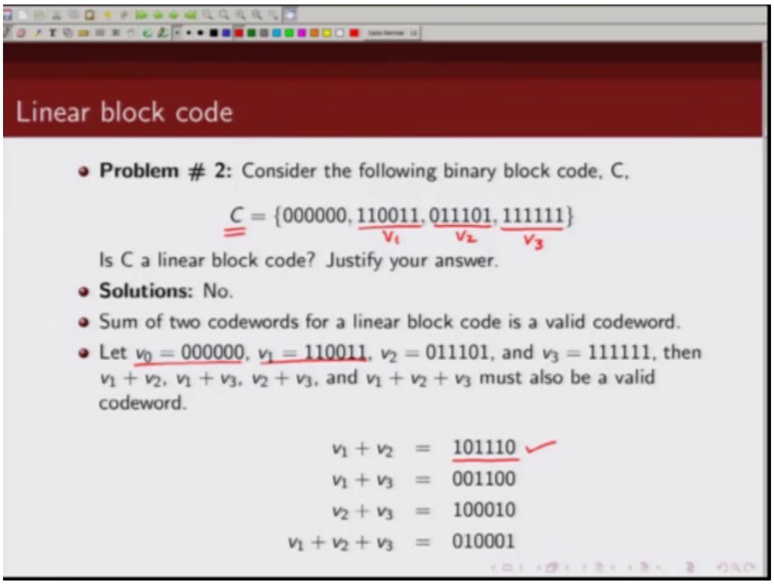
combinations of v 1, v 2, v 3, the non-zero codewords. So we consider v 1 plus v 2. What is v 1 plus v 2? v 1 plus v 2 is, you can see this is 1 0 1 1 1 0, it's given by this.

(Refer Slide Time 06:20)



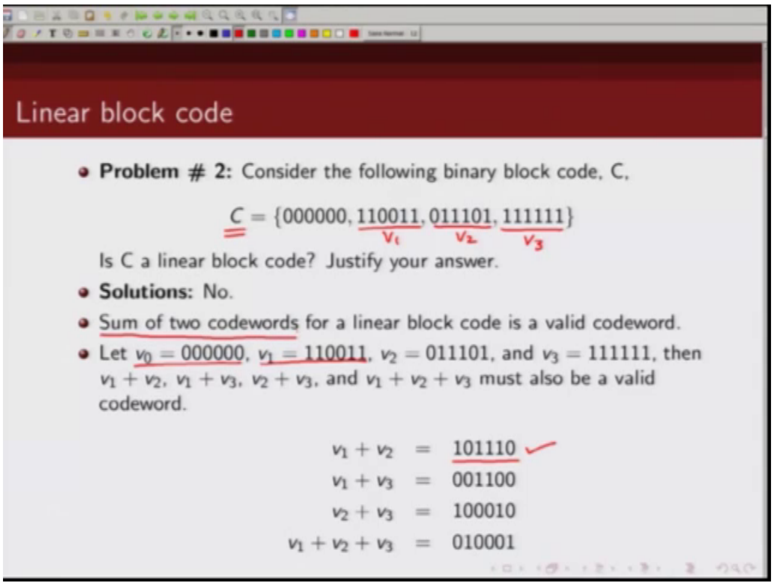
Now is this codeword in C? We don't see any codeword which is 1 0 1 1 1 0 listed here. That means this C

(Refer Slide Time 06:36)



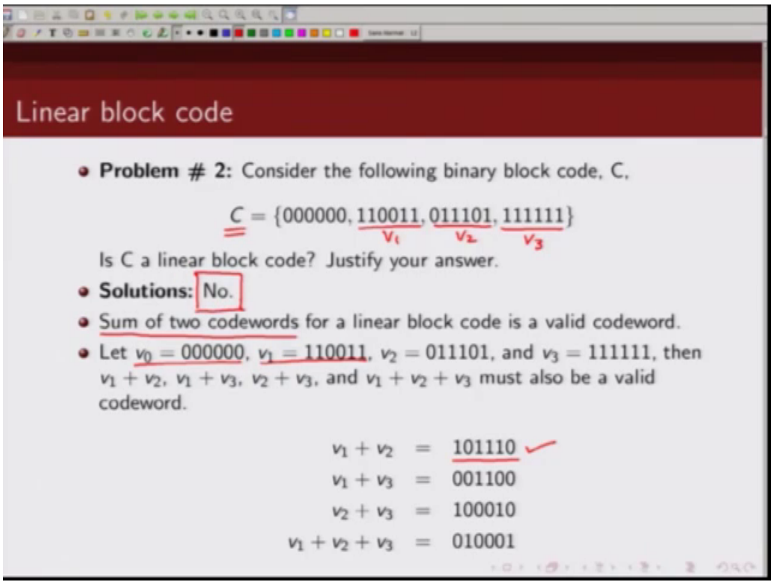
is not a linear code. Why it's not a linear code, because sum of any two codewords

(Refer Slide Time 06:45)



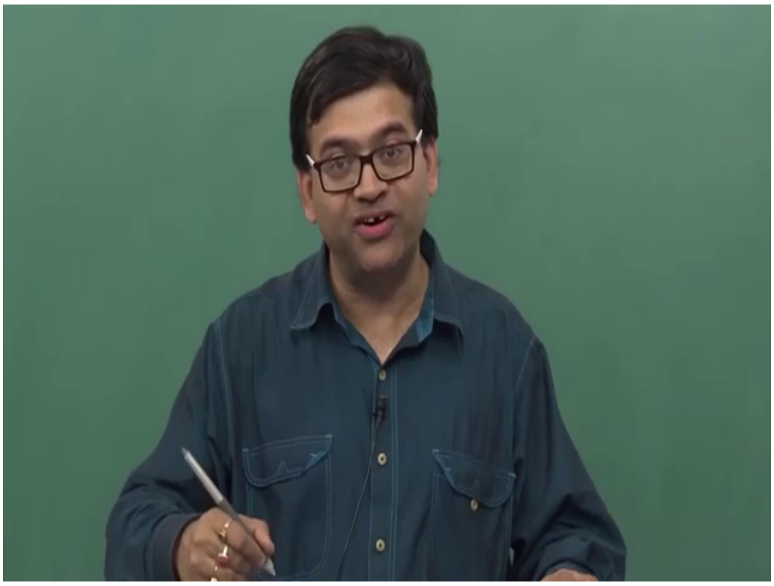
is also a valid codeword. Now v 1 and v 2 are valid codewords in C. So sum of v 1 plus v 2 should also be in C. But we notice that 1 0 1 1 1 0, which is sum of v 1 plus v 2 is not there in C. And that's why we say that C is not a

(Refer Slide Time 07:08)



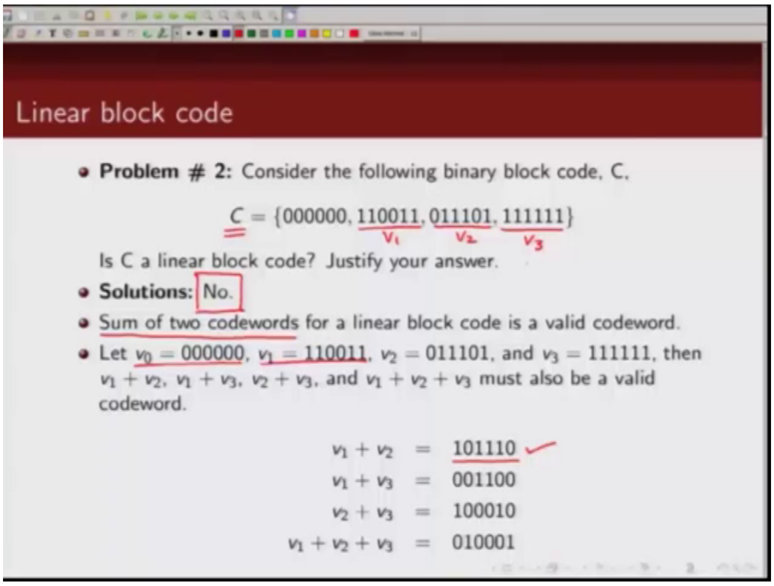
linear block code. Now my next question is can we add additional codewords here

(Refer Slide Time 07:16)



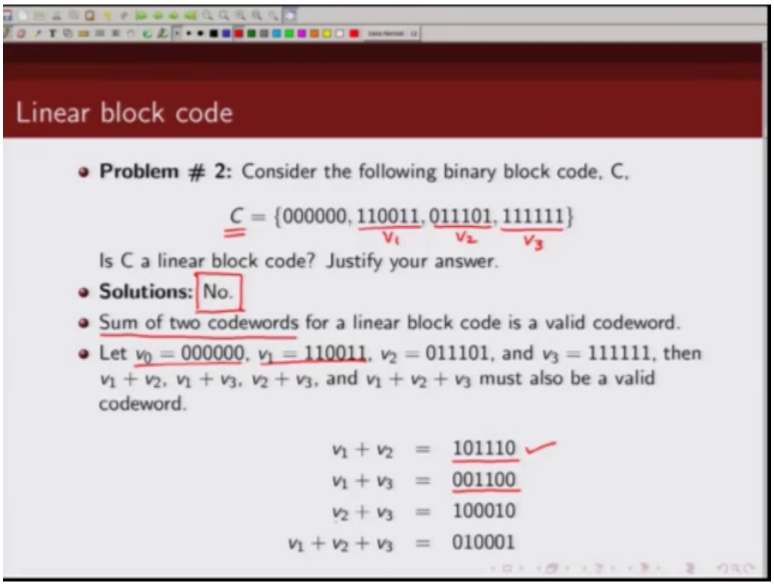
such that C becomes a linear block code? Now how do we do that? To do that, we will have to ensure all possible combinations of these

(Refer Slide Time 07:26)



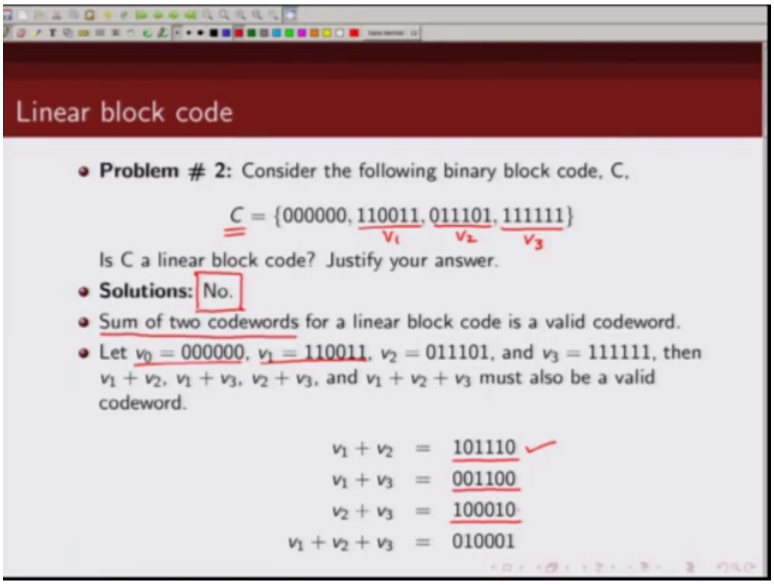
codewords is also there in C. So let's compute v 1 plus v 3 which is basically given by 0 0 1 1 0 0. Let's look at

(Refer Slide Time 07:37)



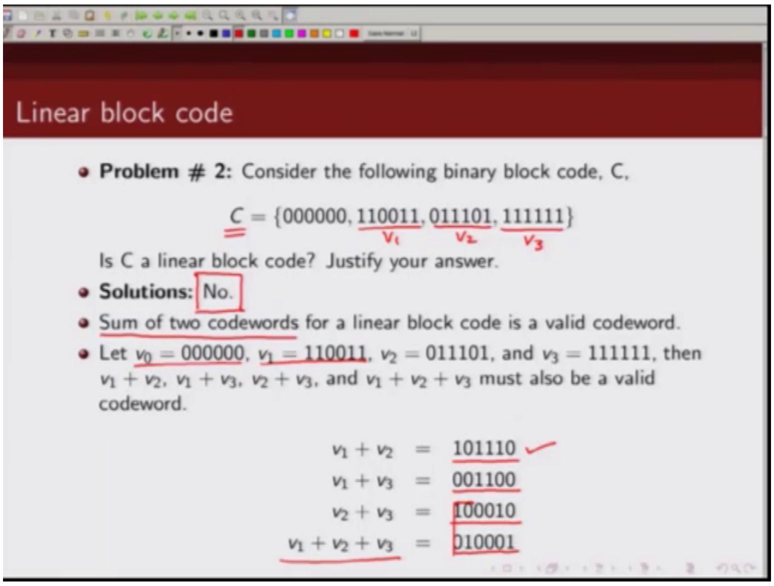
v 2 plus v 3 which is given by

(Refer Slide Time 07:40)



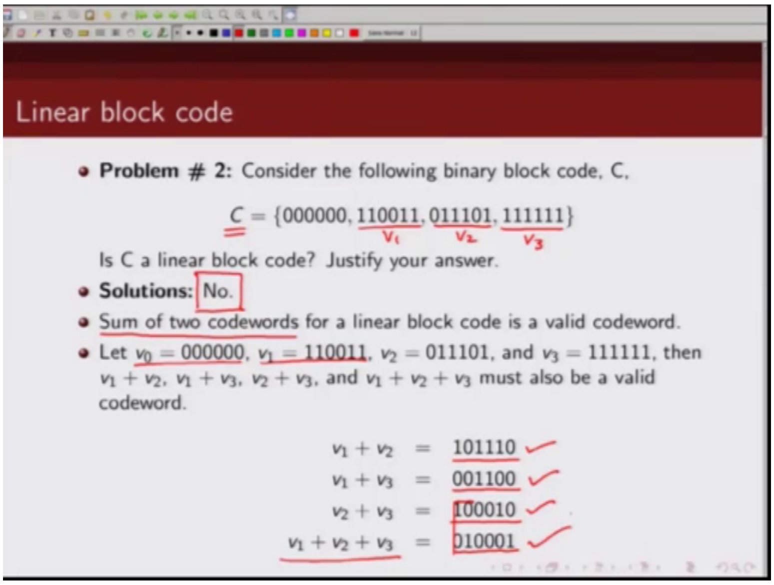
1 0 0 1 0. And let's look at v 1 plus v 2 plus v 3, is basically given by

(Refer Slide Time 07:48)



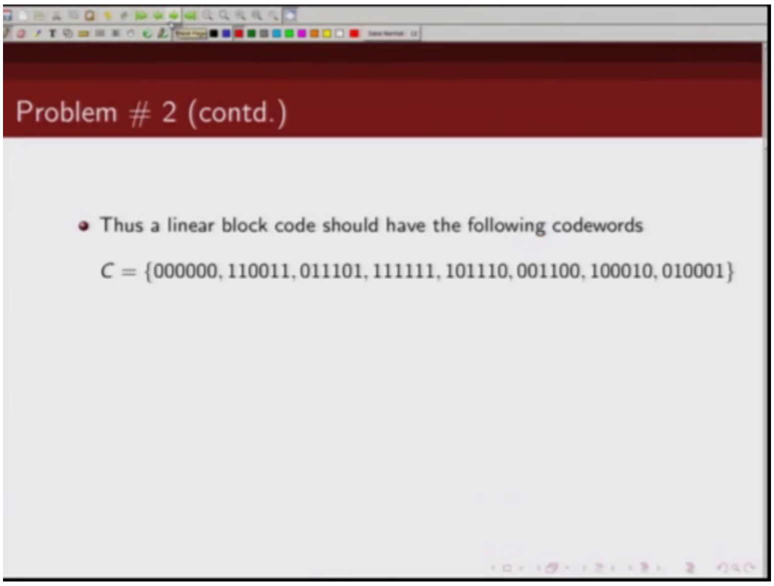
0 1 0 0 0 1. So note that I have listed all possible combinations of these codewords here. Now none of these sums are there

(Refer Slide Time 08:04)



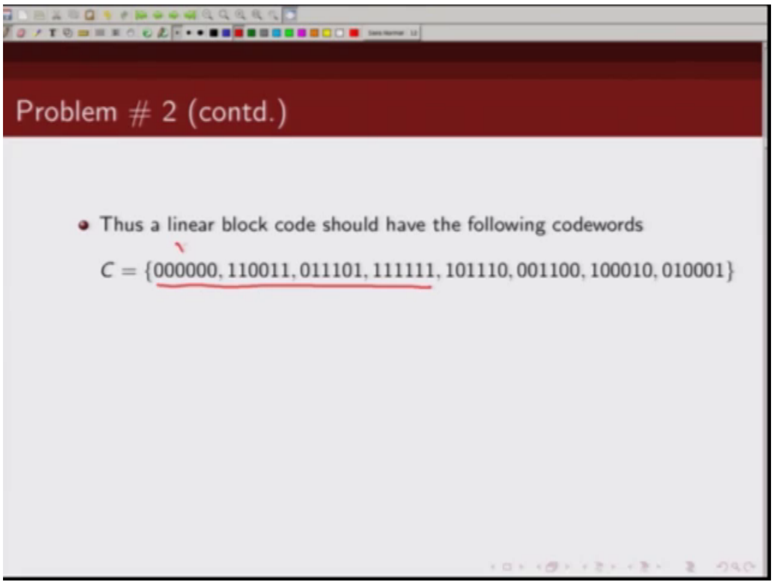
in this linear block code. So if we add them in this set of C, set of codewords, then we, our block code C will become a linear block code. So if we want to make it a linear

(Refer Slide Time 08:20)



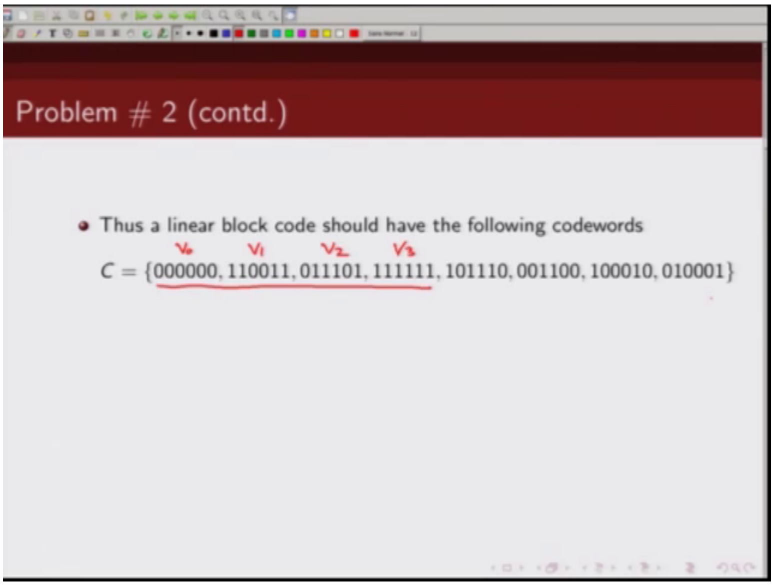
block code, what do we need to do? In this set of 4 codewords v 0,

(Refer Slide Time 08:26)



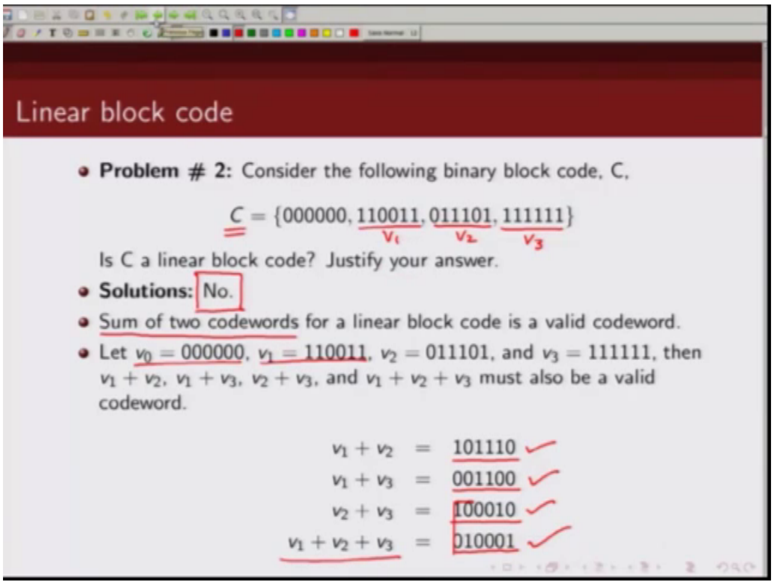
v 1, v 2 and v 3 we need to add these set of

(Refer Slide Time 08:32)



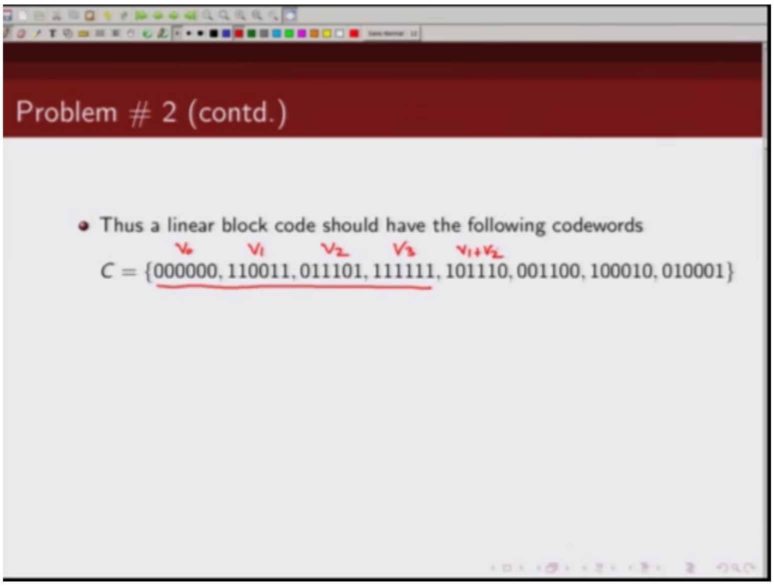
codewords which was basically

(Refer Slide Time 08:36)



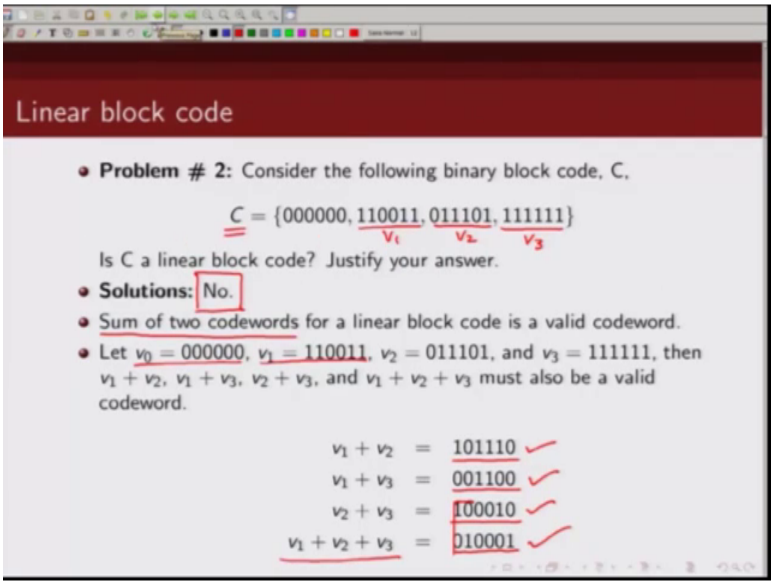
v 1 plus v 2. This is v 1 plus v 2.

(Refer Slide Time 08:41)



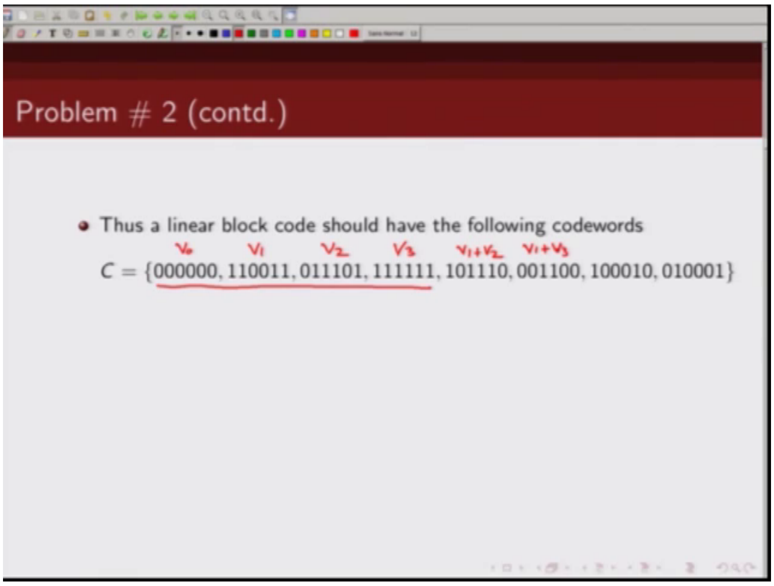
This is

(Refer Slide Time 08:43)



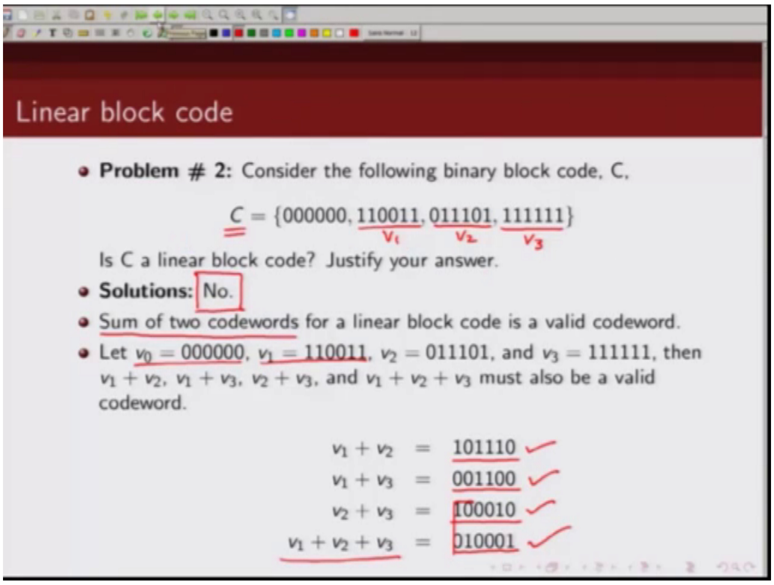
v 1 plus v 3, v 1 plus v 3.

(Refer Slide Time 08:49)



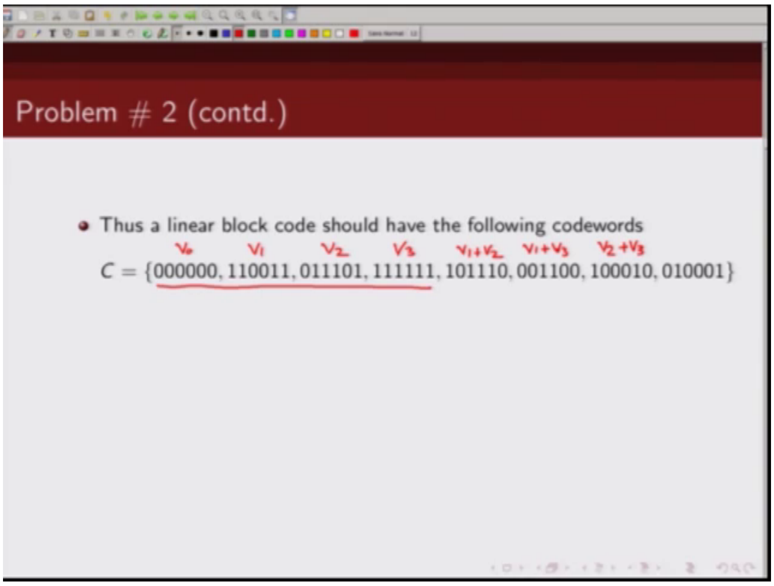
Then this one is

(Refer Slide Time 08:51)



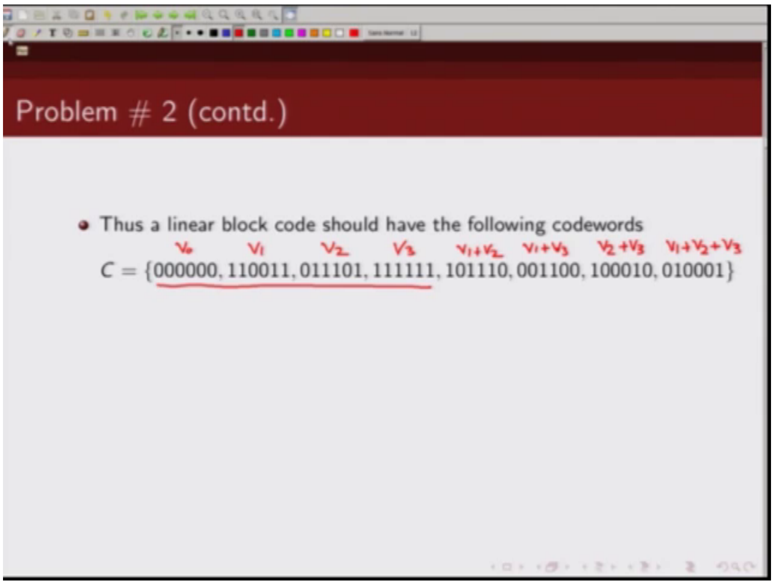
v 2 plus v 3, v 2 plus v 3 and this one was

(Refer Slide Time 08:58)



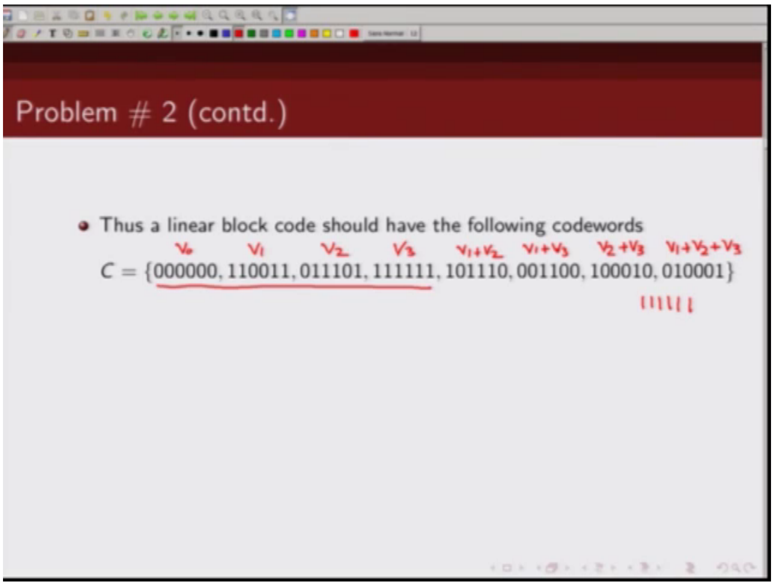
v 1 plus v 2 plus v 3.

(Refer Slide Time 09:03)



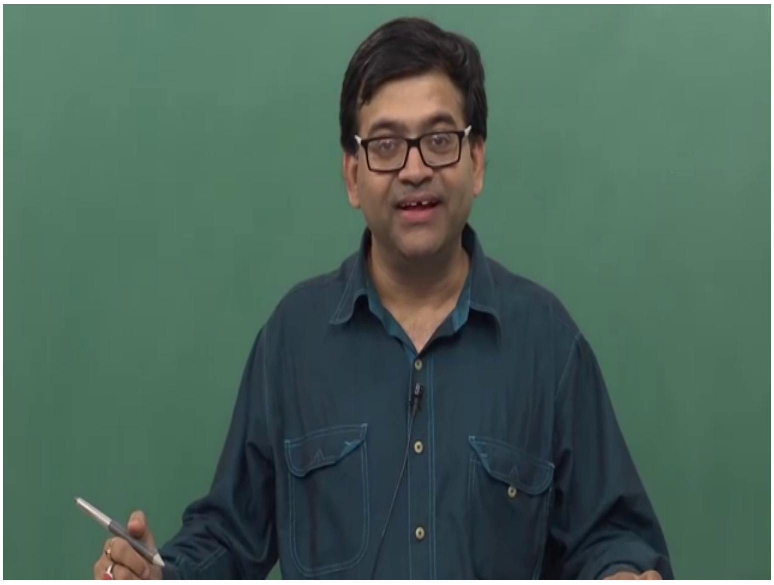
So let's look at these 2 codewords. This is v 1 plus v 2 and this is v 1 plus v 2 plus v 3. So if we add these two, what we will get is v 3. We can double check. So if we consider add these two, the first bit will be 1, this 0 plus 1 will be 1, then 1 plus 0 will be 1, then 1 plus 0 will be 1, then 1 plus 0 will be 1 and 0 plus 1 will be 1. And this is already there in this set of codewords. This is v 3, Ok. Similarly take these two.

(Refer Slide Time 09:45)

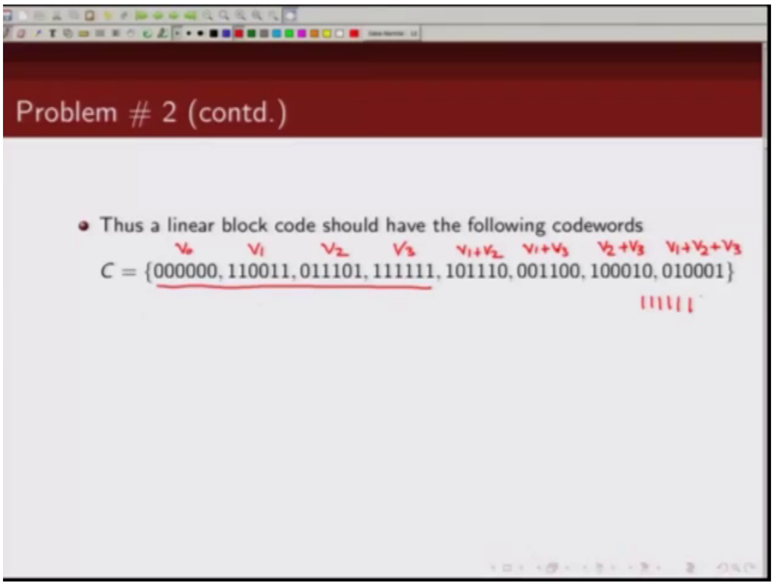


This one is v 1 plus v 2 and this is v 2 plus v 3. If we add them, what we get is v 1 plus v 3. We will get this. If we consider these two we will get v 2. We will consider this, we will get v 3. If we consider these two, sum of these two, we will get v 3. If we consider sum of these three, what we will get, we will get v 3. So you can see basically, linear combinations of all these codewords is already there in

(Refer Slide Time 10:21)

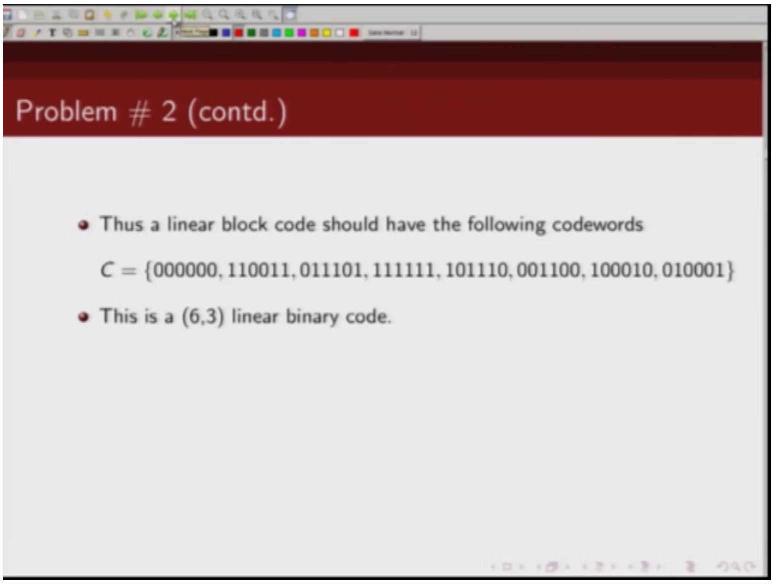

this C. So this C which contains the set of 8 codewords

(Refer Slide Time 10:26)



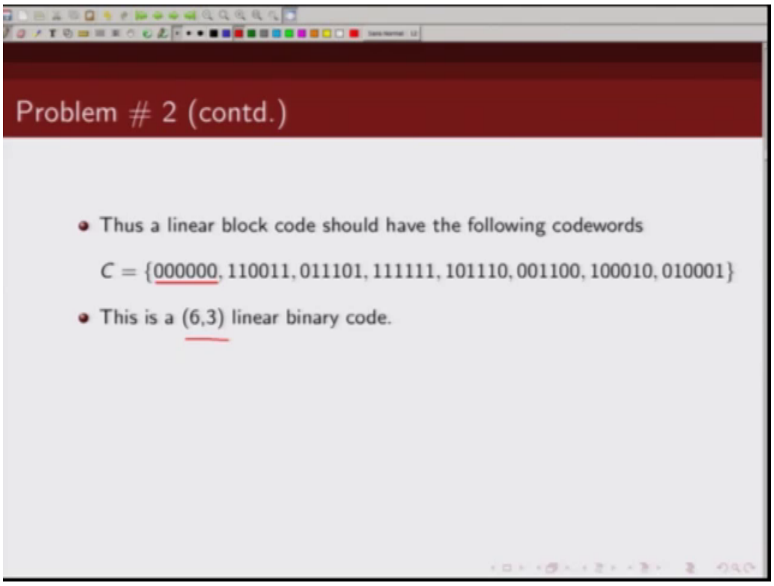
is a linear code.

(Refer Slide Time 10:34)



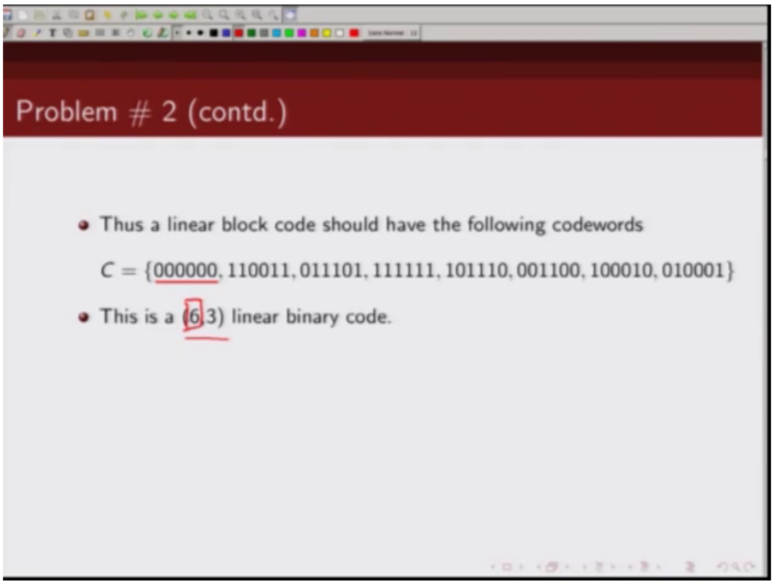
And what are the parameters n and k? Now the length of the codewords is 6.

(Refer Slide Time 10:41)



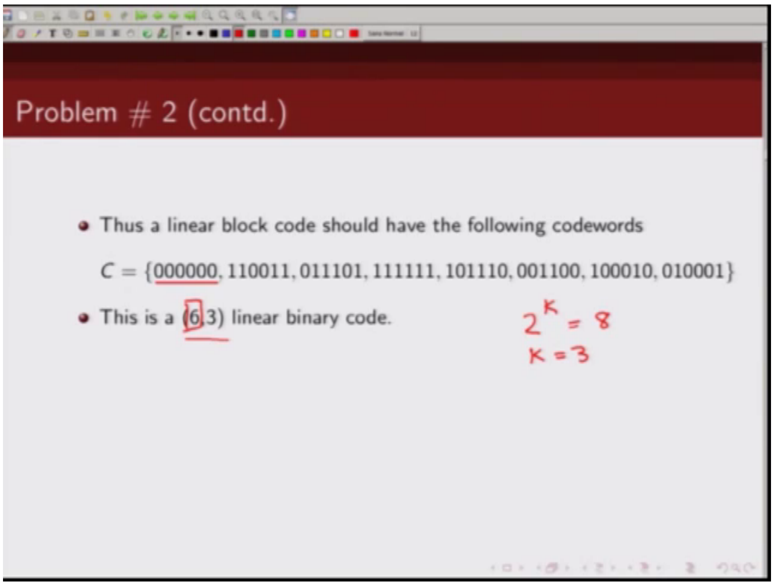
Each of these codewords are 6 bits. So that's why n is 6. And there are total

(Refer Slide Time 10:47)



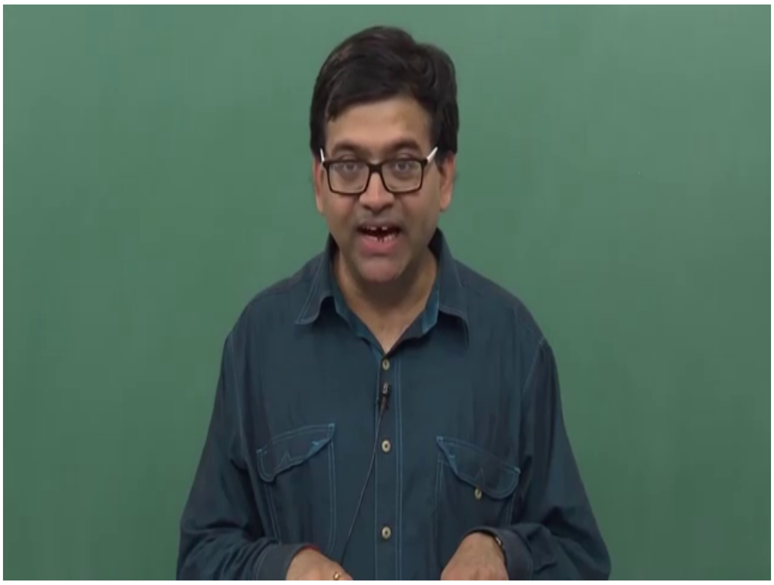
2 k codewords. And in our case 2 k is basically 8. So k is 3. So this

(Refer Slide Time 10:55)



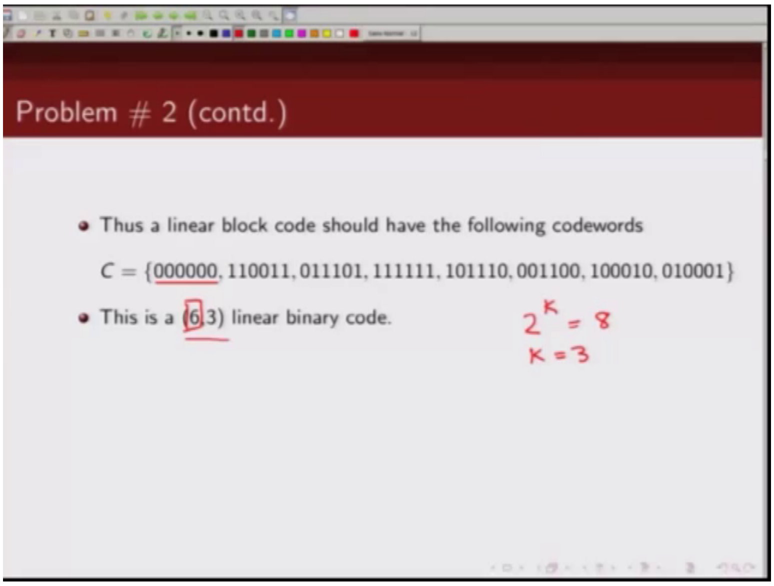
is basically a 6 3 linear

(Refer Slide Time 11:00)



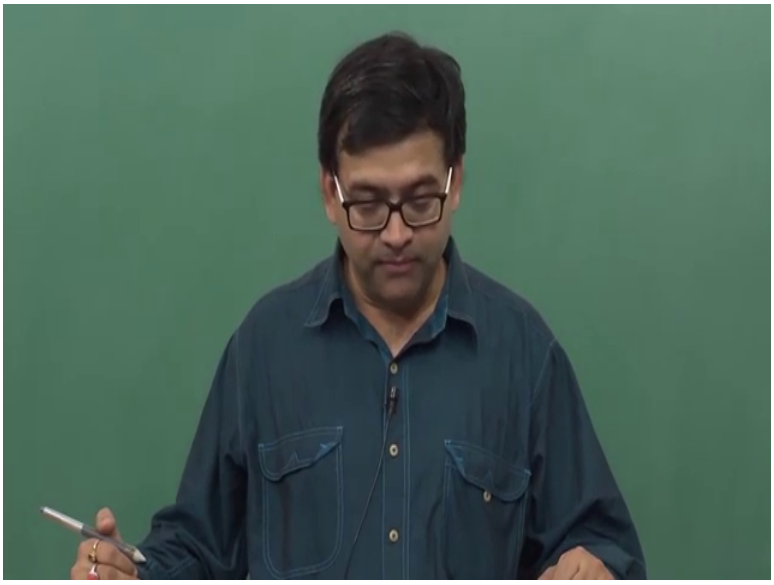
binary code.

(Refer Slide Time 11:03)



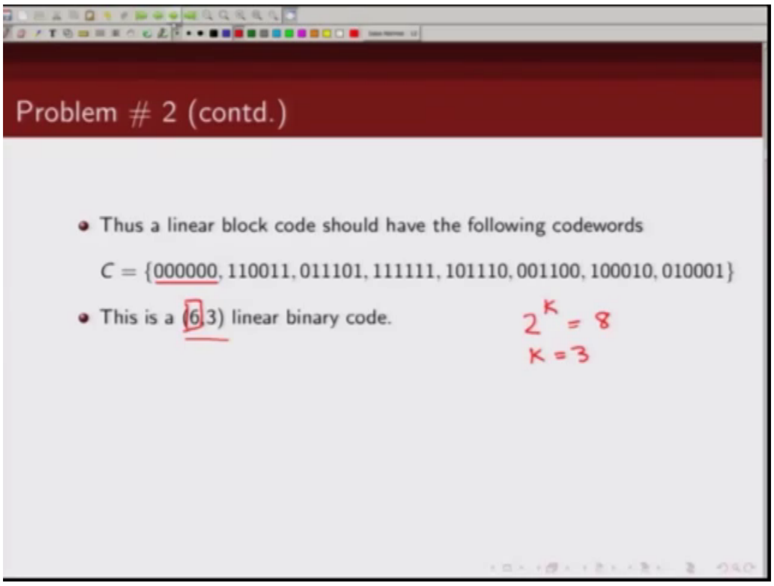
Now if I ask you, tell me what is a generator matrix that will generate this set of codewords?

(Refer Slide Time 11:10)



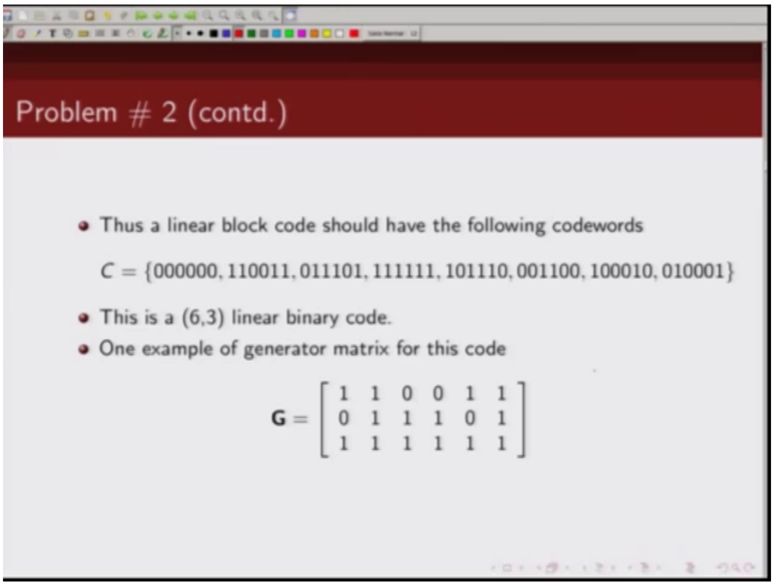
Now how can you do that?

(Refer Slide Time 11:12)



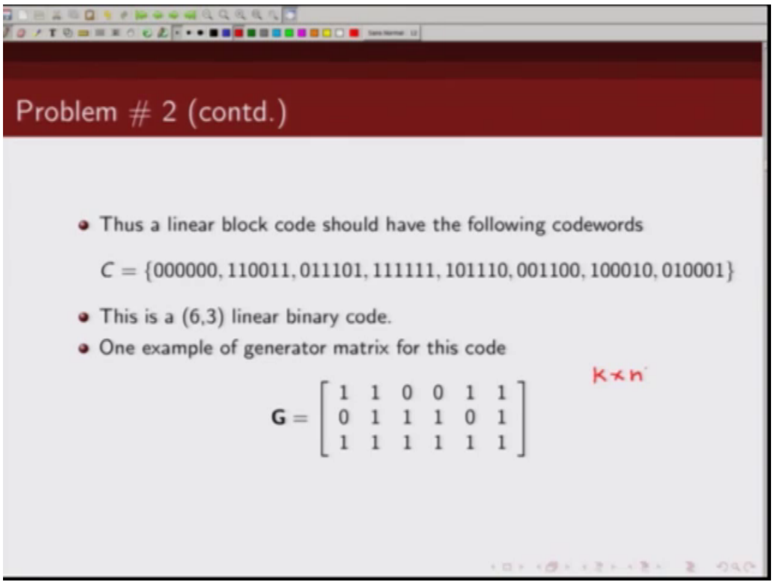
So we know the generator matrix. It's basically

(Refer Slide Time 11:16)



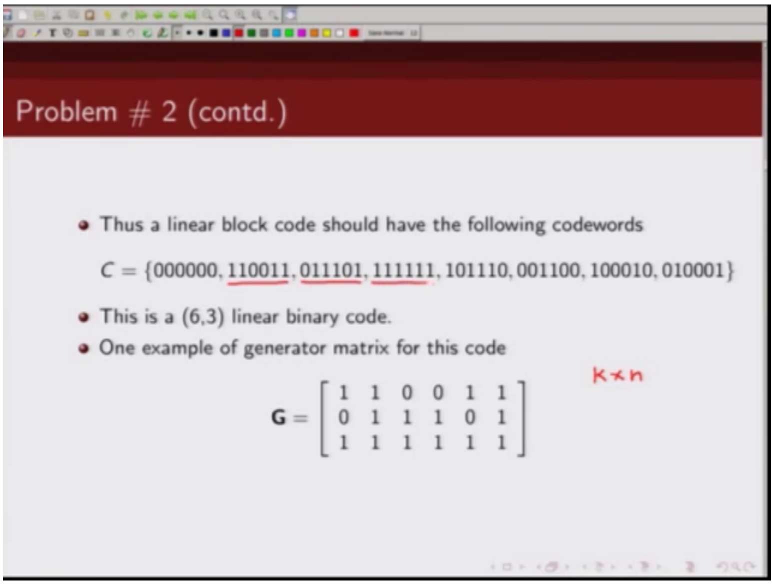
a k cross n matrix,

(Refer Slide Time 11:18)



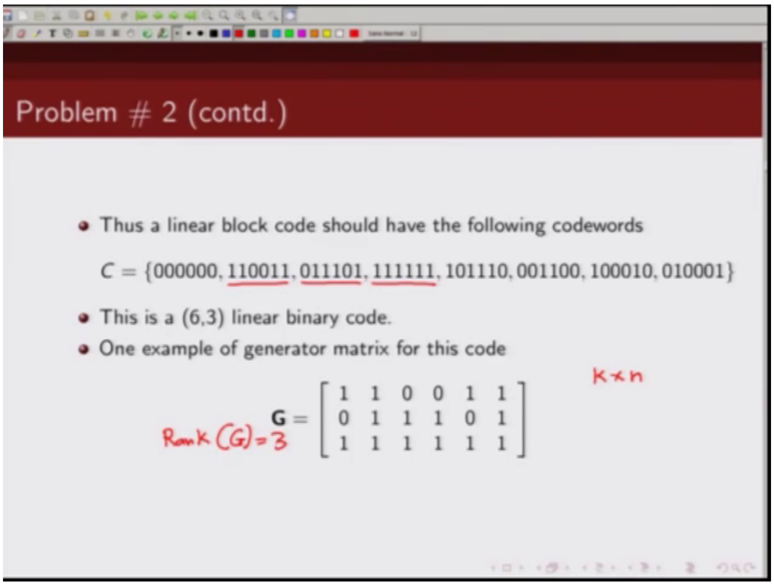
right. So if you take basically 3, k in this case is 3, if you take 3 codewords which are linearly independent basically, if you take them and form them as rows of your generator matrix, then you get your generator matrix. So I just took this v 1, v 2 and v 3

(Refer Slide Time 11:42)



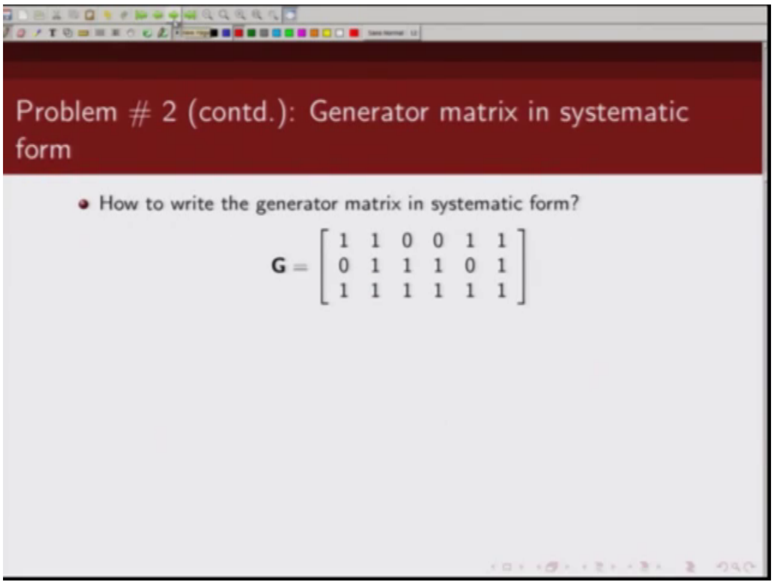
and you can verify that rank of this matrix G is 3. It's full rank,

(Refer Slide Time 11:50)



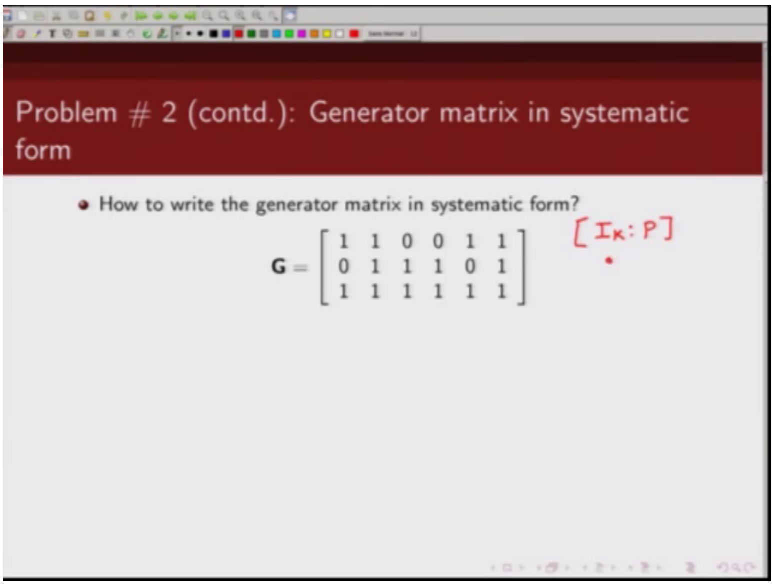
Ok. So then this G will be able to, this generator matrix will be able to generate this set of codewords.

(Refer Slide Time 12:03)



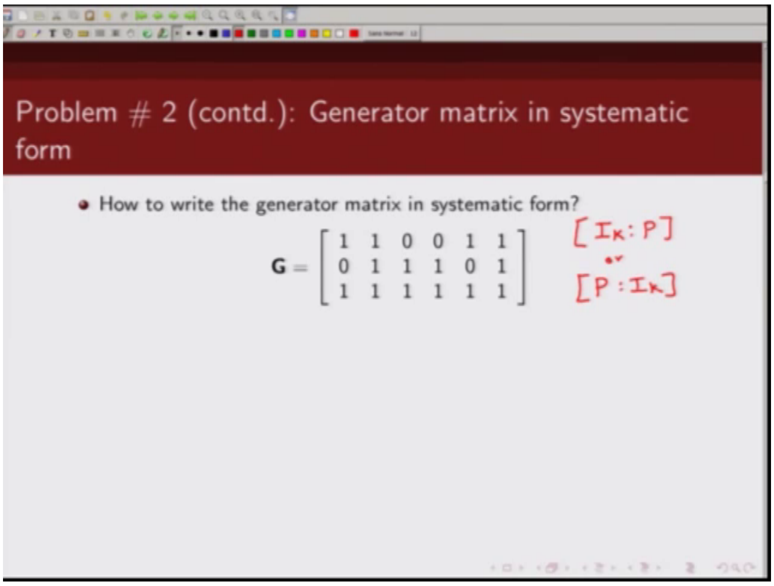
Now can we put this, is this generator matrix in systematic form? The answer is no. Because to get it in systematic form, what we need is our generator matrix should be of the form like this, or

(Refer Slide Time 12:21)



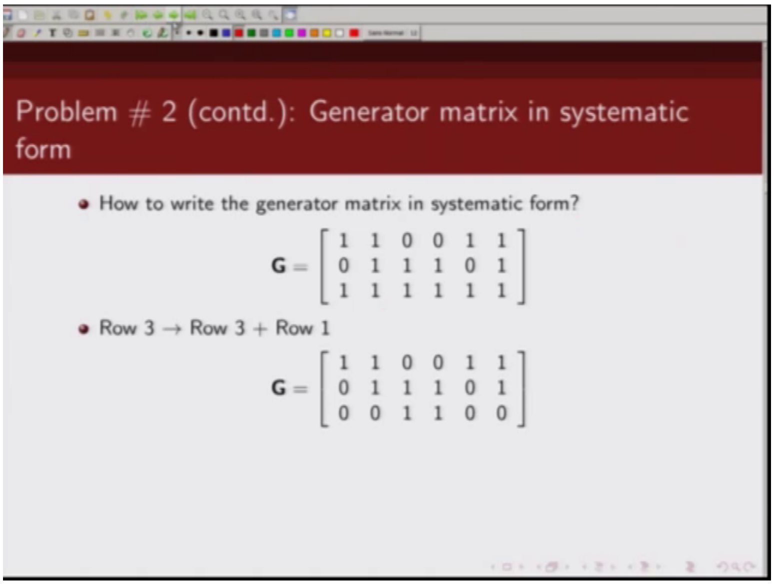
something like this, Ok.

(Refer Slide Time 12:26)



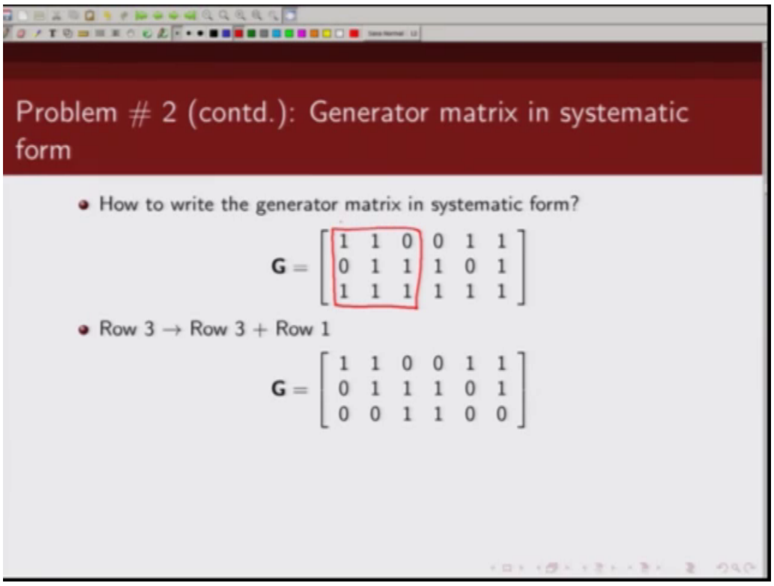
But this is not in this particular form. So we will have to get some identity matrix and some matrix P. Now by doing elementary row operation, we can put this in systematic form. So let's do that.

(Refer Slide Time 12:43)



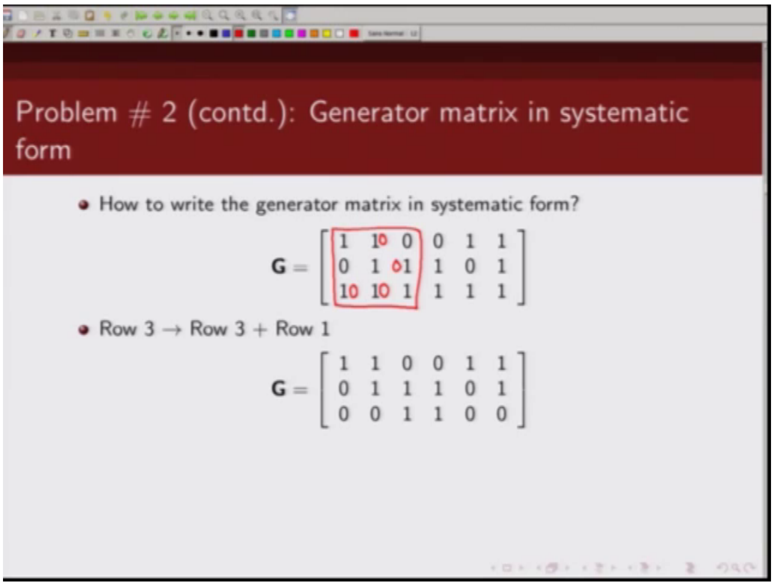
So note, if we want to get, let's say this in the form of identity

(Refer Slide Time 12:49)



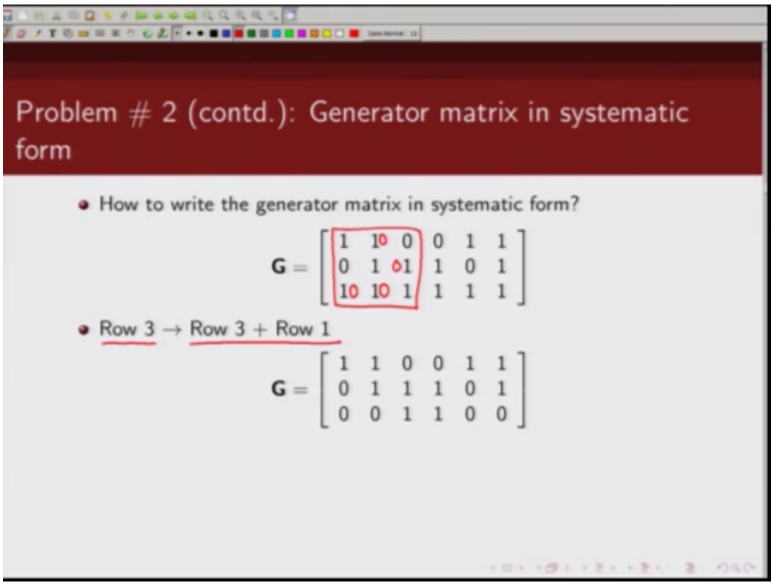
what do we need? We would need basically here, we would need a 0, here we would need a 0, here we would need a 0, here we would need a 0, right?

(Refer Slide Time 13:00)



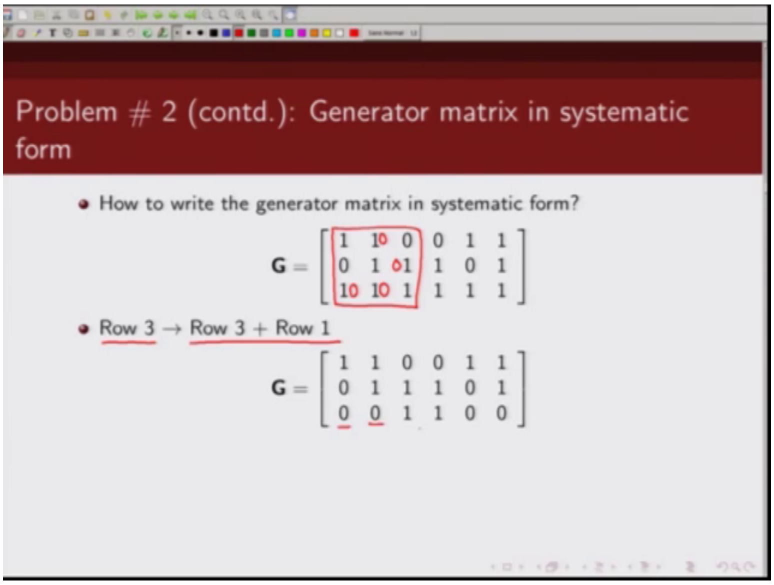
So first let's try to get this 1 to 0. Now how can we make this 0? So if we do this transformation that row 3 is row 3 plus row 1.

(Refer Slide Time 13:14)



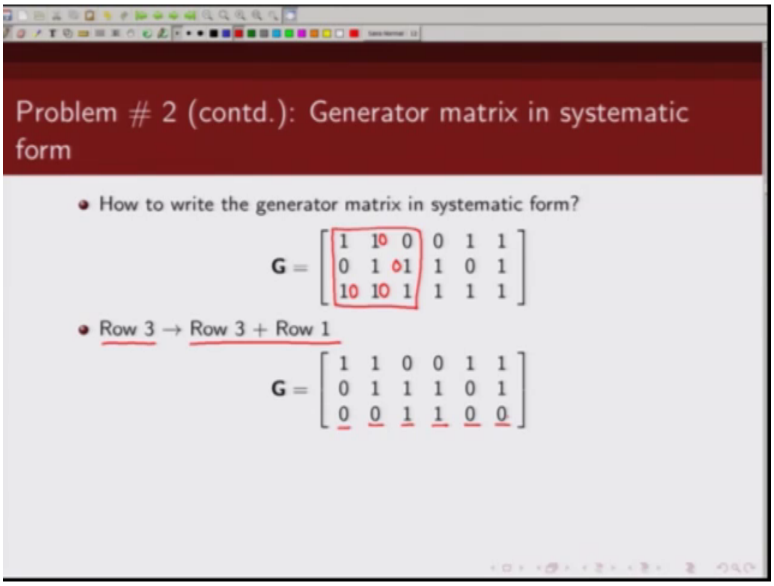
So row 3 is row 3 plus row 1, if we do that then 1 plus 1, this will be 0. 1 plus 1, this is 0.

(Refer Slide Time 13:24)



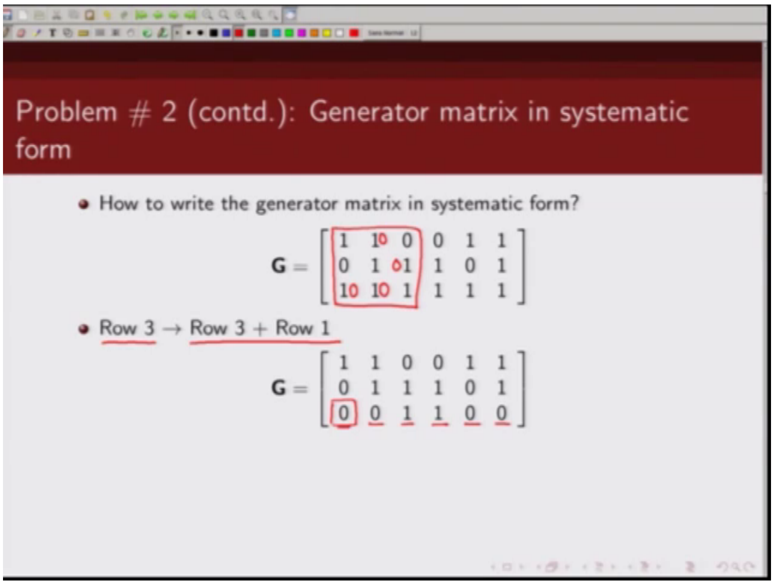
. 0 plus 1, this is 1, 0 plus 1, this is 1, 1 plus 1, this is 0 and 1 plus 1, this is 0, Ok.

(Refer Slide Time 13:35)



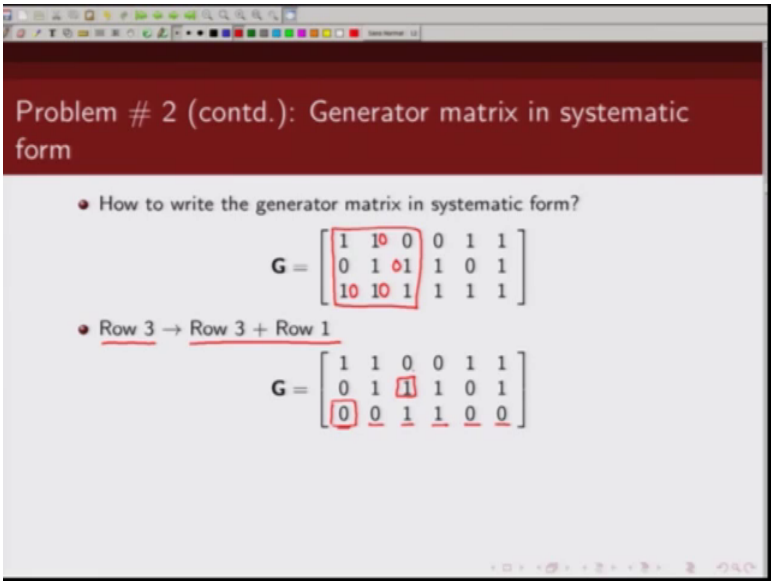
So we got a 0 here, right?

(Refer Slide Time 13:40)



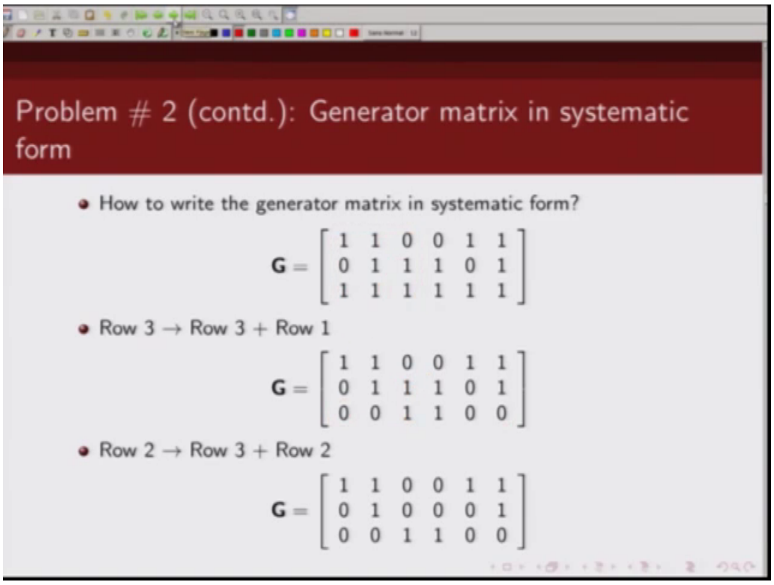
Next, we want a 0 here. We want

(Refer Slide Time 13:48)



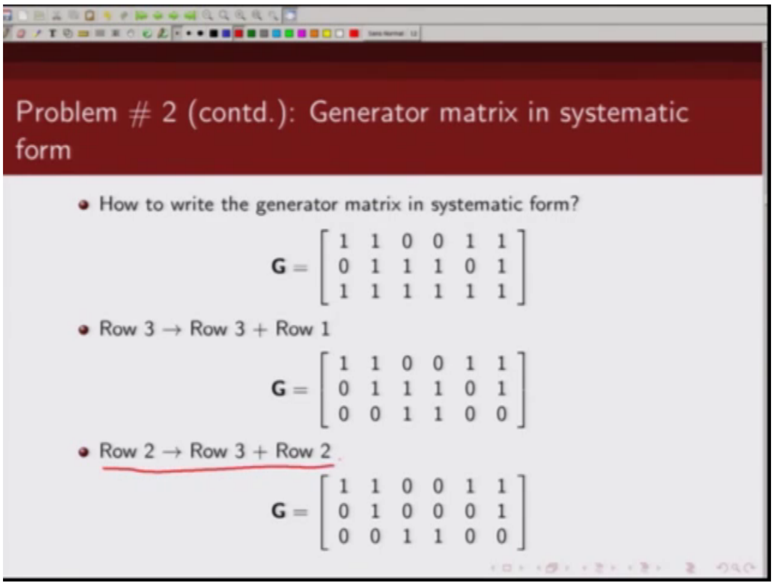
this; we want to make this 0. So how can we do that?

(Refer Slide Time 13:55)



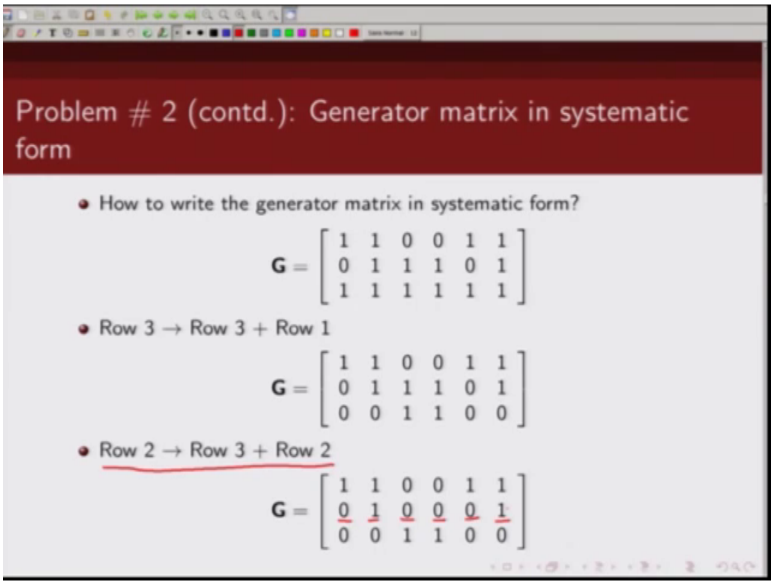
We do this transformation that row 2 is

(Refer Slide Time 14:00)



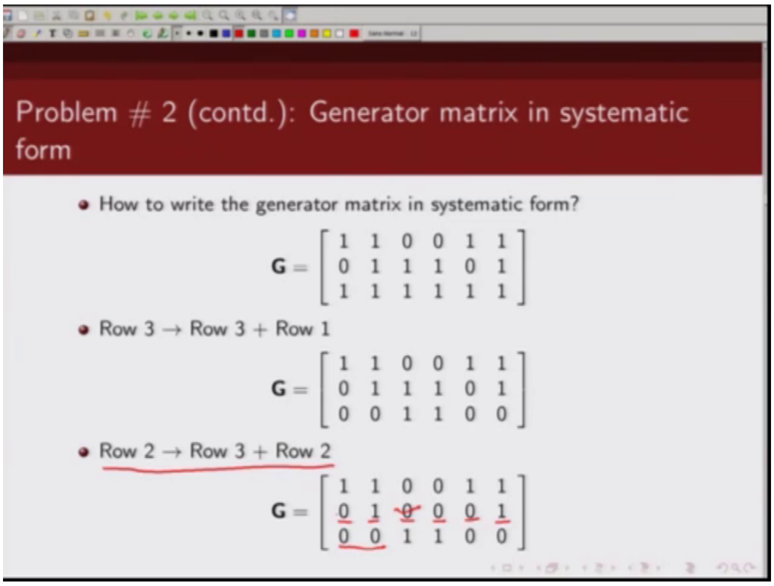
row 3 plus row 1, row 2. So if row 2 is row 2 plus row 3, then what's going to happen? This will remain 0, this will remain 1 but this 1 will become 0. So let's do that. So this is 0 plus 0 is 0. 1 plus 0 is 1, 1 plus 1 is 0, 1 plus 1 is 0, 0 plus 0 is 0, and 1 plus 0 is 1, Ok.

(Refer Slide Time 14:27)



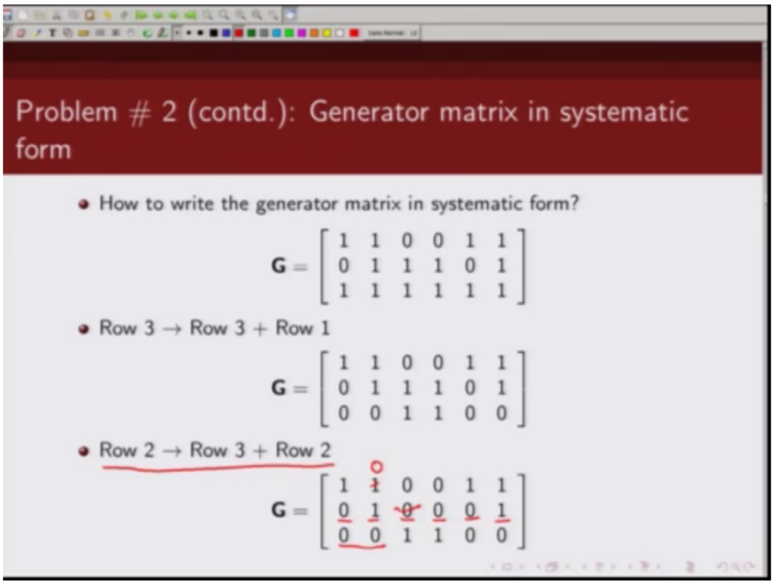
So we got these 0's, we got this 0, Ok now what do we have to do? We will have to get

(Refer Slide Time 14:35)



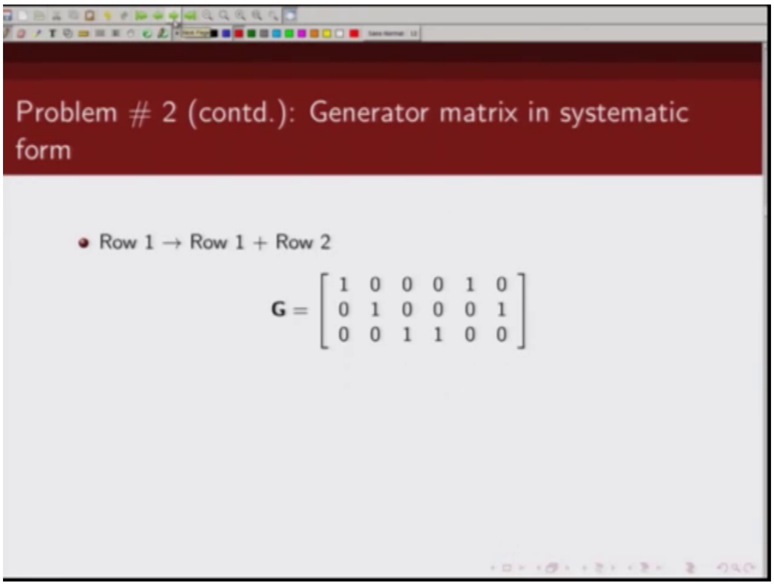
this; here we will have to get a 0.

(Refer Slide Time 14:38)



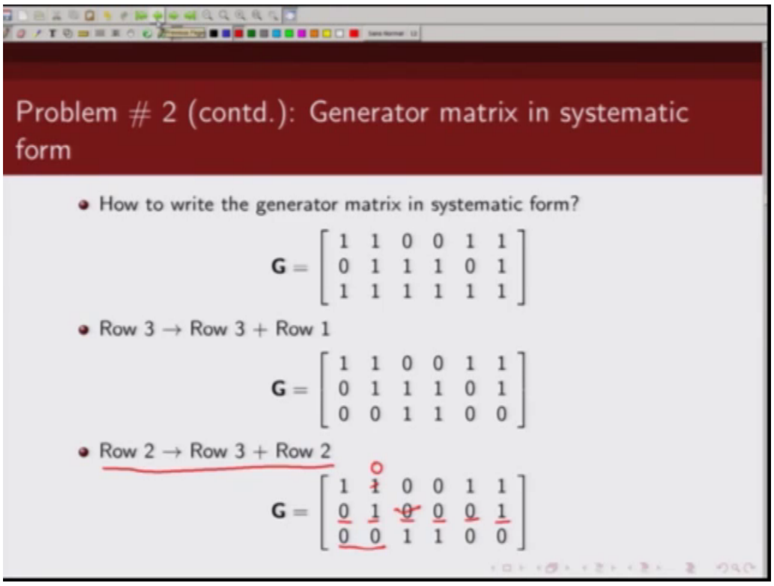
So how can we get a 0 here? We will do

(Refer Slide Time 14:44)



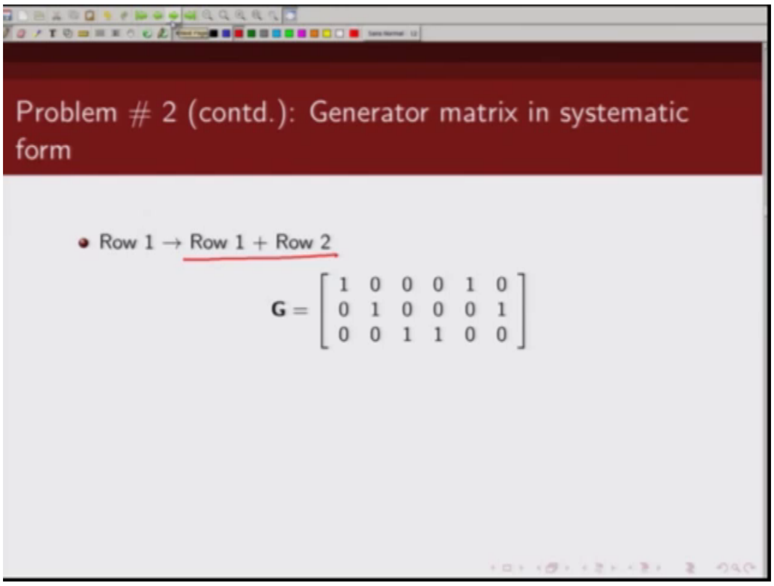
this transformation. We will add row 1 and row 2 and replace row 1 by this. So we are going to add these 2 rows.

(Refer Slide Time 14:53)



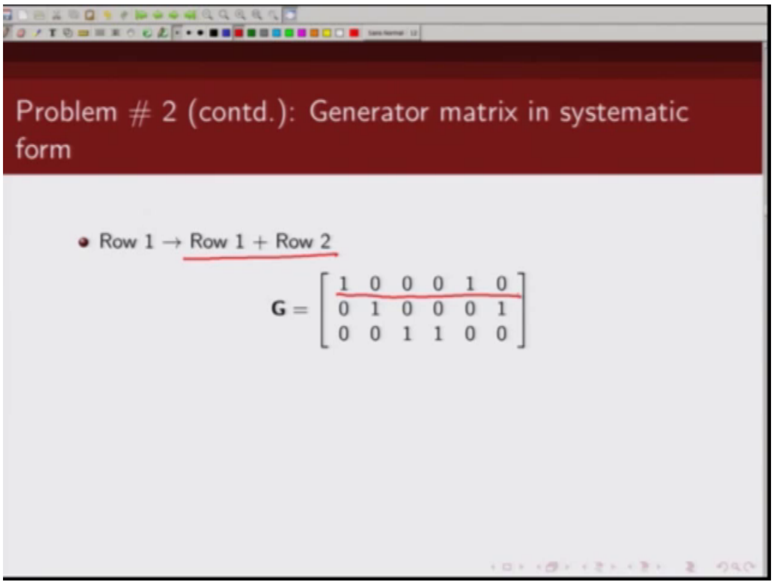
If we add these 2 rows, what's going to happen? This 1 will remain 1. 1 plus 1, this will become 0 and this will remain 0. This will be 0, this will be 1, and this will be 0. So if we do this transformation, what

(Refer Slide Time 15:11)



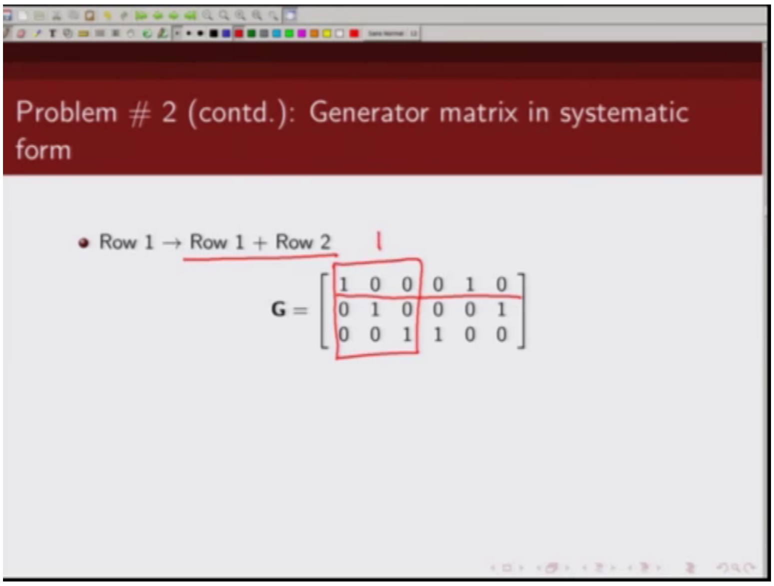
we get is this. Now

(Refer Slide Time 15:14)



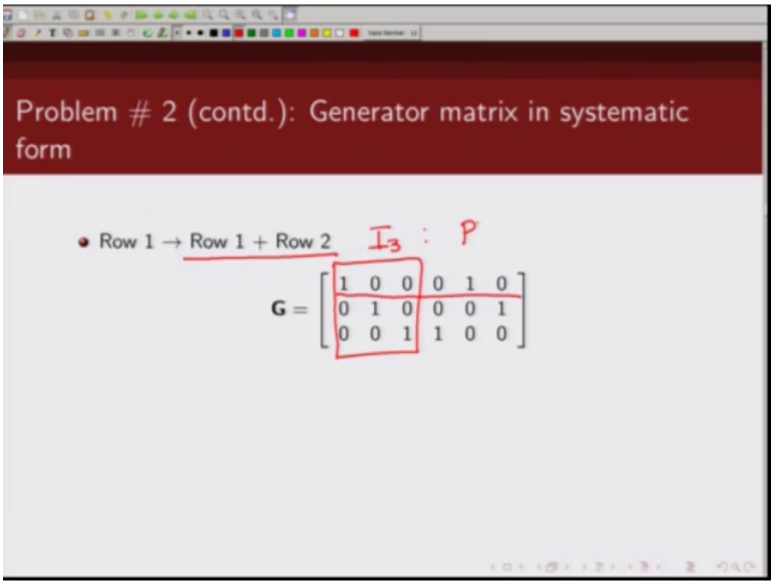
note that this is our identity matrix.

(Refer Slide Time 15:19)



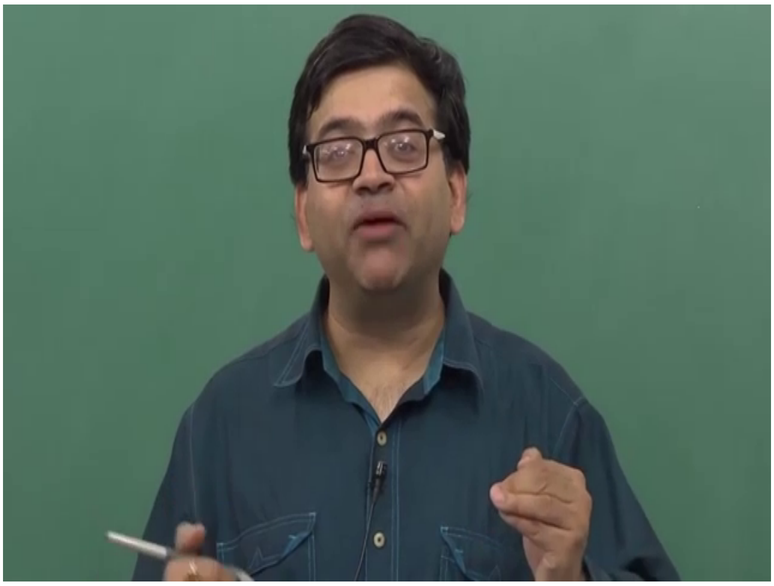
This is 3 cross 3 identity matrix and then this is your another matrix P, Ok. So

(Refer Slide Time 15:27)



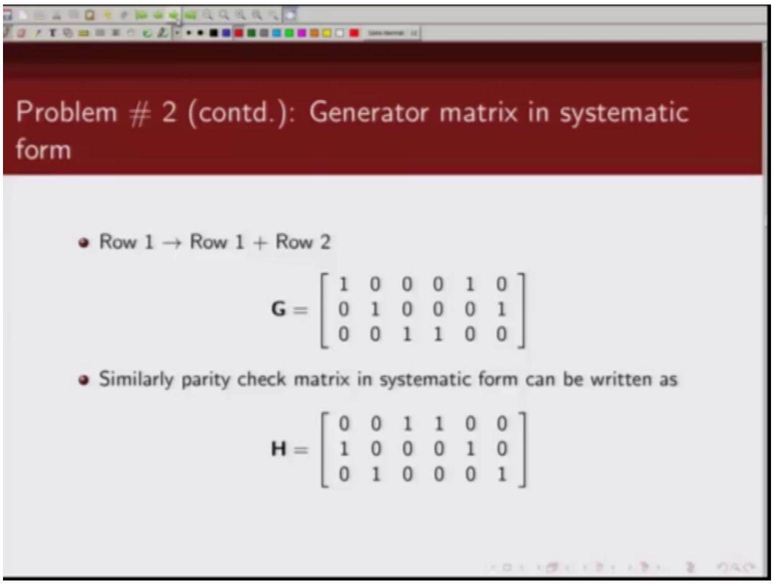
by doing elementary row

(Refer Slide Time 15:29)



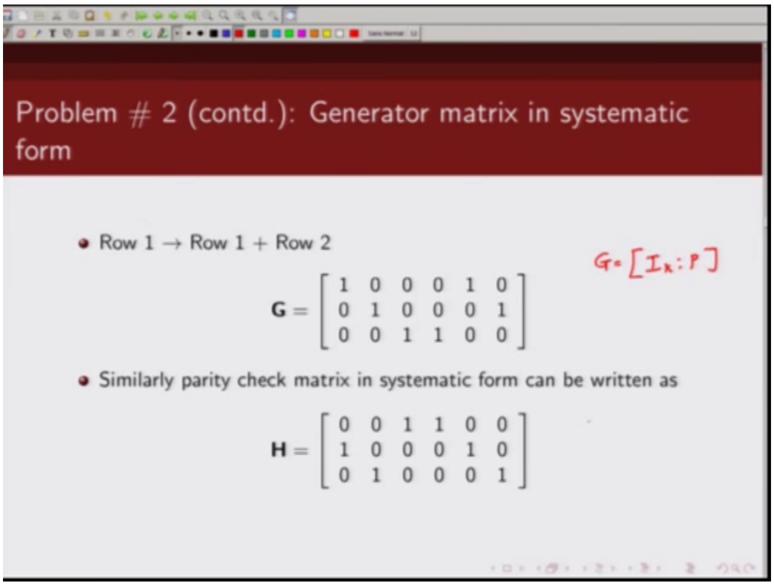
operation, we are able to get our generator matrix in a systematic form. And

(Refer Slide Time 15:37)



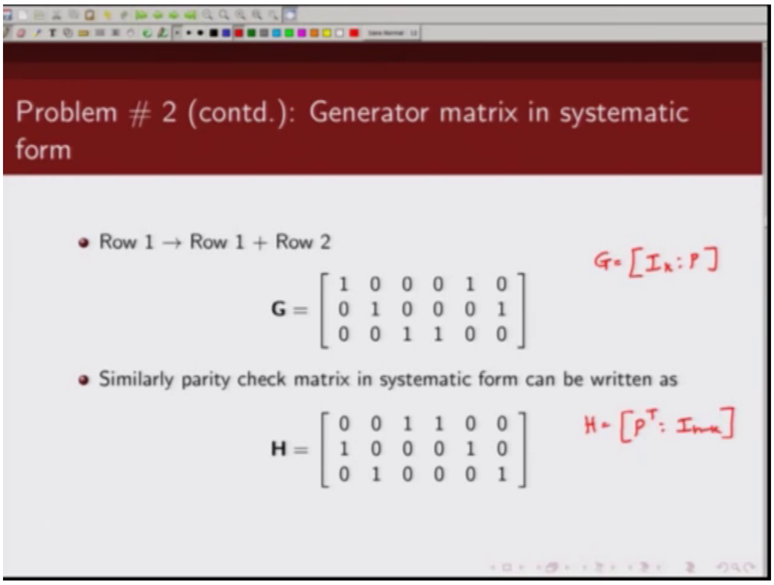
if we have a generator matrix in a systematic form we can very easily find the parity check matrix in systematic form. So this is like I k P then this H matrix

(Refer Slide Time 15:50)



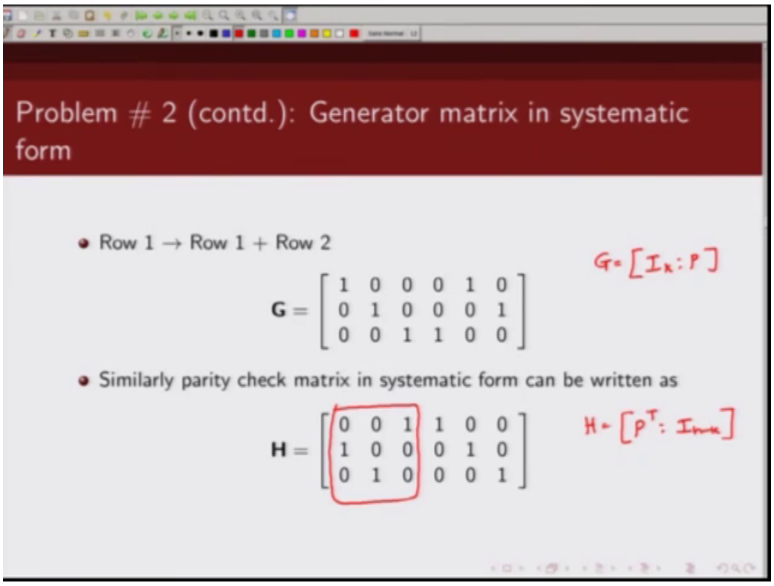
will be P transpose I n minus k.

(Refer Slide Time 15:56)



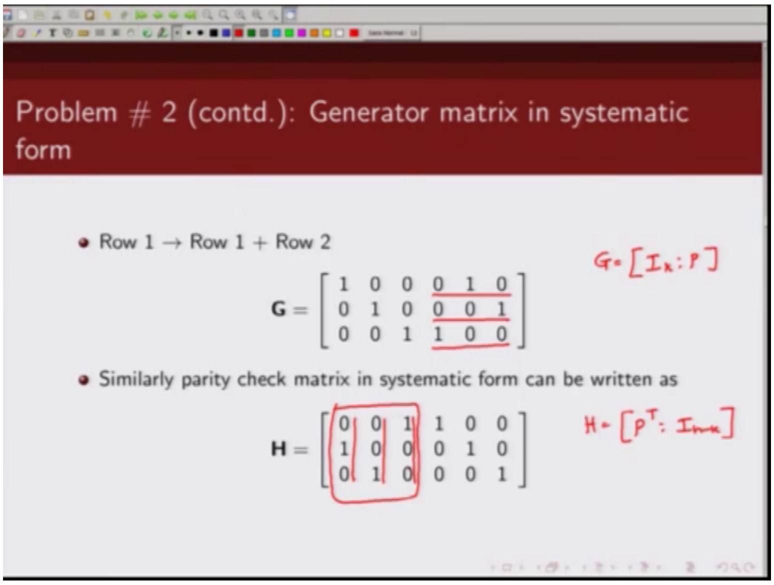
So this, this is basically your P transpose. So this

(Refer Slide Time 16:03)



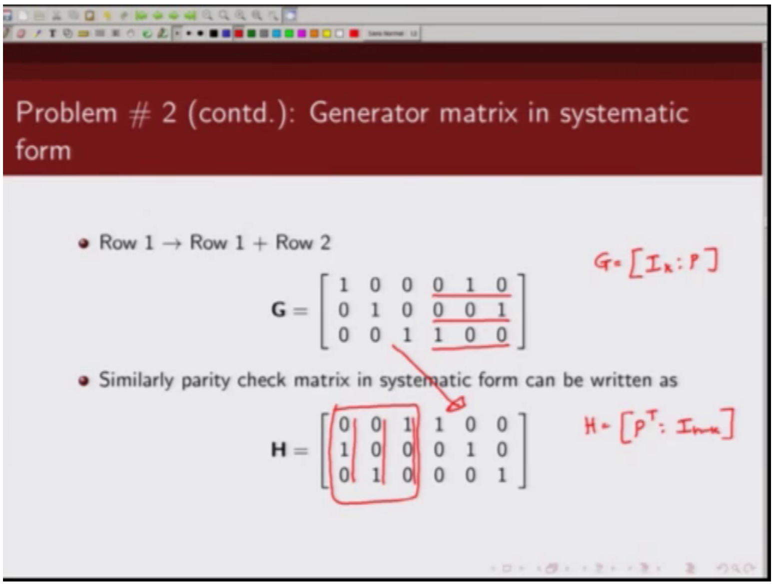
is 0 1 0, this will come here, 0 1 0, 0 0 1, this is 0 0 1. And 1 0 0 is this, 1 0 0.

(Refer Slide Time 16:13)



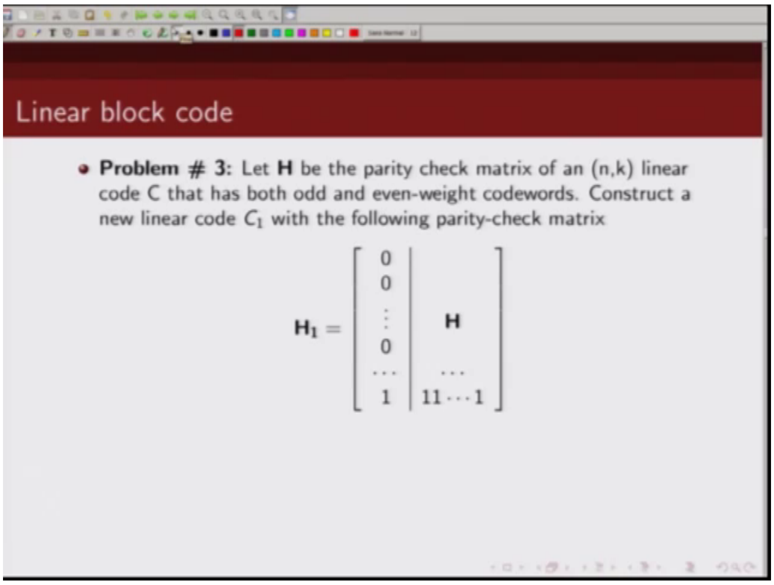
And then you have this identity matrix which is here,

(Refer Slide Time 16:18)



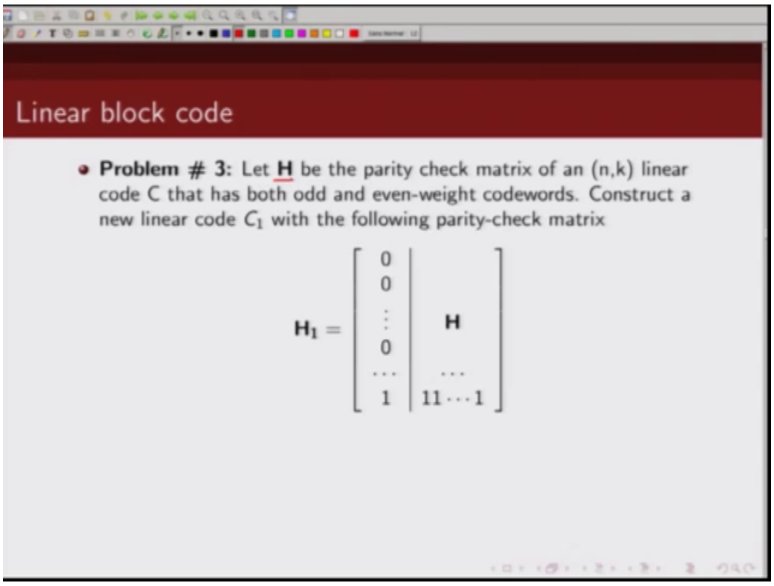
Ok. Next

(Refer Slide Time 16:23)



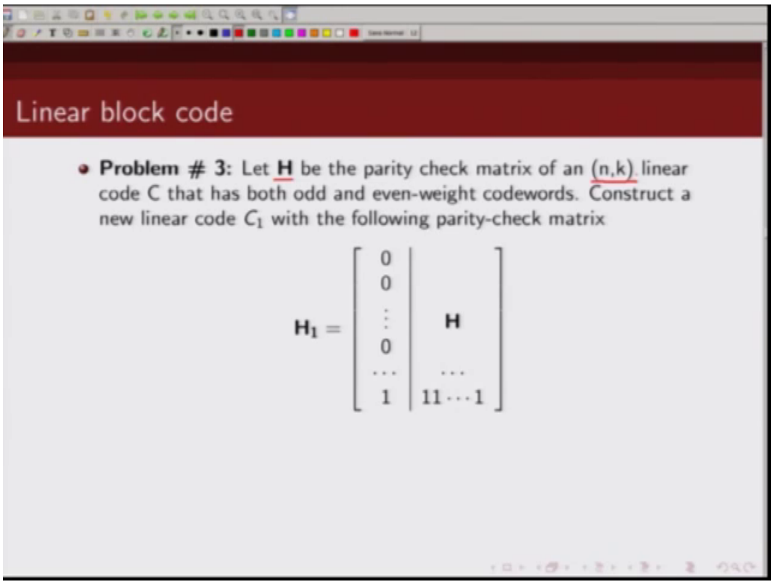
we are given a parity check matrix H of a linear

(Refer Slide Time 16:30)



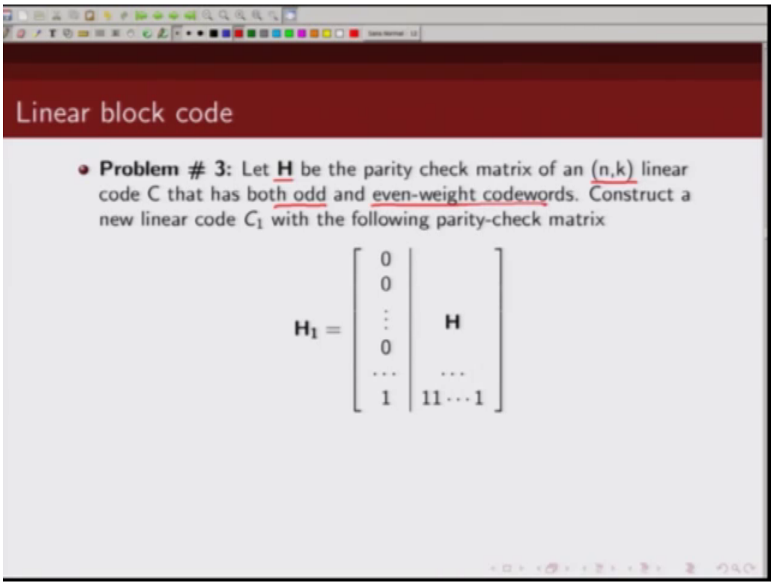
block code

(Refer Slide Time 16:33)



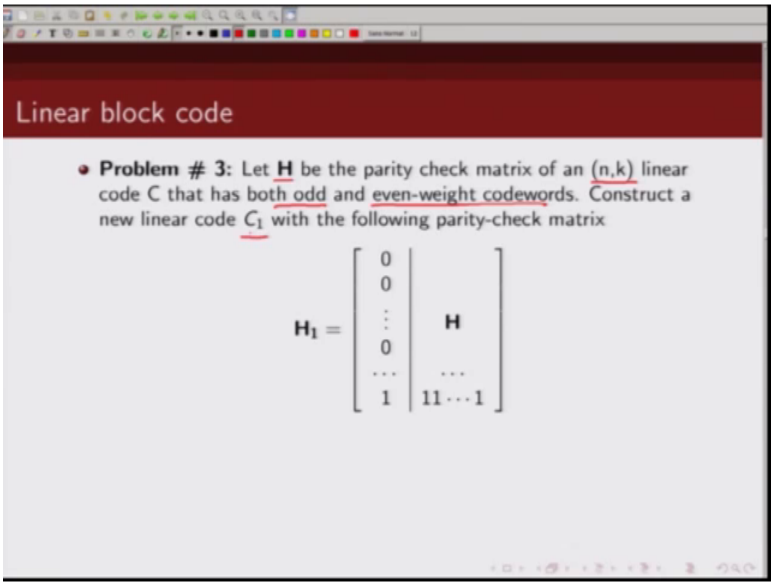
with parameter n and k. And it is given that this code C has both odd weight codewords and even weight codewords. In

(Refer Slide Time 16:44)



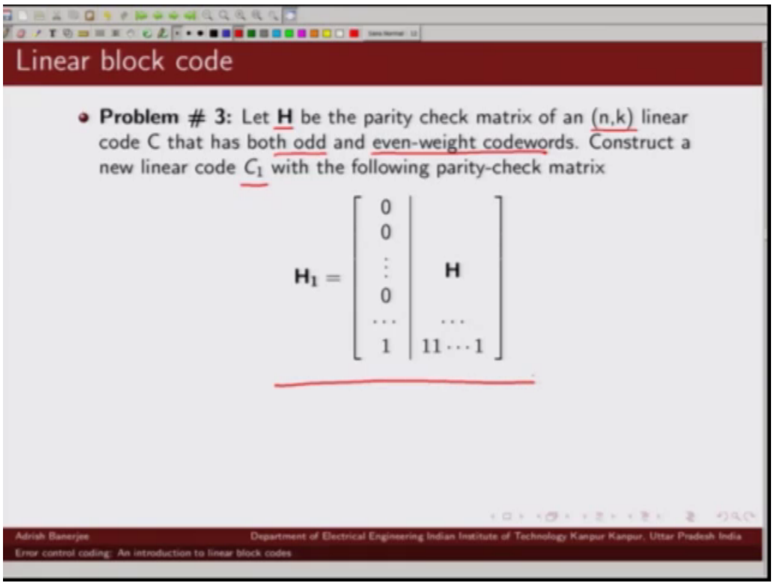
other words, the number of 1's in the codewords, it contains both odd number of 1's as well as even number of 1's. And we are constructing a new code that we are calling as C 1

(Refer Slide Time 16:59)



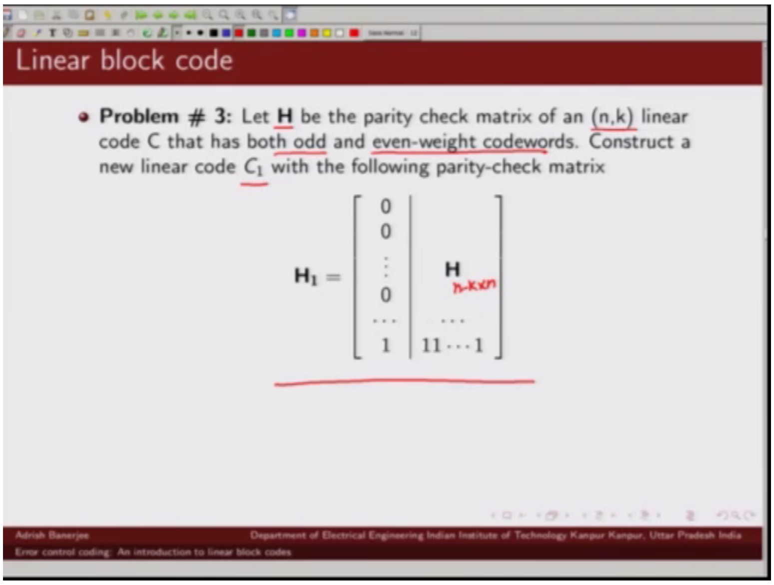
and the parity check matrix of the new code C 1 is given by this.

(Refer Slide Time 17:09)



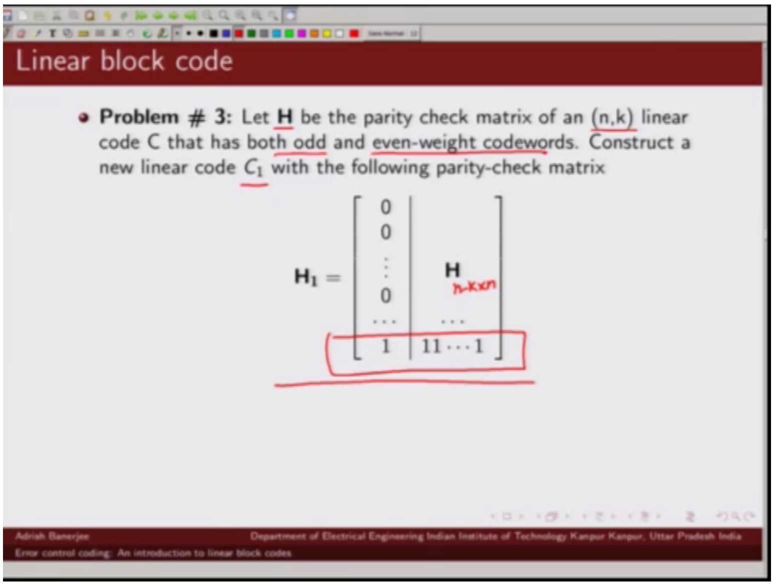
So how do we find this new matrix, parity check matrix H 1? We are adding a new column which is 0 in the initial rows except in the last row which is a 1 and here we have put our original n minus k cross n matrix. And the last row

(Refer Slide Time 17:35)



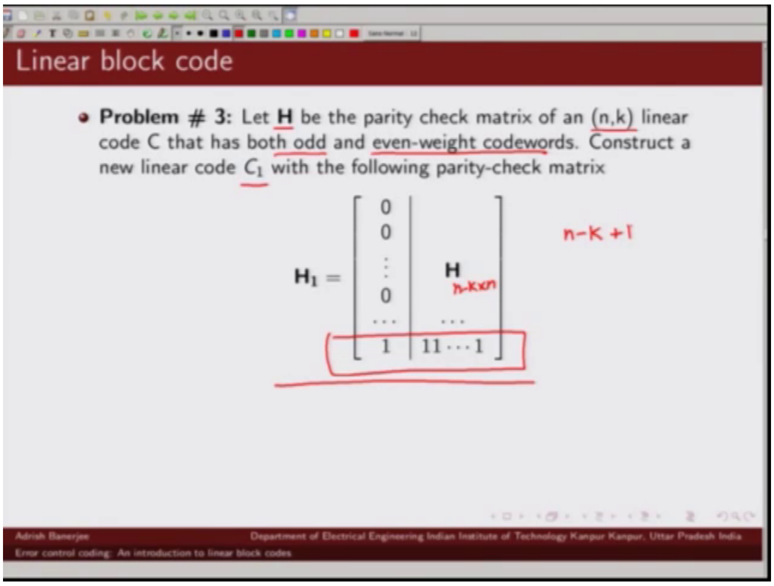
is basically all 1's,

(Refer Slide Time 17:38)



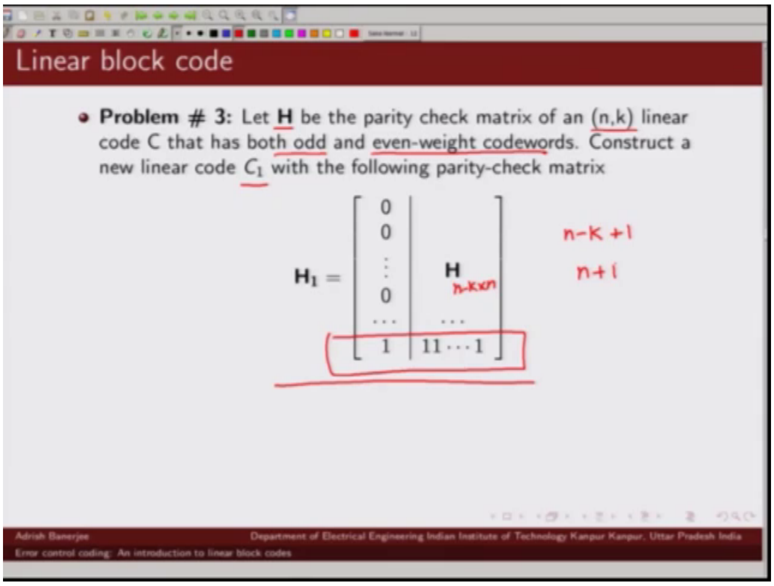
Ok. So the dimension of this matrix is, so number of rows is n minus k plus 1 and number of columns are

(Refer Slide Time 17:51)

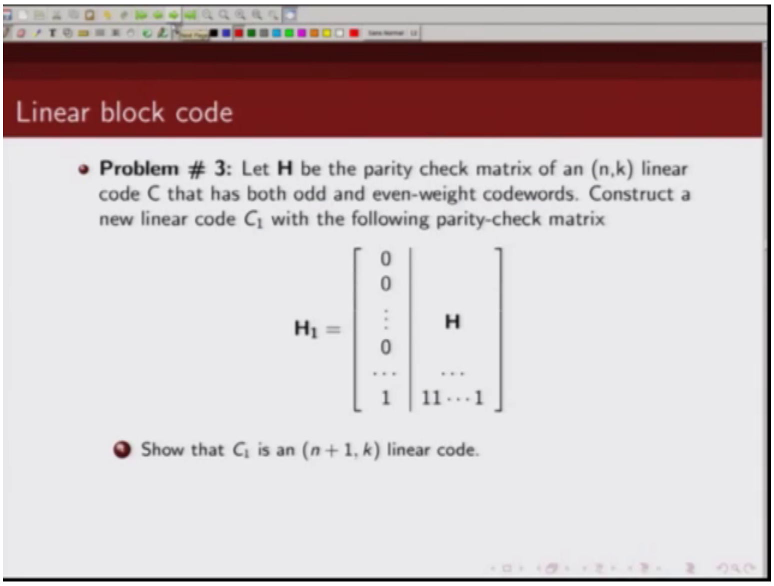


n plus 1

(Refer Slide Time 17:56)

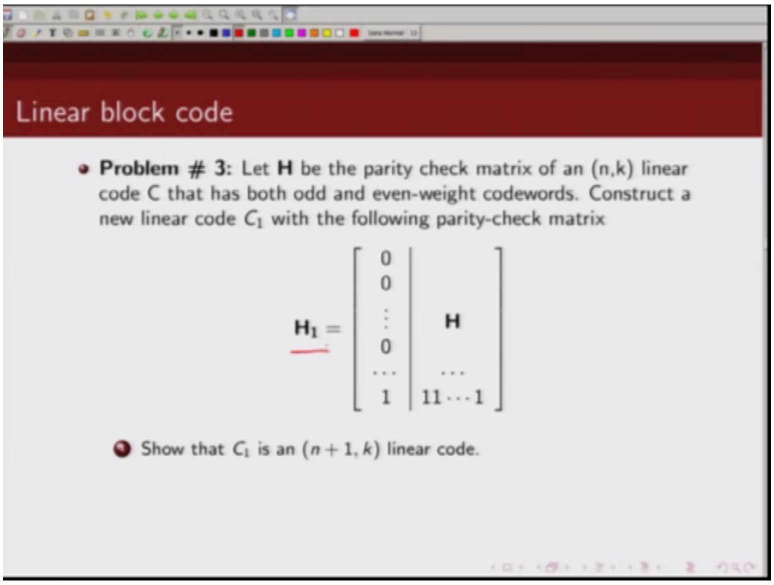


(Refer Slide Time 17:57)



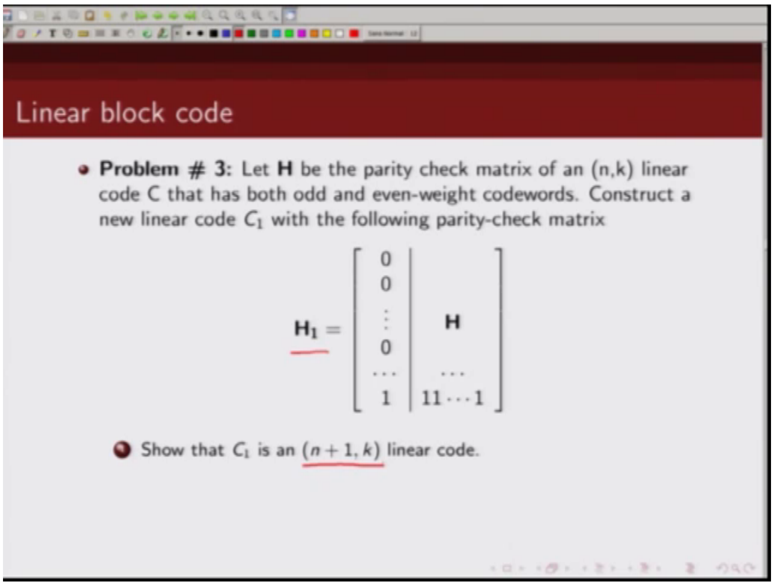
Now you are asked to show that the code generated by this

(Refer Slide Time 18:03)



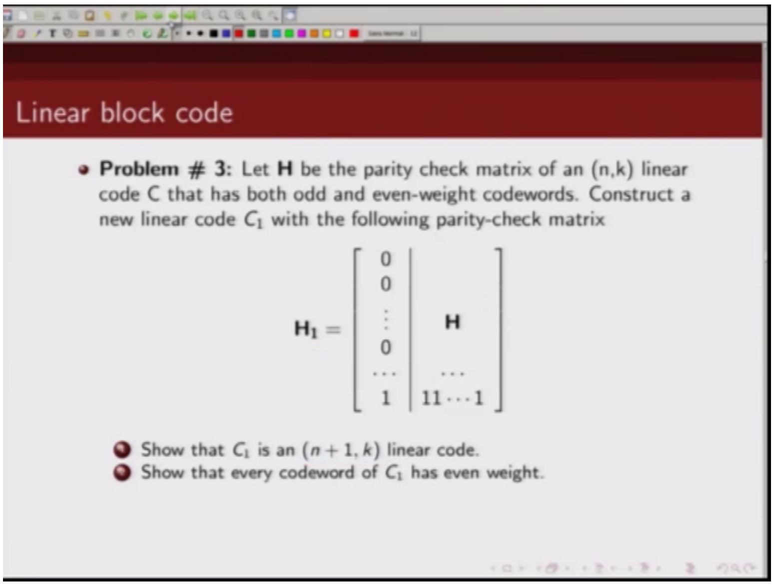
parity check matrix H 1 is a linear code with parameters

(Refer Slide Time 18:10)



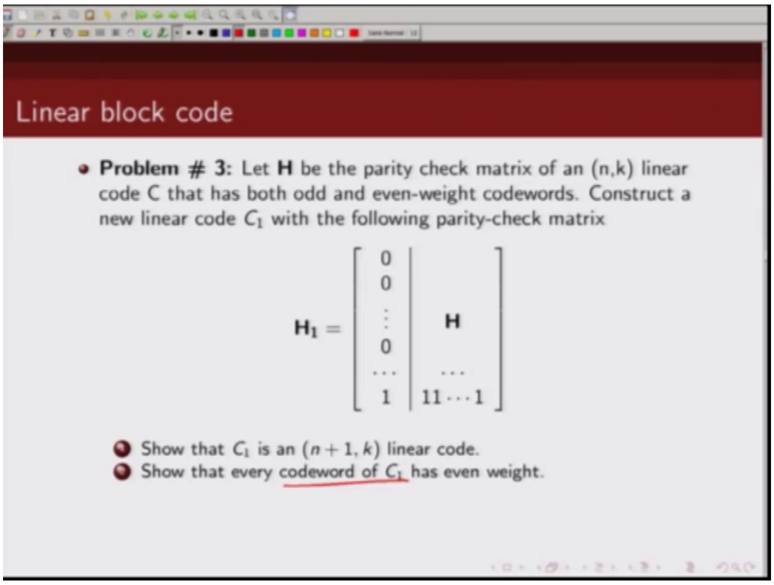
n plus 1 and k. Second thing you are

(Refer Slide Time 18:16)



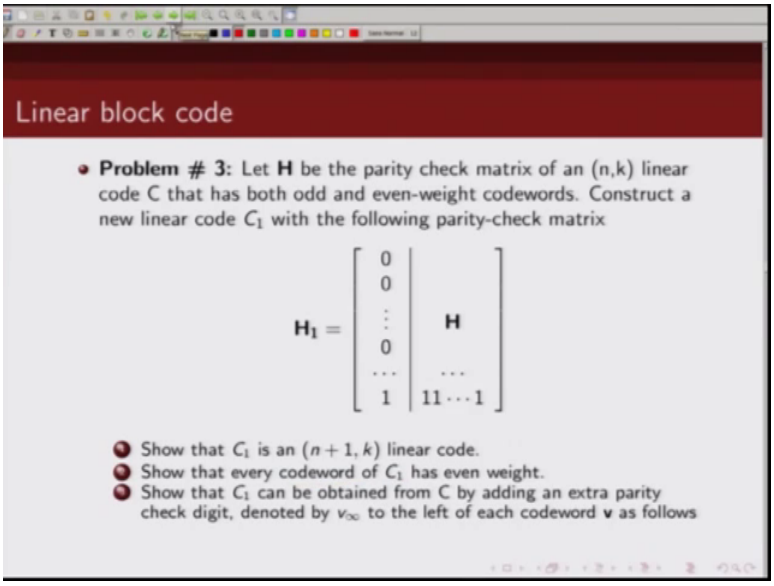
asked to prove is that all the codewords

(Refer Slide Time 18:22)



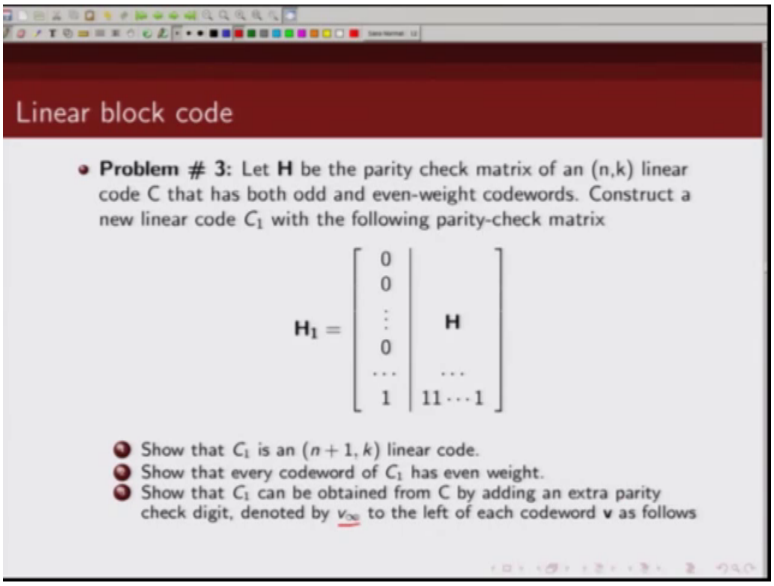
of this new code C 1 will have even weight. That means they will have even number of 1's in them.

(Refer Slide Time 18:32)



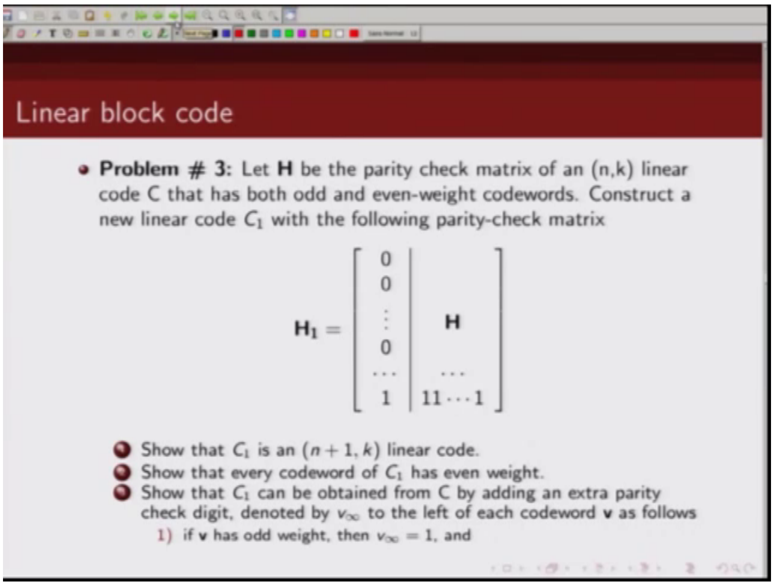
The third thing you have to prove is, this new code C 1 is obtained from old code C by adding an additional parity bit which we are denoting by v infinity

(Refer Slide Time 18:45)



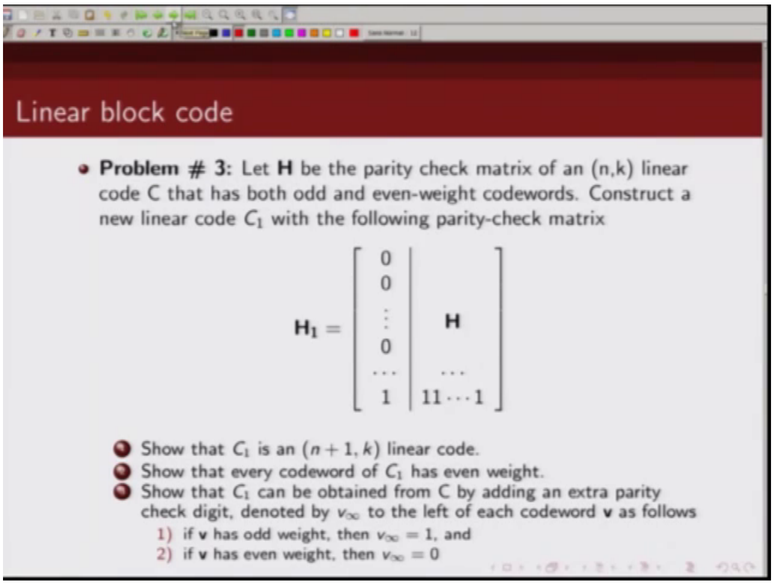
to the left of this codeword and how do you select this parity bit v infinity?

(Refer Slide Time 18:50)



If the original codeword has odd weight, then you put v infinity as 1 otherwise

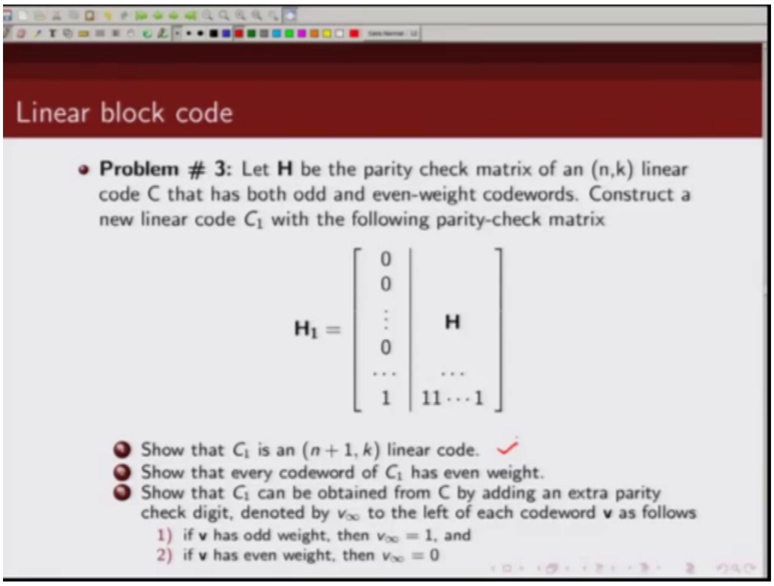
(Refer Slide Time 18:58)



if the original codeword has even weight then you put this v infinity as 0.

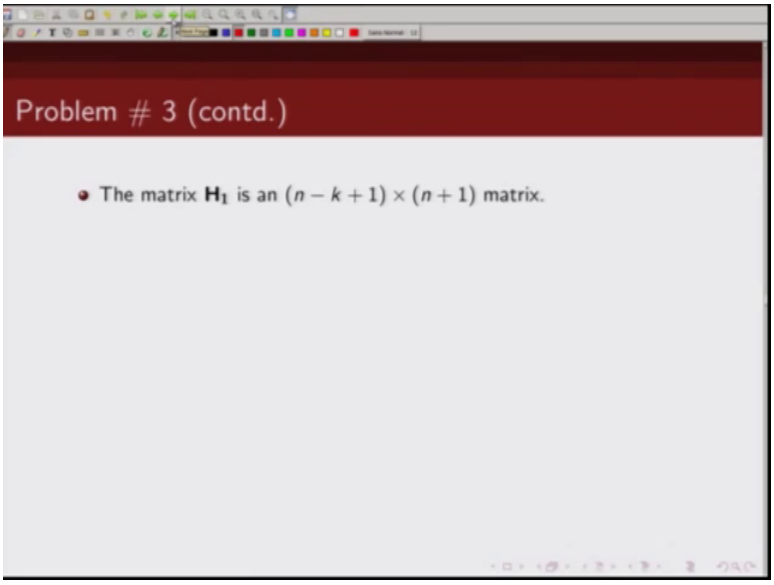
So let's prove one by one. Let's first prove this that code generated by this new parity check matrix

(Refer Slide Time 19:14)



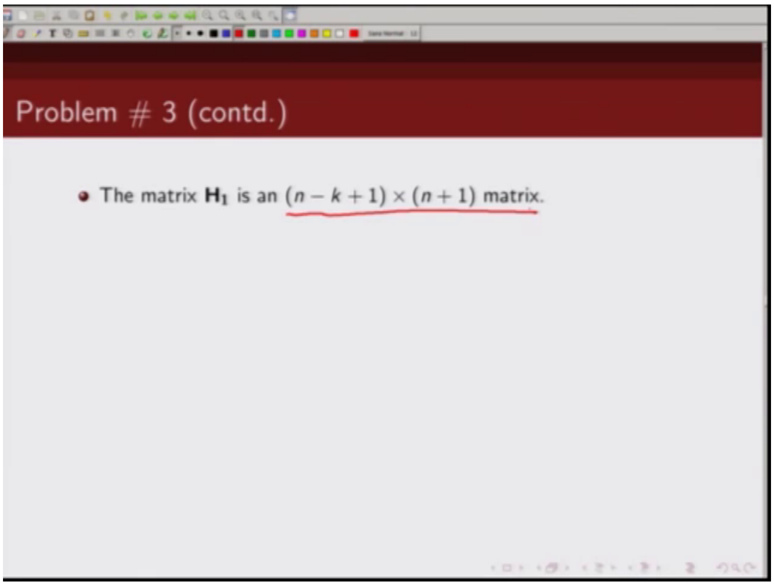
is basically a new code with n given by n plus 1 and k given by k. So

(Refer Slide Time 19:24)



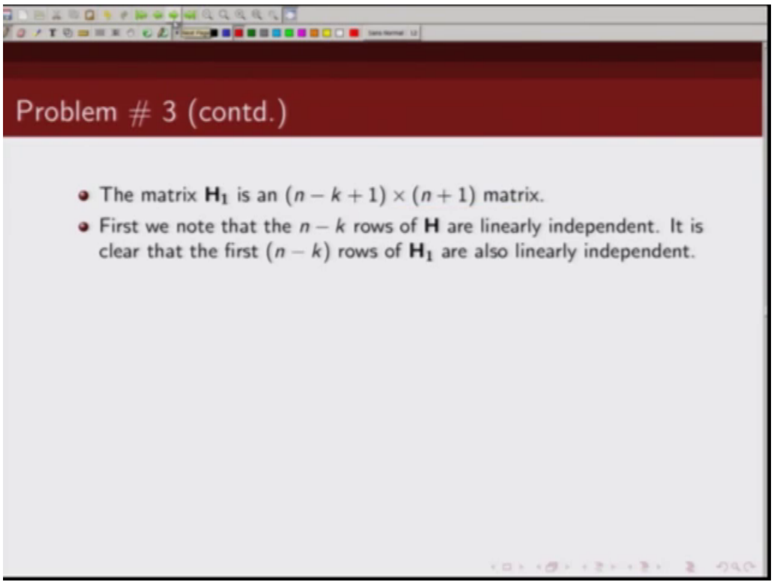
as we know that this H matrix has these dimensions because we are adding

(Refer Slide Time 19:30)



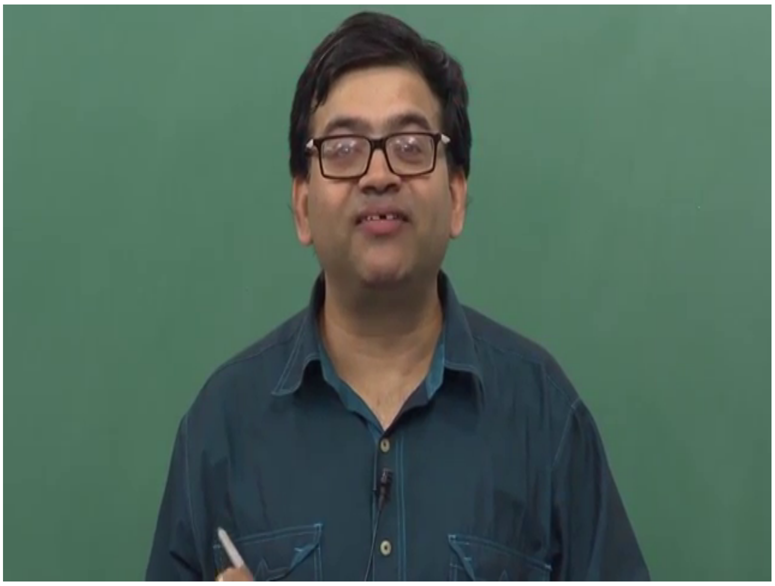
a new column and we are adding a new row. Next,

(Refer Slide Time 19:36)



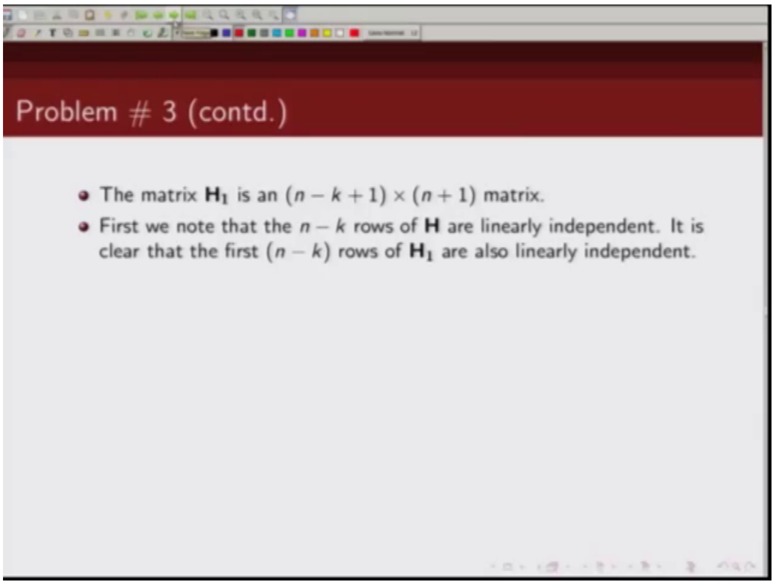
now what is the rank of the original matrix H? The rank of the original matrix H is

(Refer Slide Time 19:44)



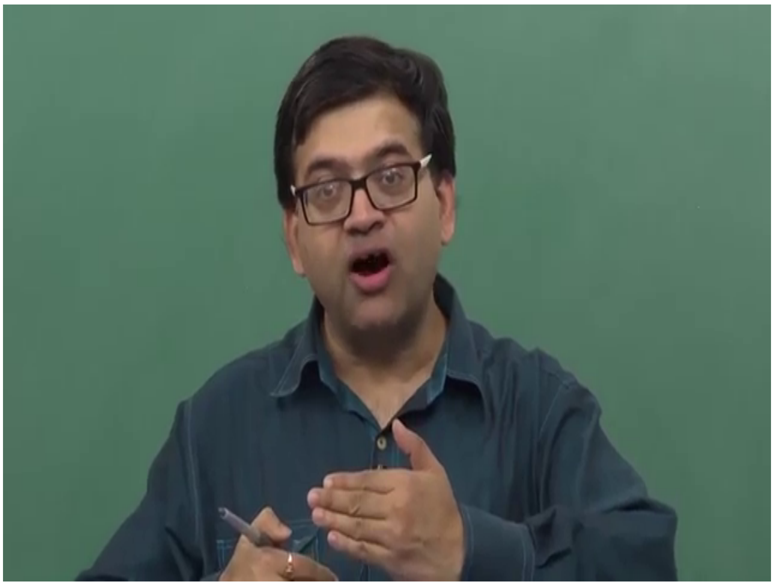
n minus k. That means

(Refer Slide Time 19:48)



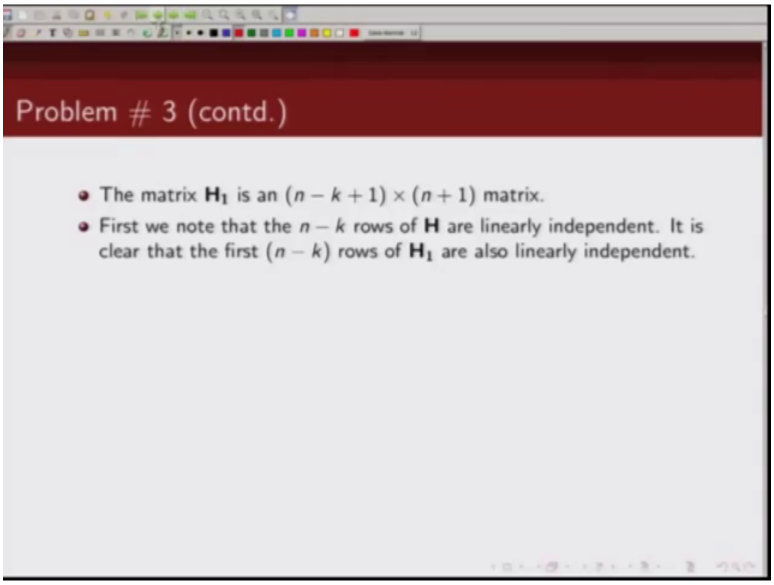
the n minus k rows of the original

(Refer Slide Time 19:51)

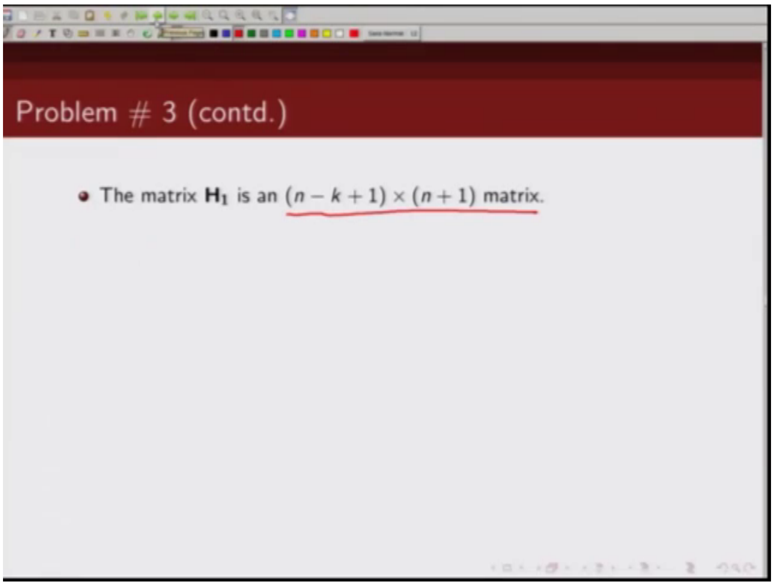


parity check matrix H are linearly independent, Ok. Now go back

(Refer Slide Time 19:58)

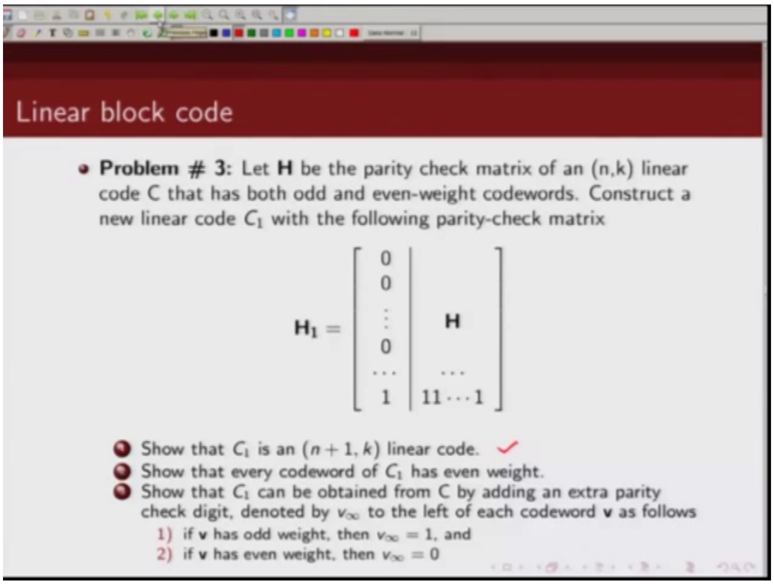


(Refer Slide Time 19:59)



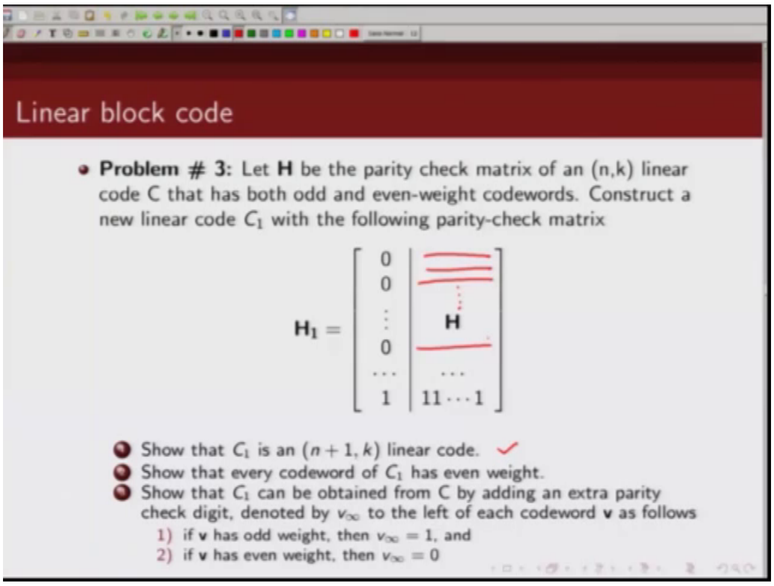
and look at the new

(Refer Slide Time 20:01)



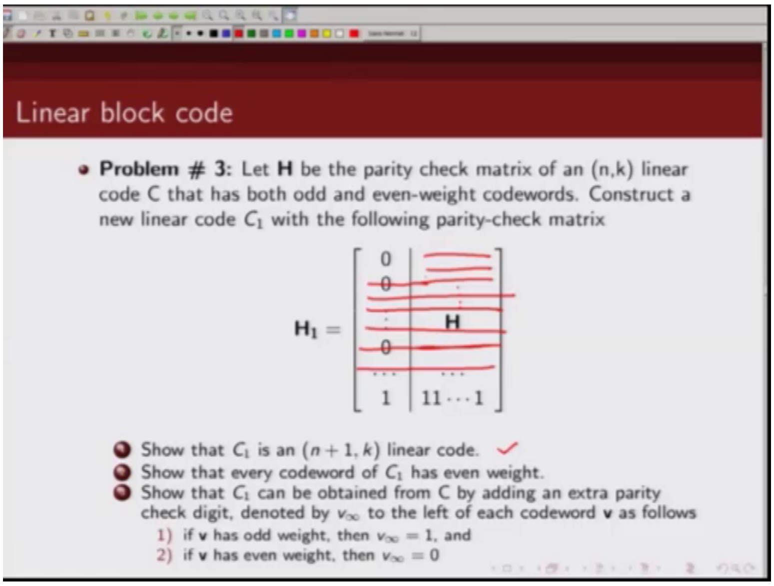
construction. So these n minus k rows are linearly independent.

(Refer Slide Time 20:08)



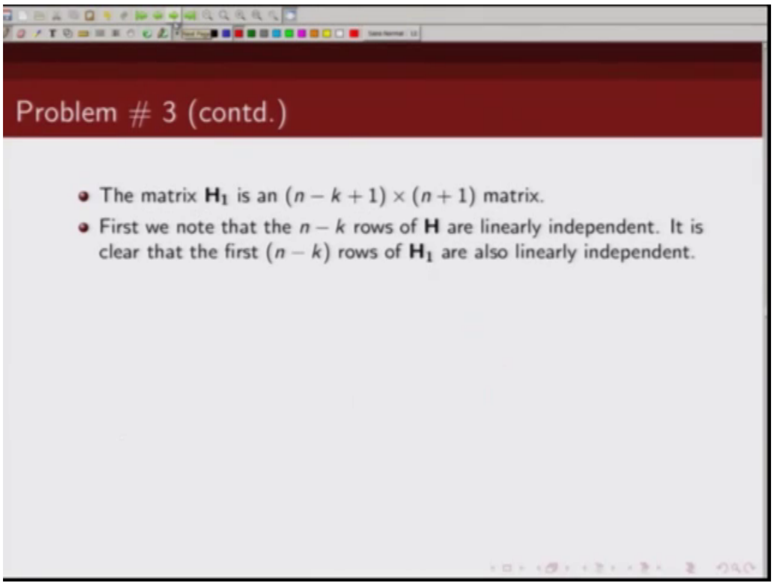
And what have we added here? We have added 0 here. So these new rows, these new n minus k rows will also be

(Refer Slide Time 20:20)



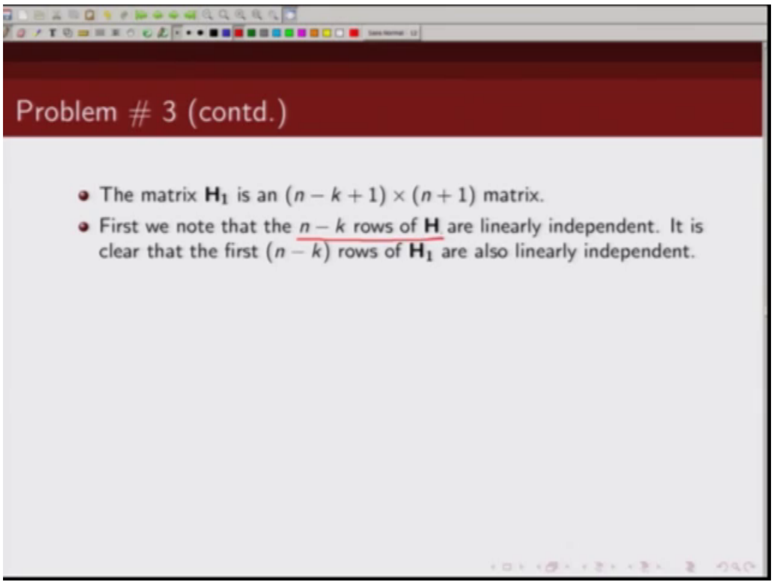
linearly independent.

(Refer Slide Time 20:24)



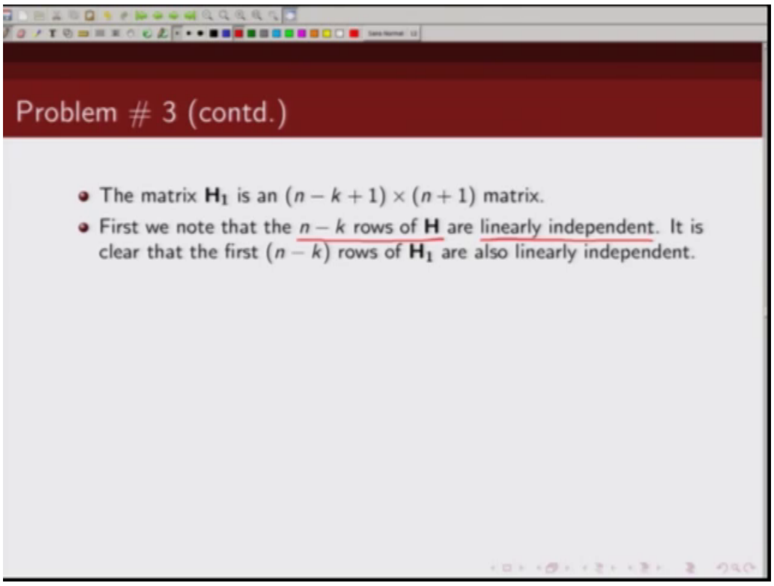
So that's what we are saying that since n minus k

(Refer Slide Time 20:29)



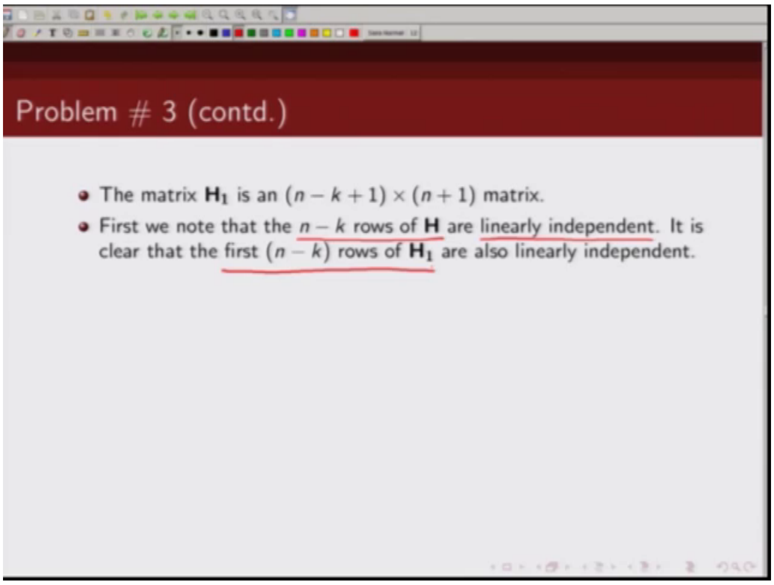
rows of the original parity check matrix H are linearly independent, so

(Refer Slide Time 20:36)



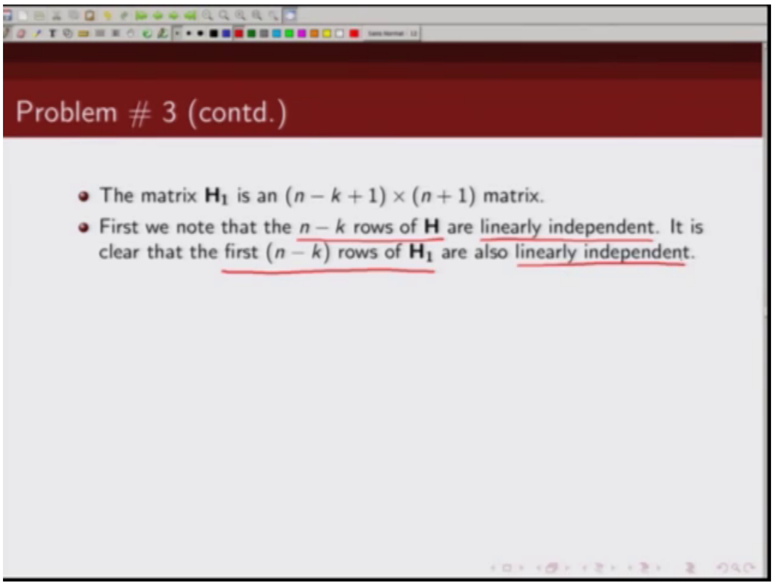
the first n minus k

(Refer Slide Time 20:38)



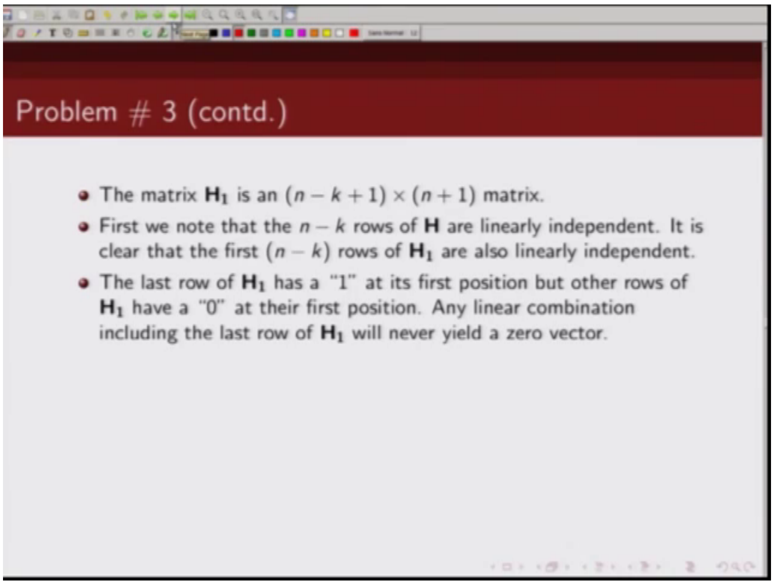
rows of the original parity check matrix H 1 will also be linearly independent.

(Refer Slide Time 20:45)



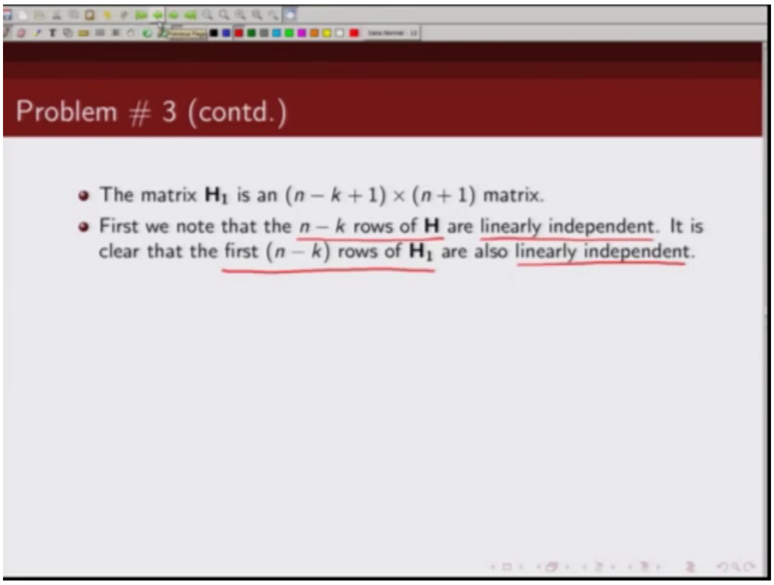
Now let's look at

(Refer Slide Time 20:49)

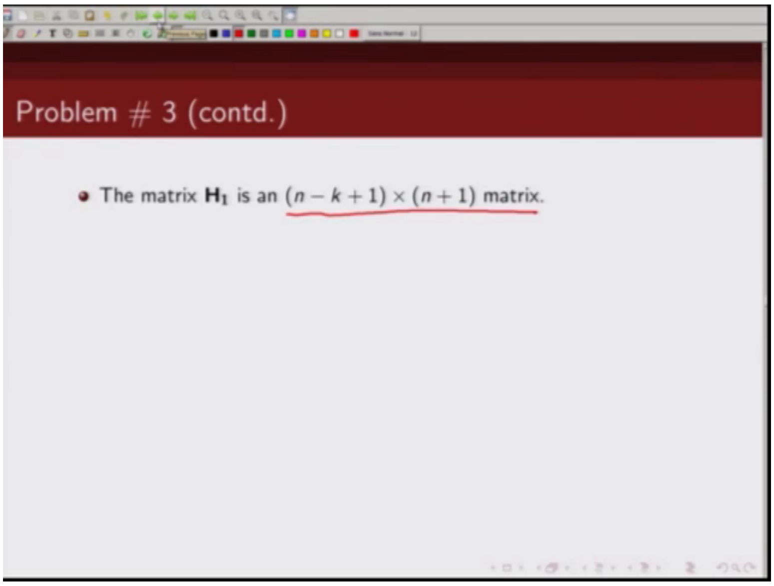


the last row of this new parity check

(Refer Slide Time 20:54)

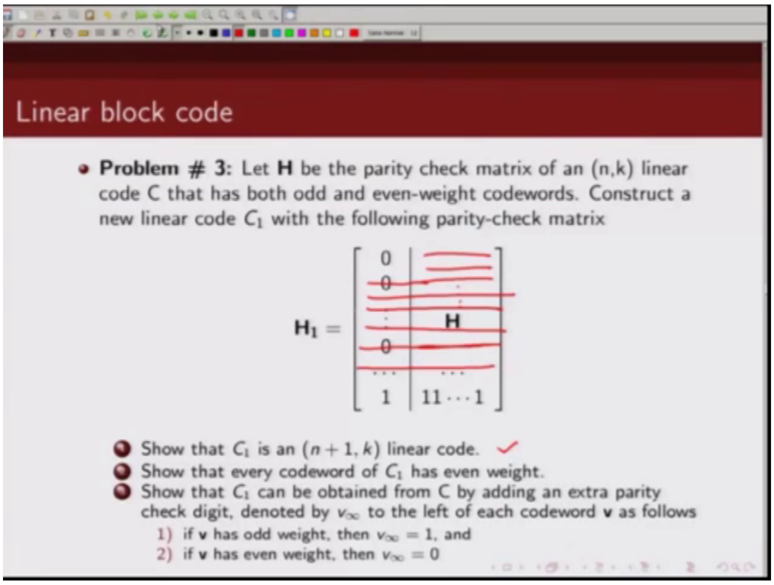


(Refer Slide Time 20:55)



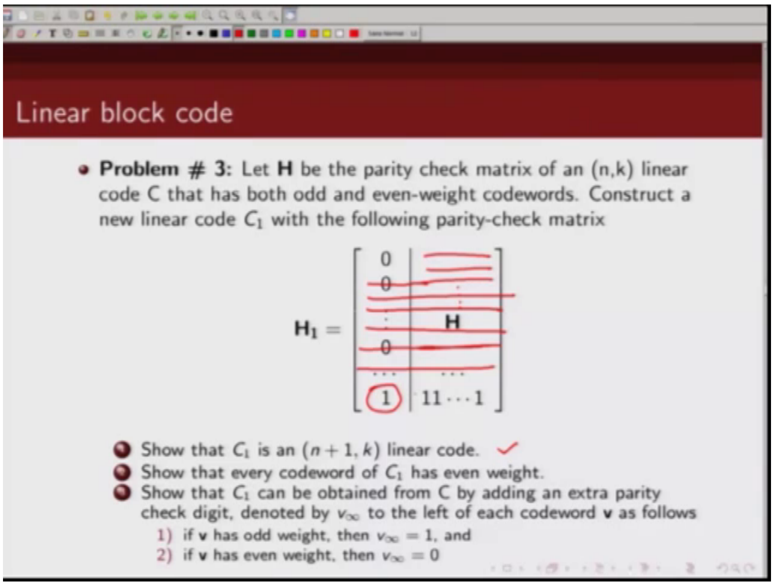
matrix H 1.

(Refer Slide Time 20:56)



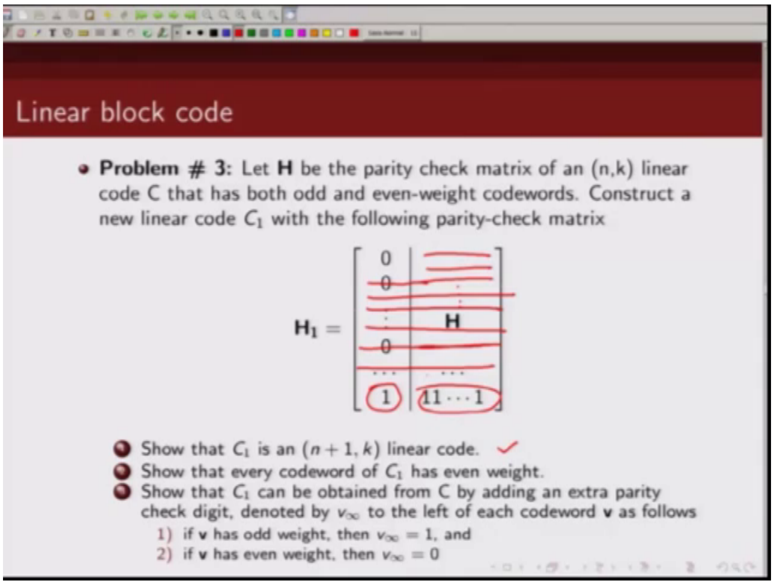
Note that we have 1 here. And these are all

(Refer Slide Time 21:02)



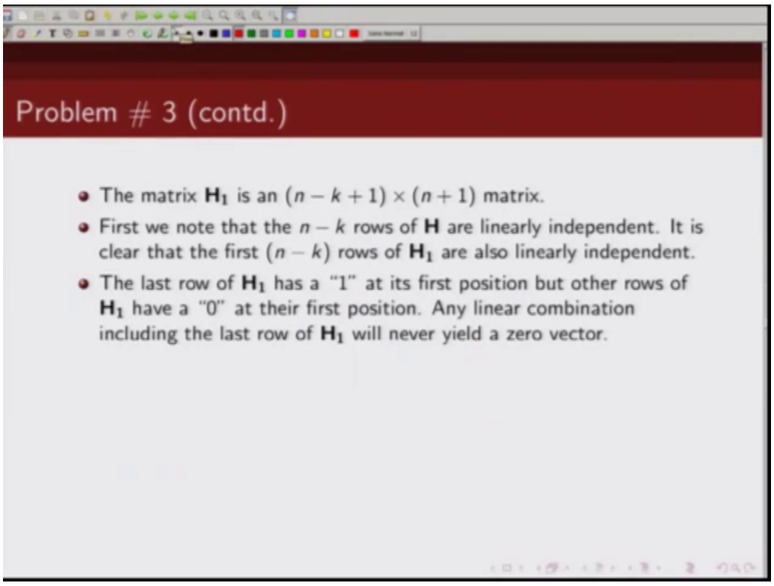
all 1's here.

(Refer Slide Time 21:04)



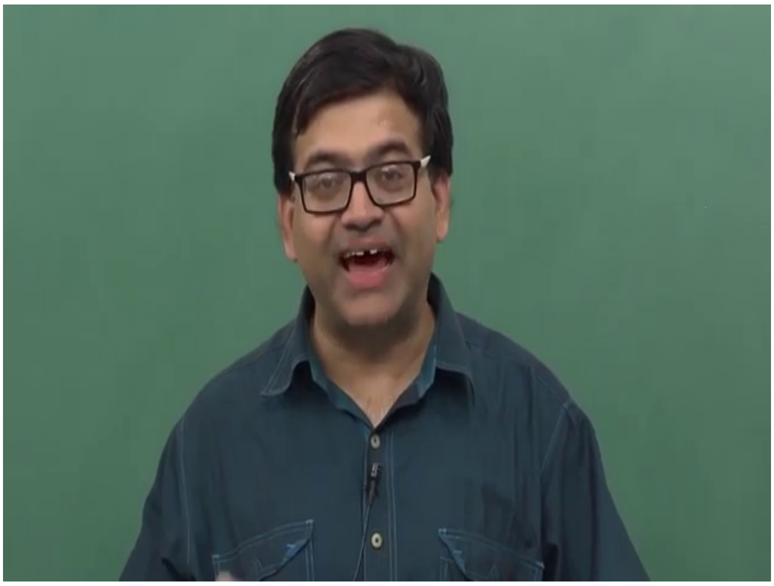
Whereas here, all of these are 0's. So this new row will also be linearly independent from any of the other rows of this parity check matrix H 1.

(Refer Slide Time 21:20)



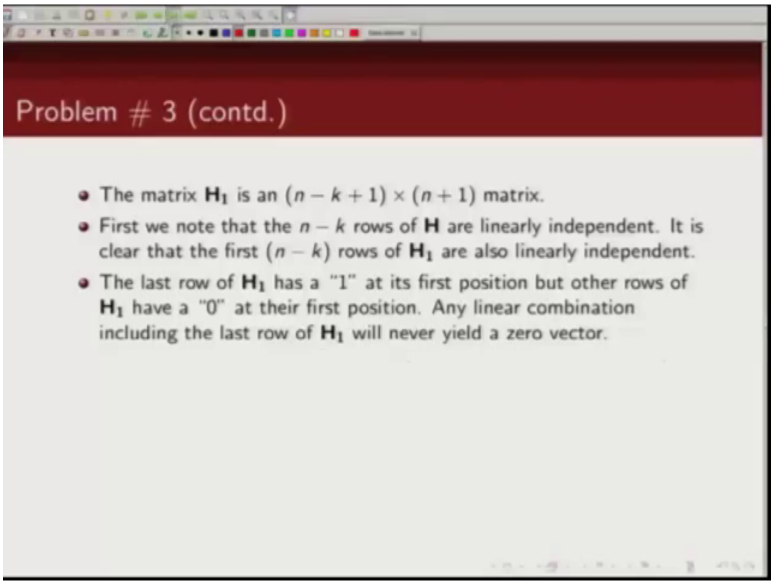
So any linear combination including the last row of H 1 will never result in a all zero vector. So what does it mean? It means that

(Refer Slide Time 21:32)



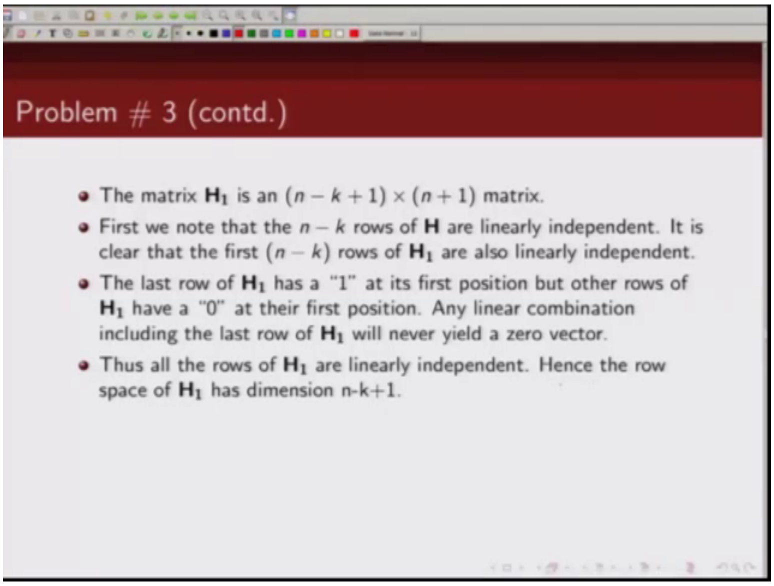
n minus k plus one rows of our new parity check matrix H one are linearly independent.

(Refer Slide Time 21:44)



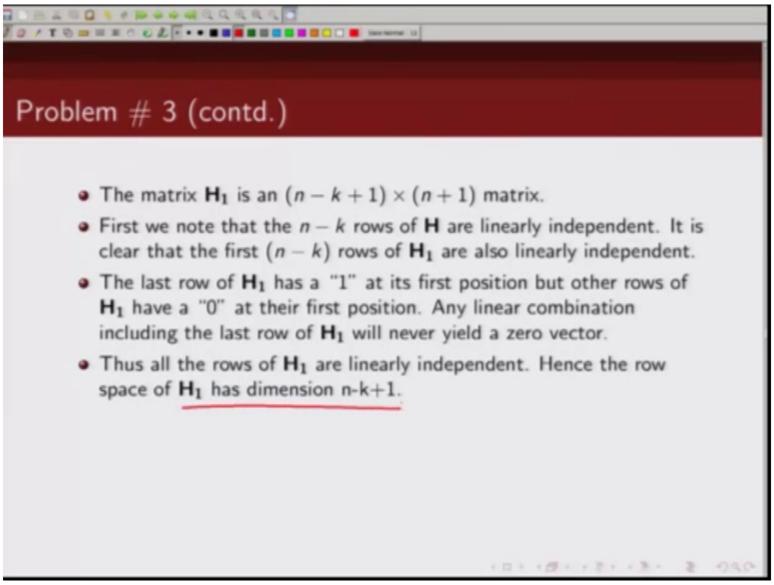
Hence

(Refer Slide Time 21:46)



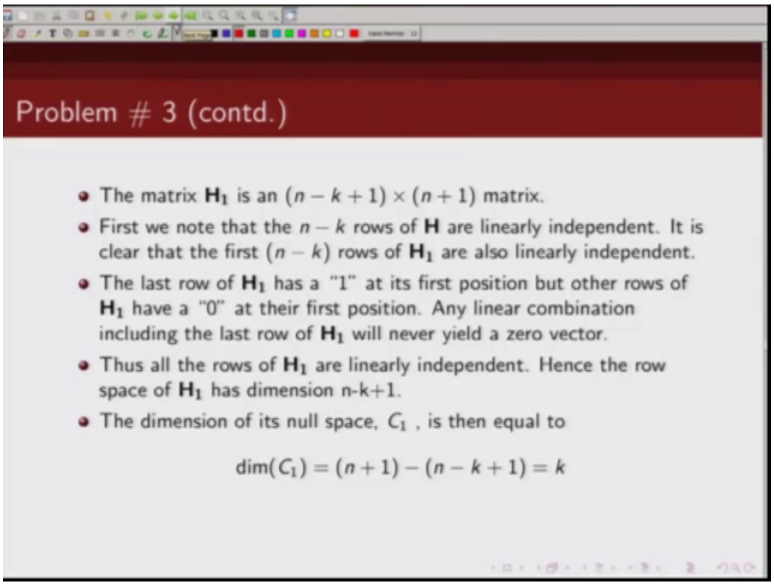
the dimension of H 1 is

(Refer Slide Time 21:49)



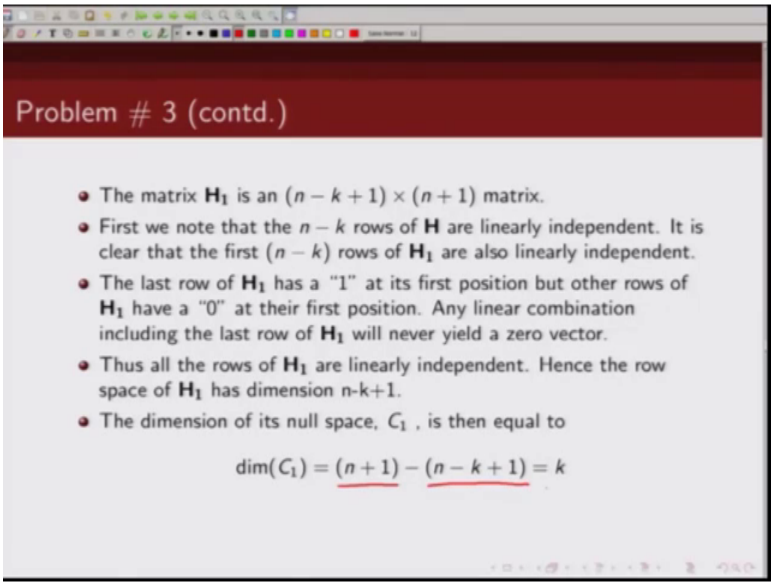
n minus k plus 1. Now how do we find

(Refer Slide Time 21:55)



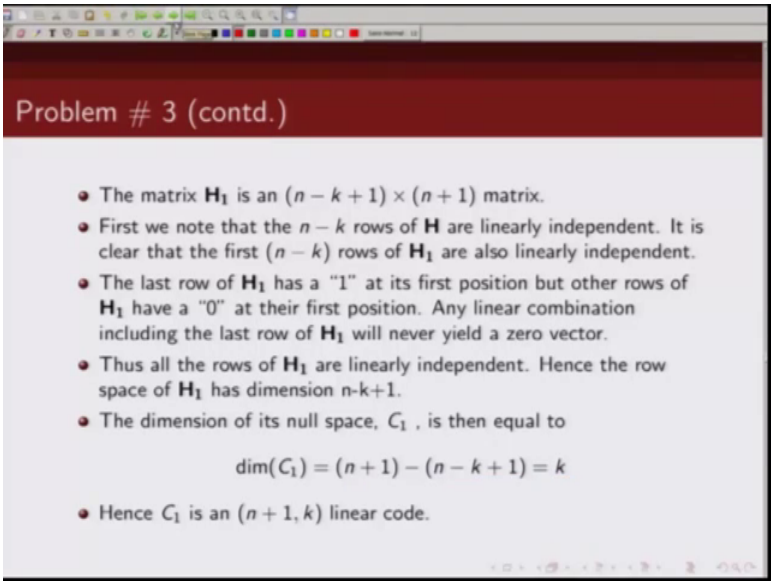
the dimension of basically, the null space of this parity check matrix H 1? It is given by, so number of columns is n plus 1. The dimension of H 1 is given by this.

(Refer Slide Time 22:10)



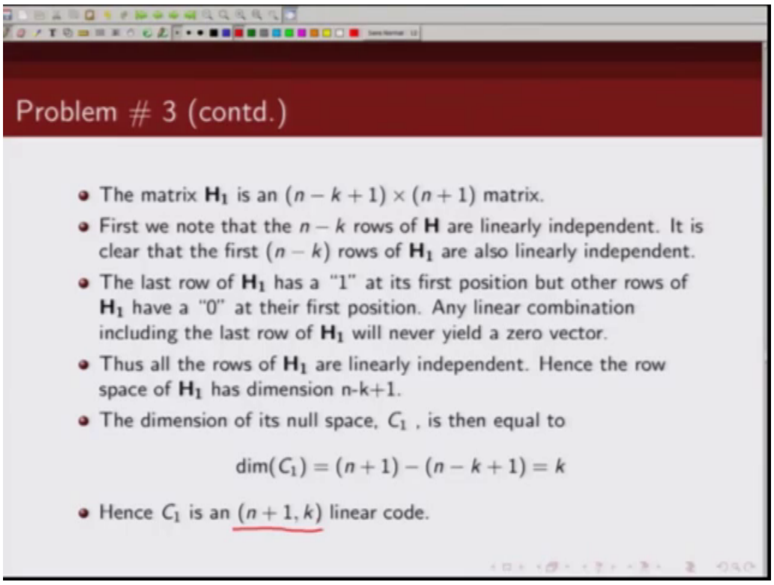
So this is the dimension of the null space of this parity check matrix. So then basically number of information bits is then k and number of coded bits is n plus 1. So this proves that C 1 is an

(Refer Slide Time 22:28)



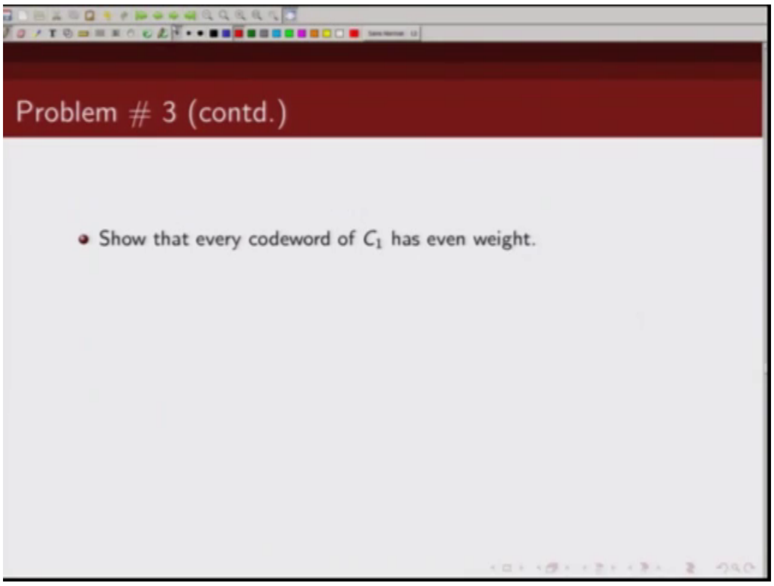
n plus 1 k linear code.

(Refer Slide Time 22:31)



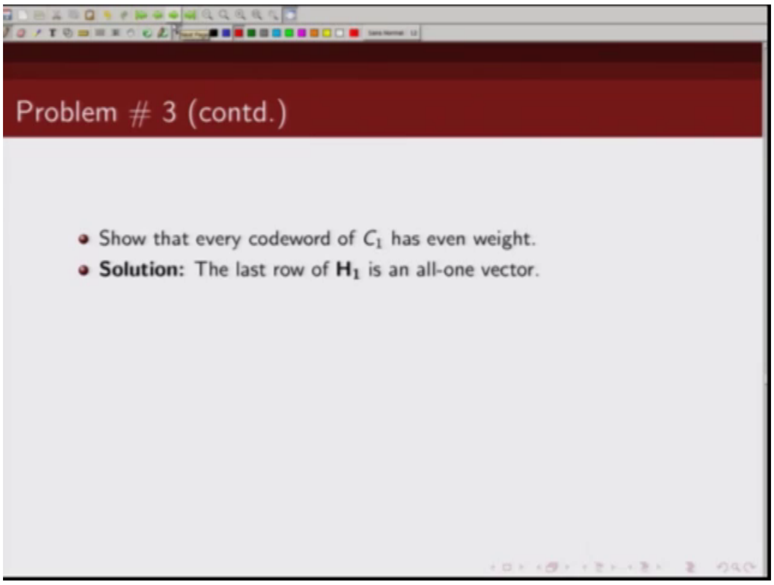
Next we are going to show is

(Refer Slide Time 22:36)



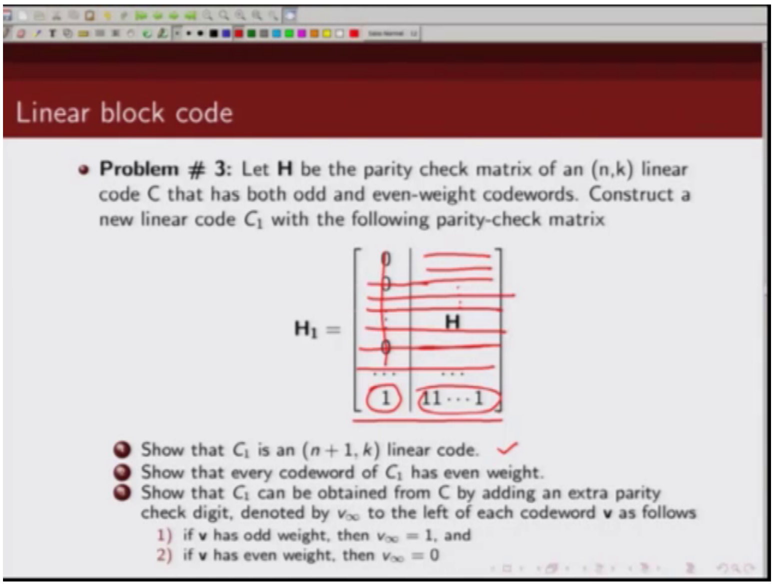
every codeword of C 1 has even weight. So how do we prove this?

(Refer Slide Time 22:44)



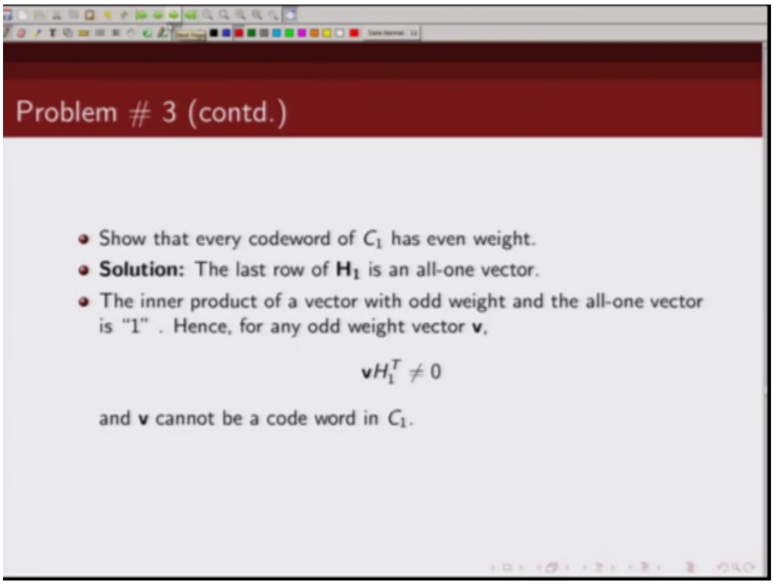
Please note that the last row of this parity check matrix H 1 contain all 1 vector. If you go back, recall the last row of this parity check matrix has all 1's.

(Refer Slide Time 23:00)



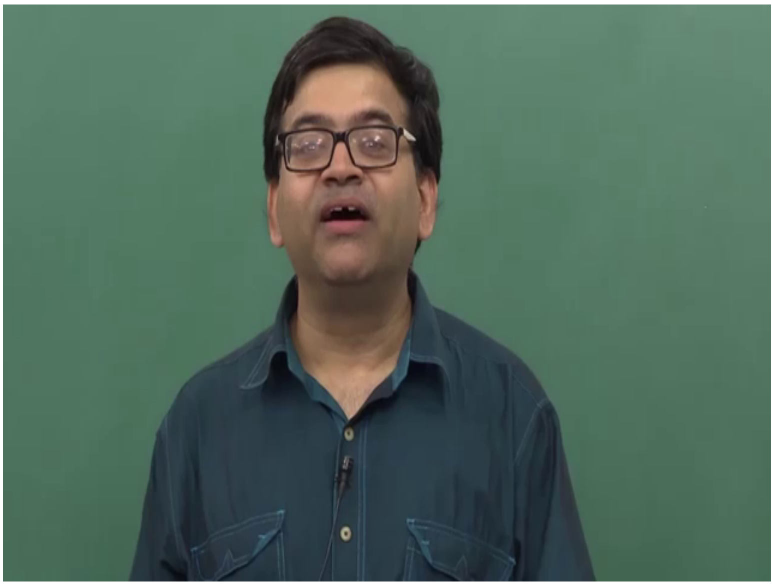
And if v is a valid codeword what property does it satisfy? If v is a

(Refer Slide Time 23:09)



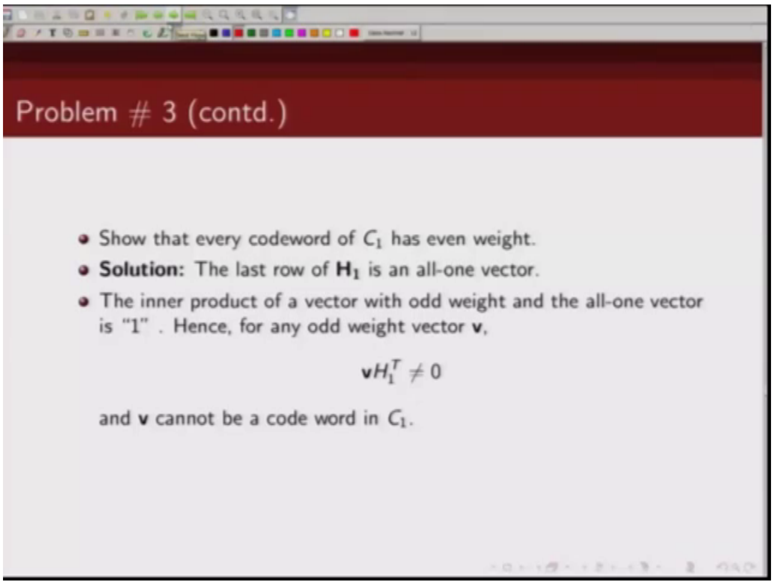
valid codeword then

(Refer Slide Time 23:11)



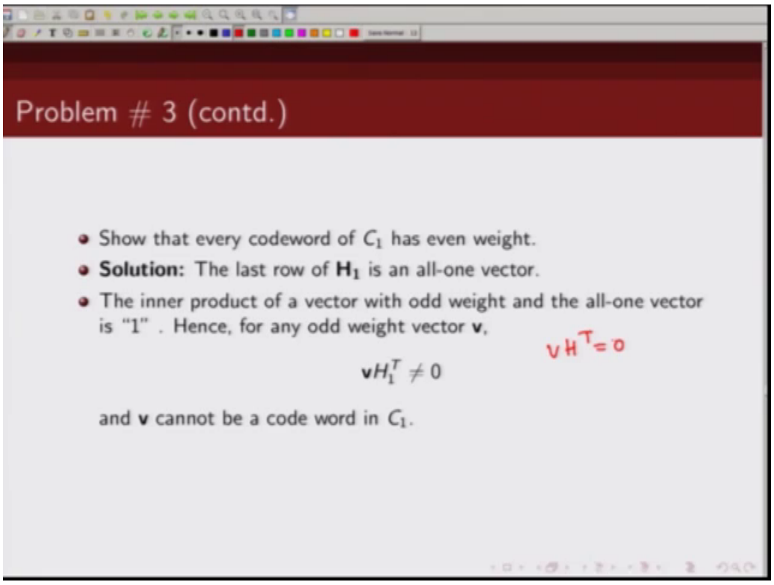
v H transpose should be 0. Now let us take

(Refer Slide Time 23:16)



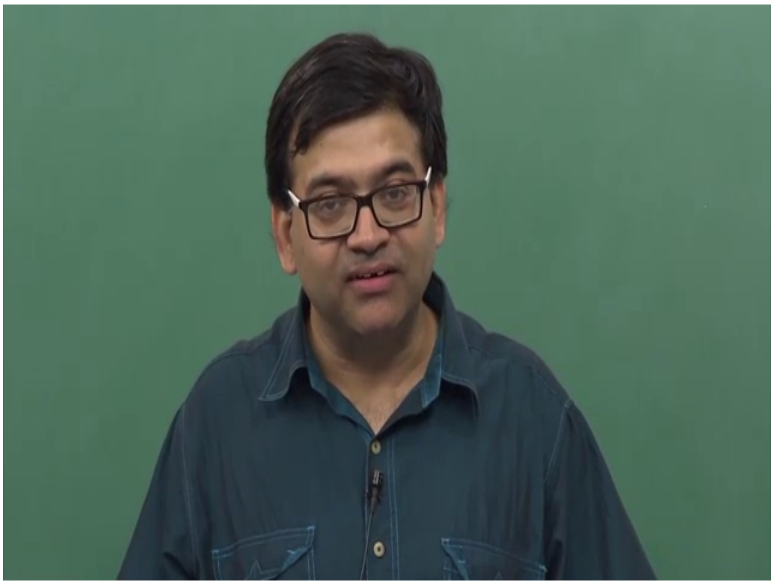
a codeword, let's say there exists a codeword with odd weight which is generated by, described by this parity check matrix H 1. Now if we do v H transpose, so when you are

(Refer Slide Time 23:34)



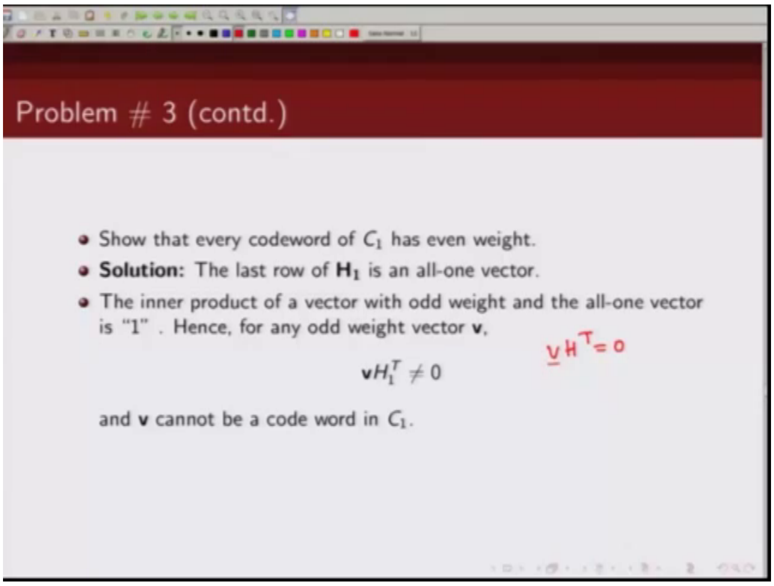
going to take the inner product of this code vector v with the last row of this parity check matrix what will you get?

(Refer Slide Time 23:45)



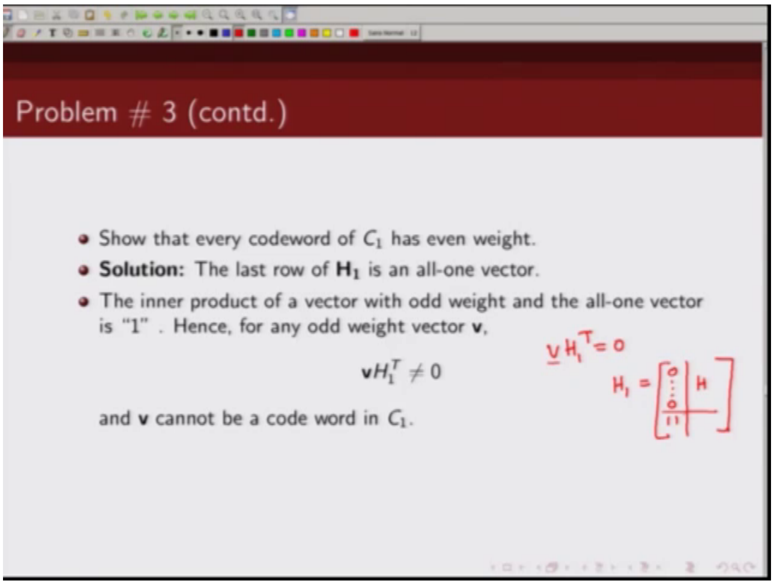
You will essentially get sum of, so basically

(Refer Slide Time 23:52)



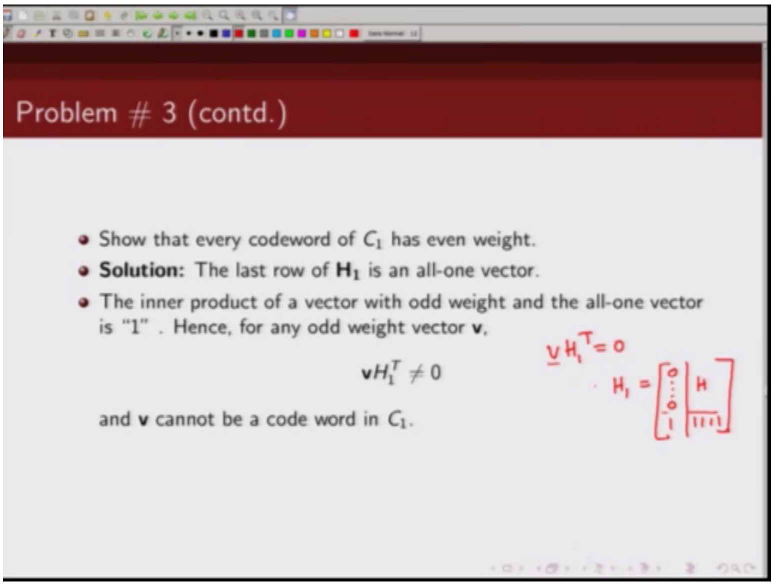
if you do v H transpose so your H is of the, so v H transpose where H is, H 1 transpose where H 1 is given like this. This is all 0's. You have your parity check matrix H and you have here all 1 matrix,

(Refer Slide Time 24:11)



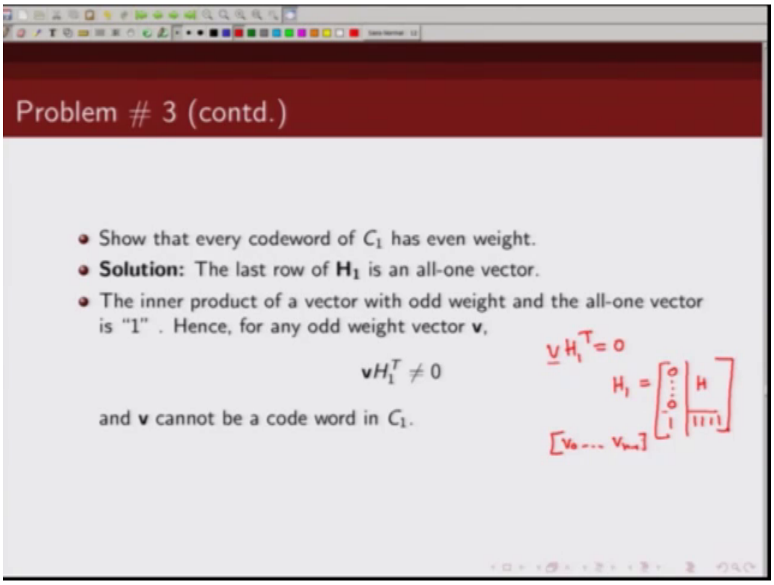
sorry you have your, this is 1, this is 1 and this is all 1 vector. Now when you do v H transpose

(Refer Slide Time 24:24)



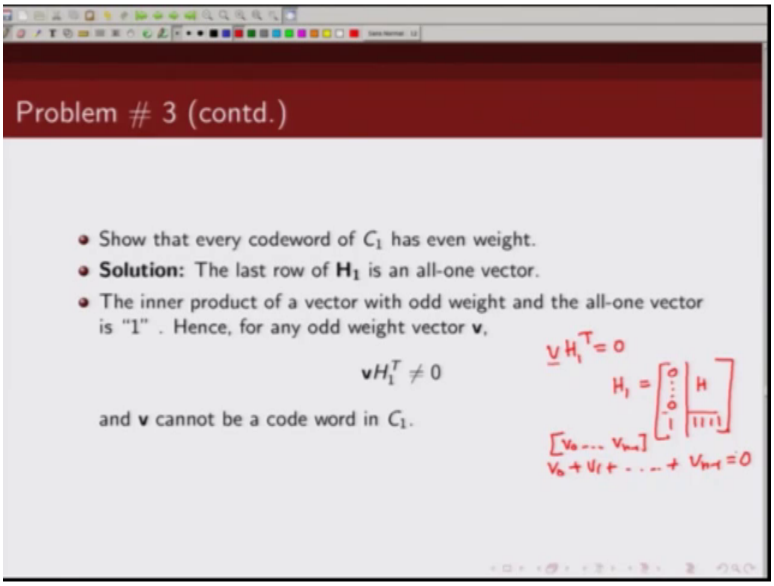
so let's say v is your v 0 to v n minus 1. When you do v H 1 transpose what you will get

(Refer Slide Time 24:32)



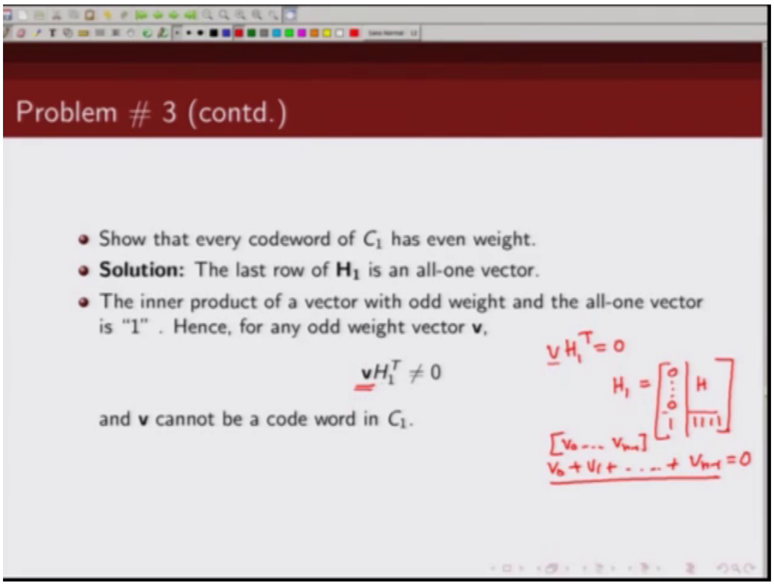
is v 0 plus v 1 plus v 2 up to v n minus 1 is going to

(Refer Slide Time 24:40)



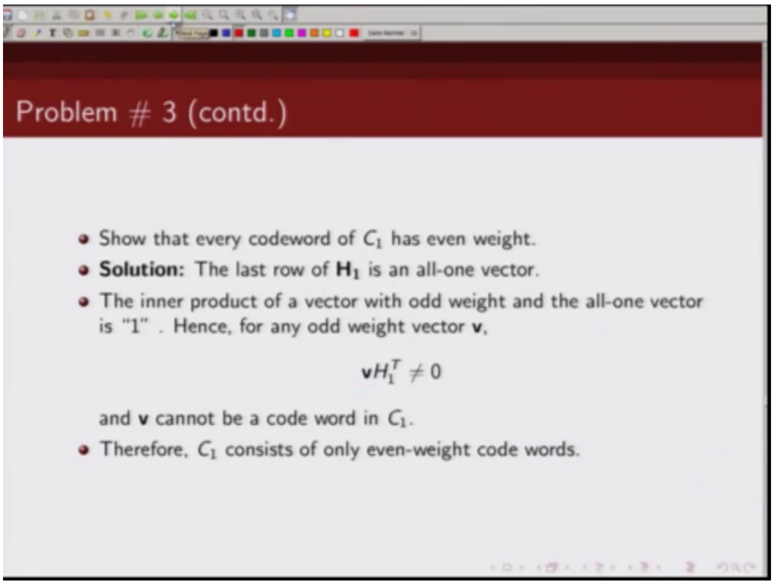
be 0. Now if this v has odd number of 1's this sum cannot be 0, right? Hence we prove that v has to have even number of 1's because we know if v is a valid codeword then v H 1 transpose should be 0. So if we do v H transpose because the last row of this parity check matrix H 1 is all 1, the condition that we will get is the individual components of this parity, code vector v, v 0 plus v 1 plus v 2 plus v 3 up to v n minus 1, basically v n minus 1 they should all add up to 0. Hence we cannot have an odd weight vector which will give

(Refer Slide Time 25:38)



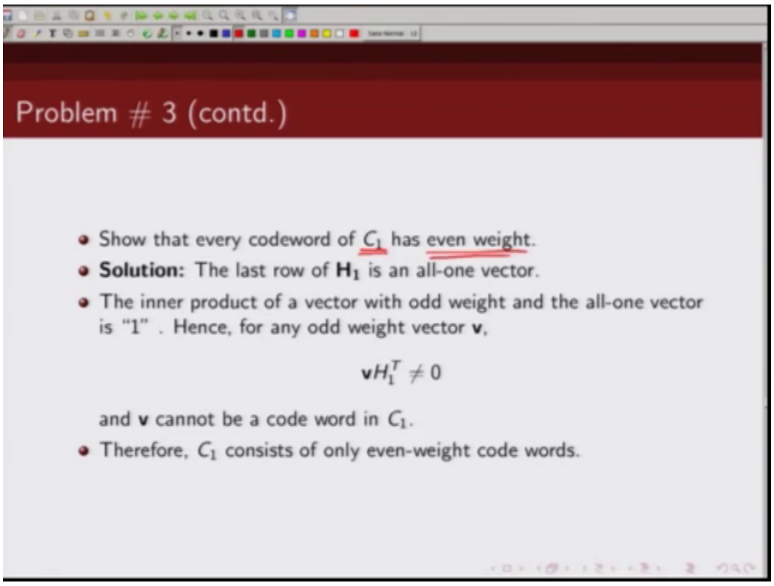
v H transpose, v H 1 transpose to be 0. Hence every codeword

(Refer Slide Time 25:45)

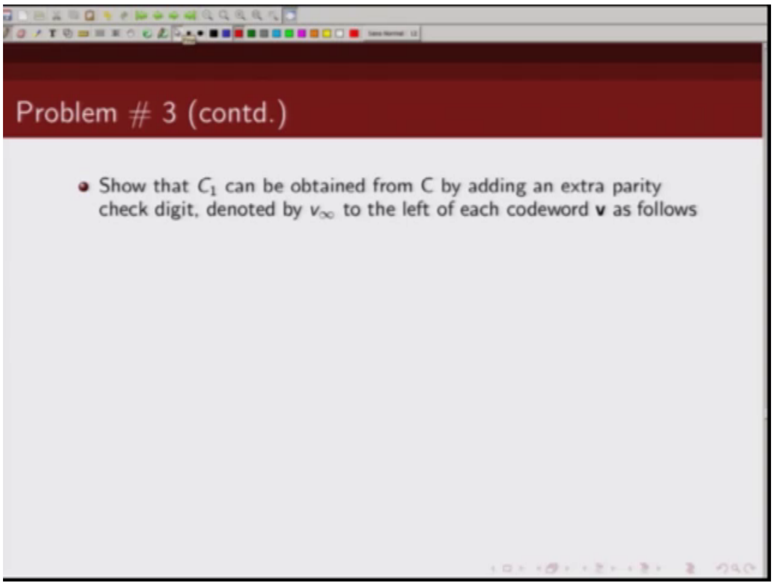


in C 1 has even weight.

(Refer Slide Time 25:49)

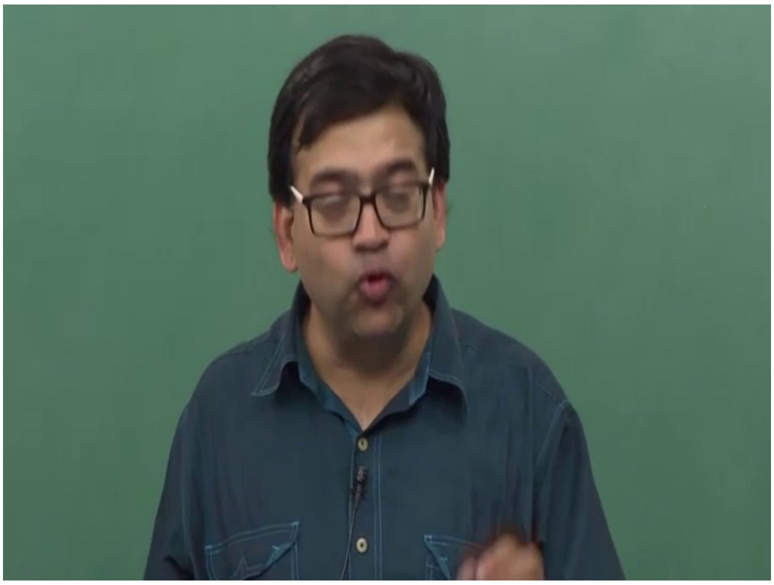


(Refer Slide Time 25:51)



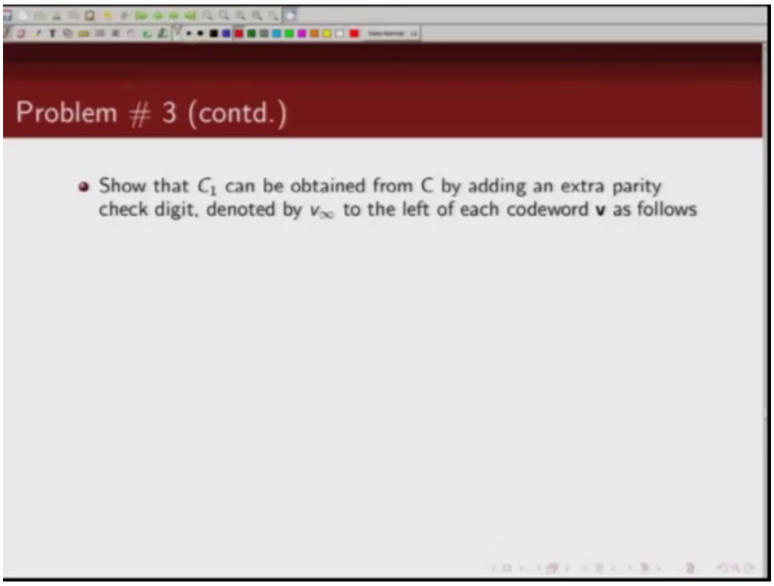
Next we are going to show, prove how we can generate this new

(Refer Slide Time 25:57)



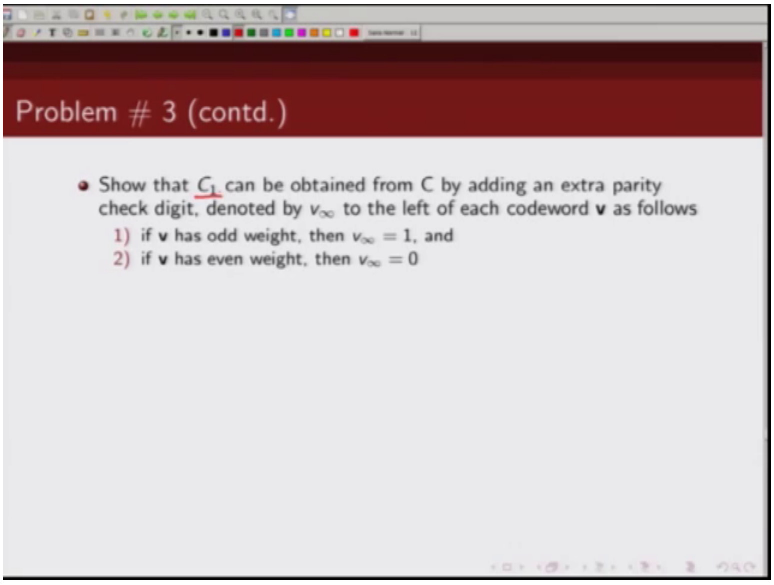
code C 1 from the original code C and what did we mention?

(Refer Slide Time 26:02)



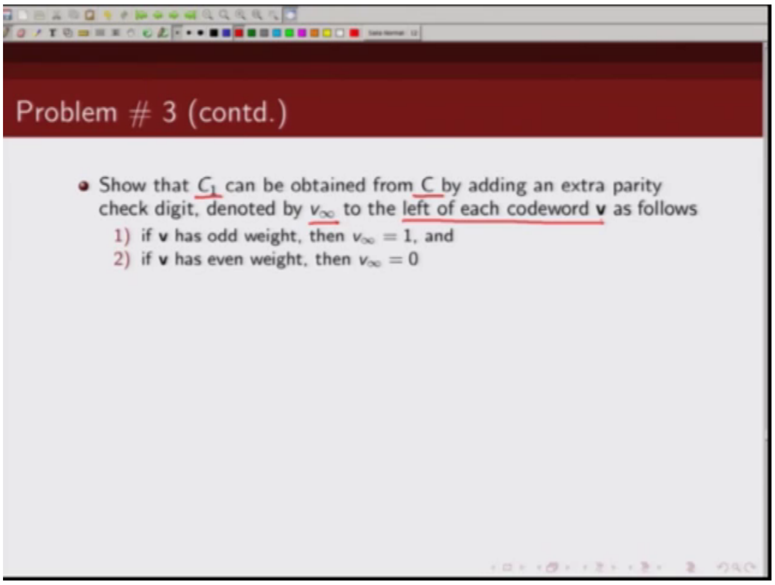
We mentioned that this new code C 1 can be

(Refer Slide Time 26:06)



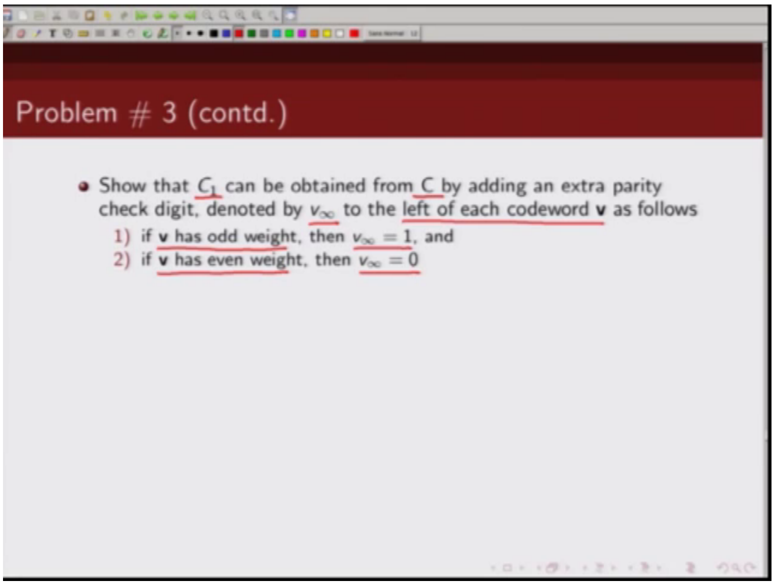
obtained from the original code C by adding an extra parity bit which we are denoting by v infinity to the left of the original codeword v in this fashion.

(Refer Slide Time 26:21)



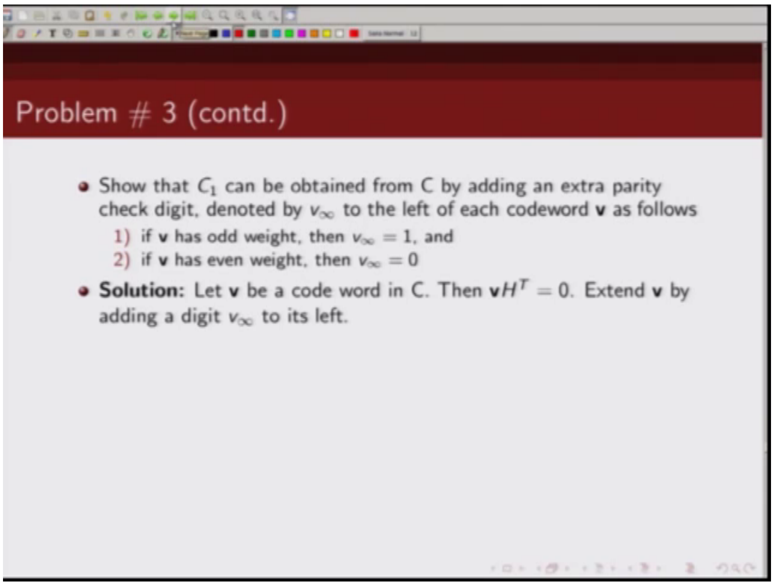
If v has odd weight then v infinity is odd parity and if v has even weight then v infinity is zero parity, is zero, is even parity.

(Refer Slide Time 26:34)



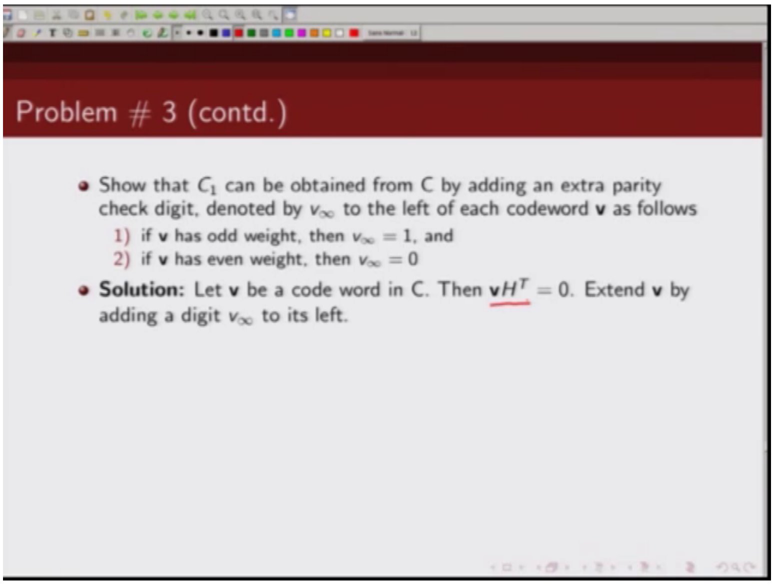
So let's prove this. So let's see,

(Refer Slide Time 26:37)



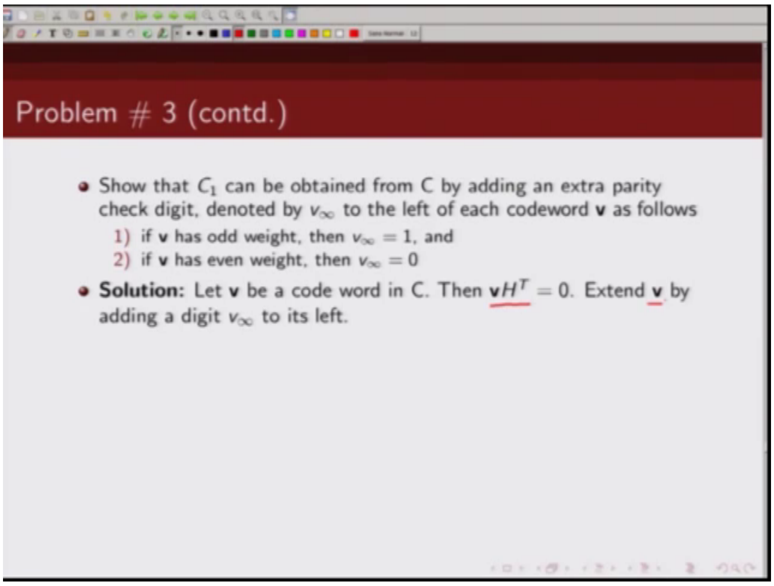
let v be a codeword in C. Then v H transpose will be 0.

(Refer Slide Time 26:45)



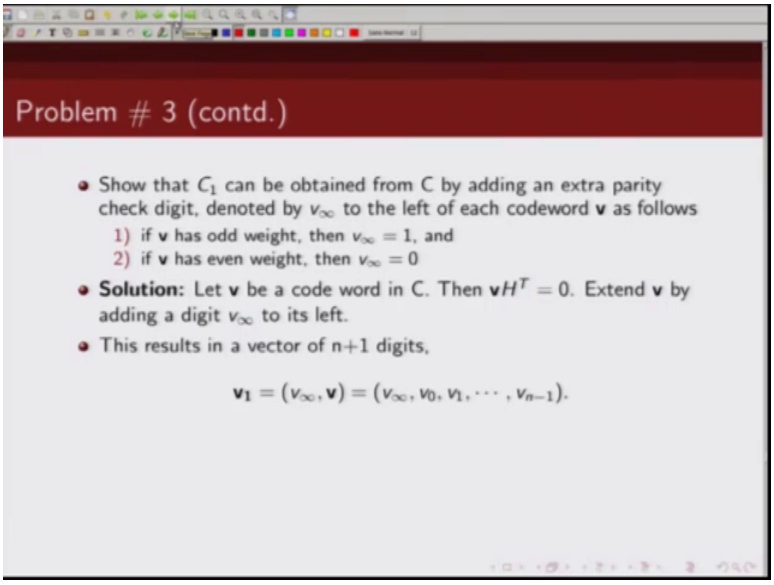
Now we are extending this original code v by adding a bit

(Refer Slide Time 26:50)



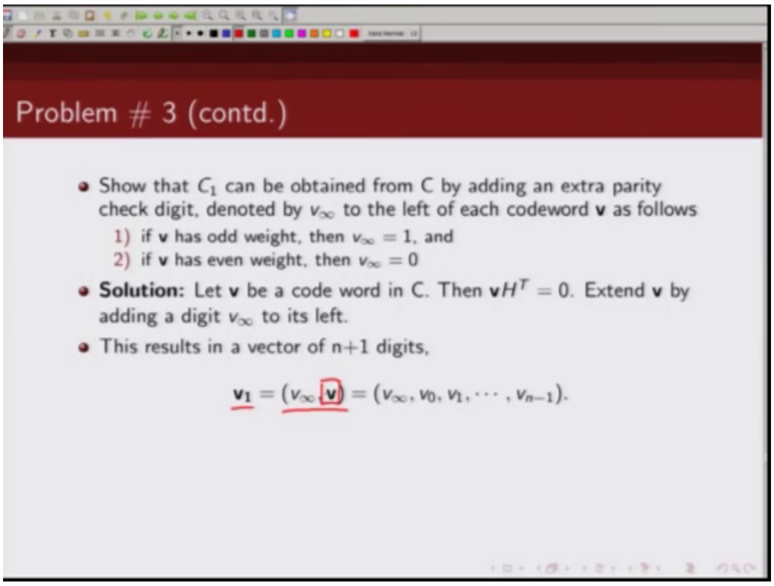
v infinity to its left. So we are defining a new codeword

(Refer Slide Time 26:53)



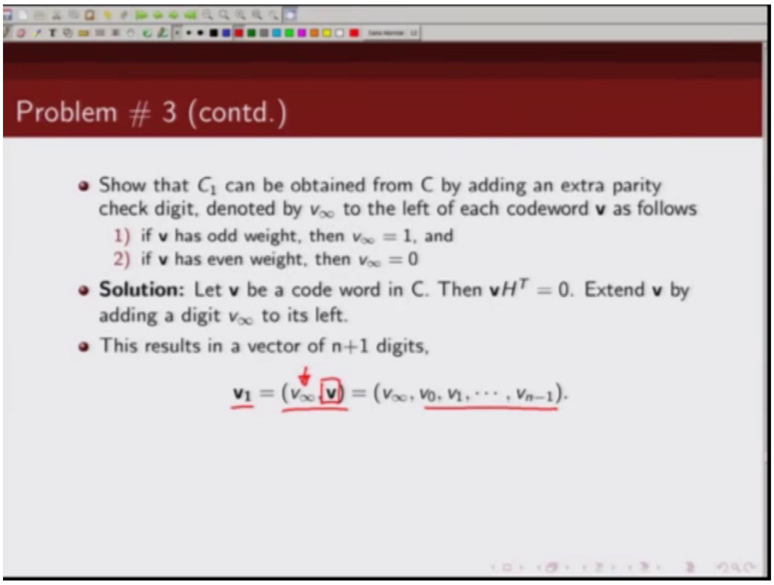
of length n plus 1 which is defined as follows. So this is your original codeword v which is basically

(Refer Slide Time 27:02)



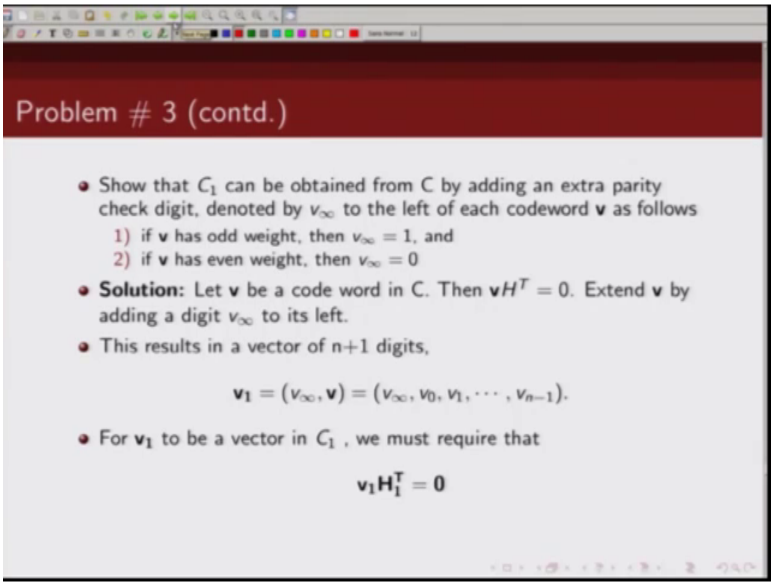
v 0 to v n minus 1 and then this is the additional parity bit

(Refer Slide Time 27:07)



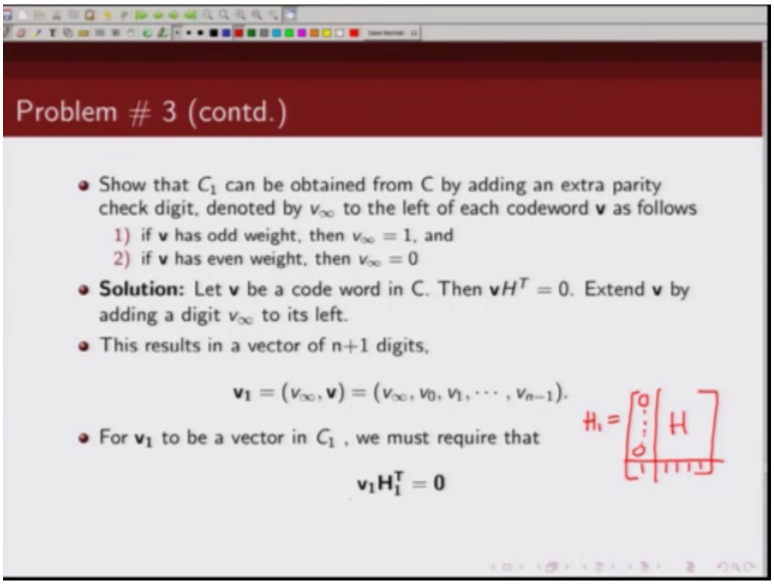
that you added to the left. Now if

(Refer Slide Time 27:13)



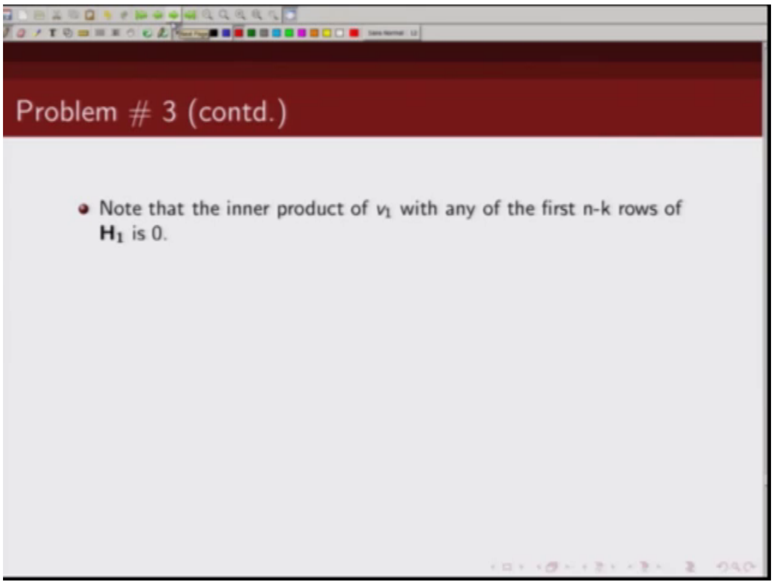
v 1 is a valid codeword, then it should satisfy the property that v 1 H 1 transpose should be 0. And what is our H 1? Again please recall, our H 1 is of form like this. So the first column here is 0, then you have here the original H matrix H, and this is all 1 vector. So when we do

(Refer Slide Time 27:42)



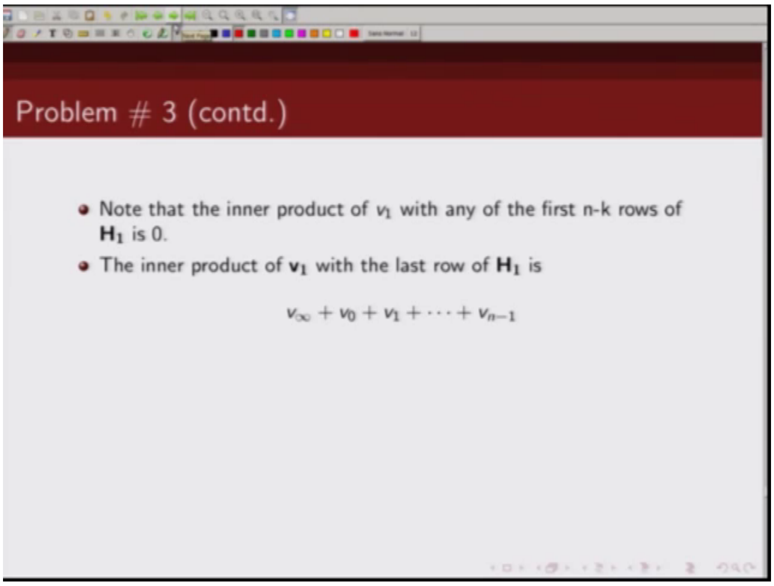
v H transpose, so when v 1 will be multiplied by this last row what we will get is condition of

(Refer Slide Time 27:54)



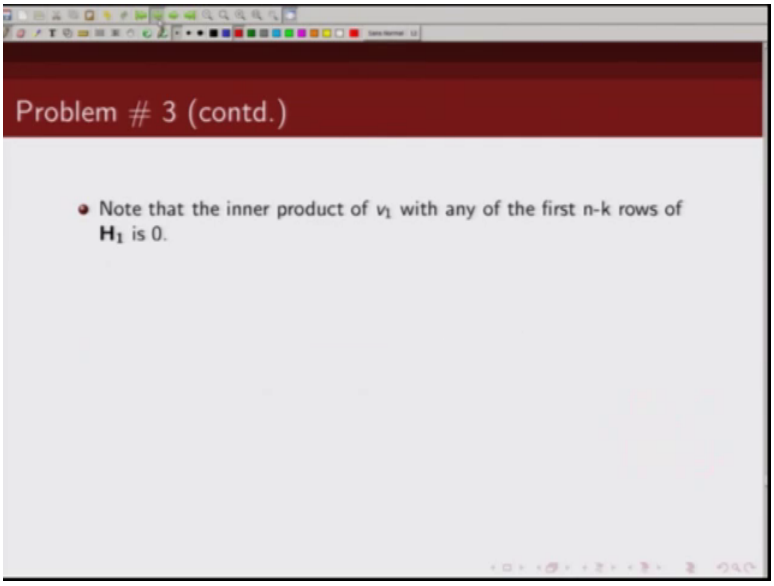
this form

(Refer Slide Time 27:57)



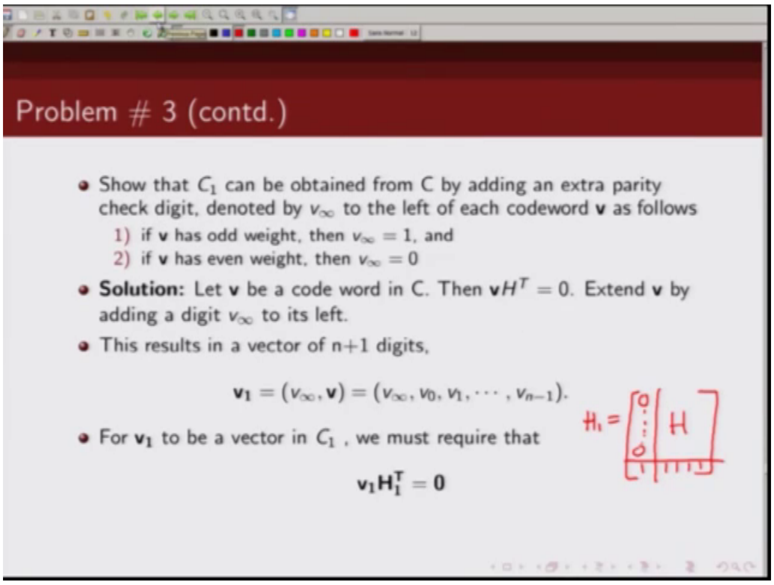
v infinity plus v 0 plus v 1 plus v 2 plus v n minus 1, that's basically should be equal to 0, Ok. Now how are we getting this condition again? We are making

(Refer Slide Time 28:12)



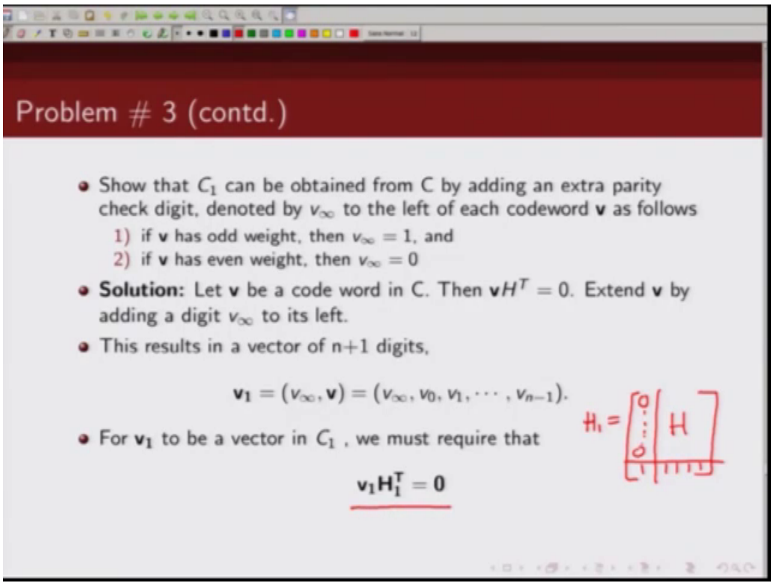
use of the

(Refer Slide Time 28:13)



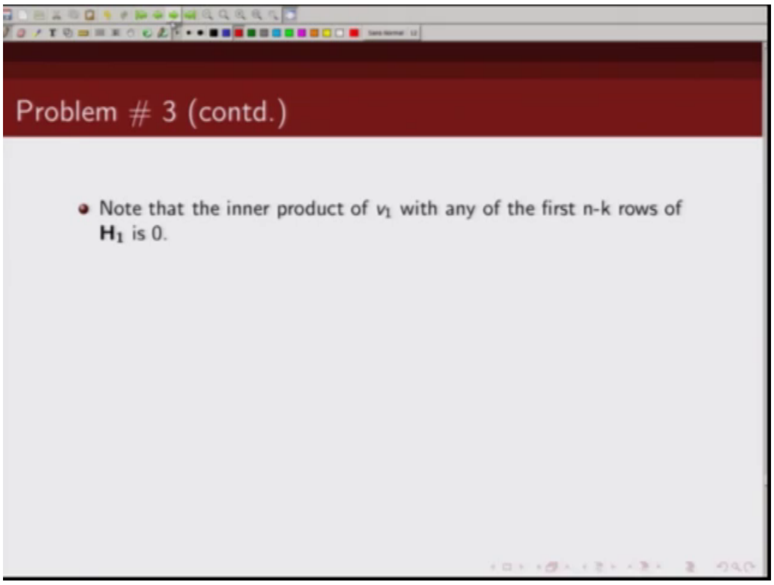
fact that v 1 H 1

(Refer Slide Time 28:16)



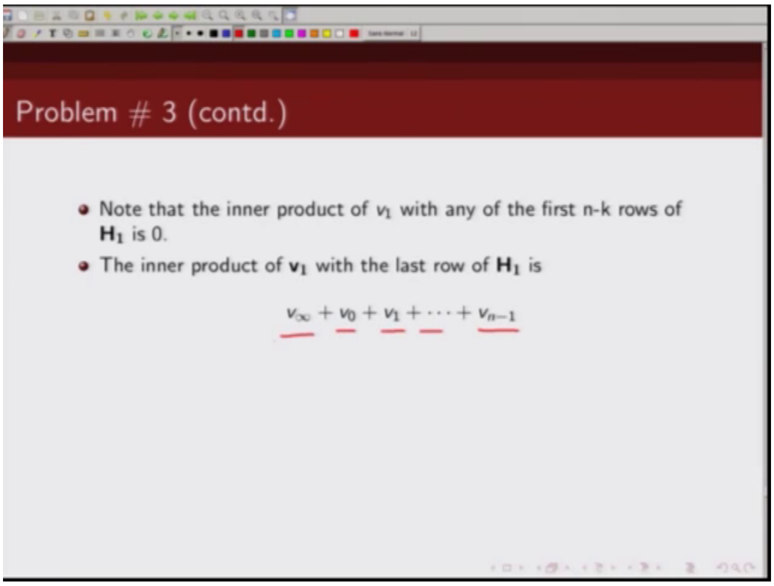
transpose is 0 and H 1 is of form like this. So when we do v 1 H 1 transpose, the last row which will be H transpose will be last column. If we multiply v with that H 1 transpose column what we would get is something of this

(Refer Slide Time 28:37)



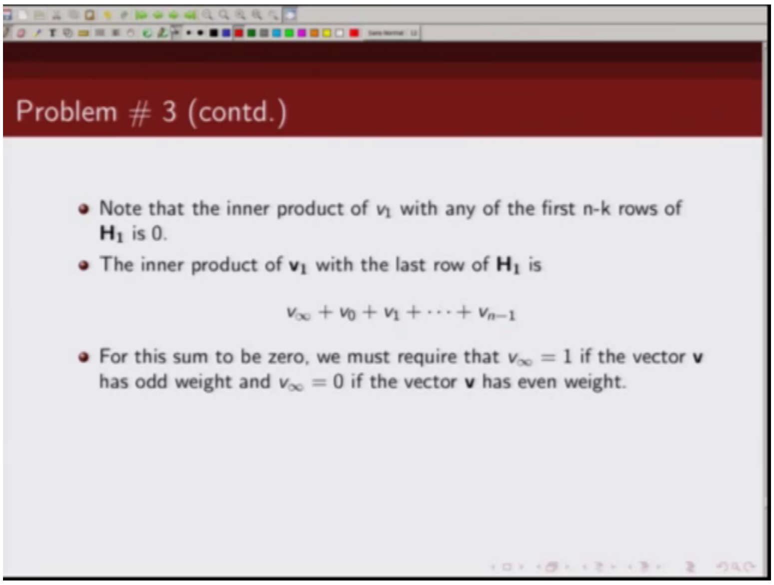
form.

(Refer Slide Time 28:38)



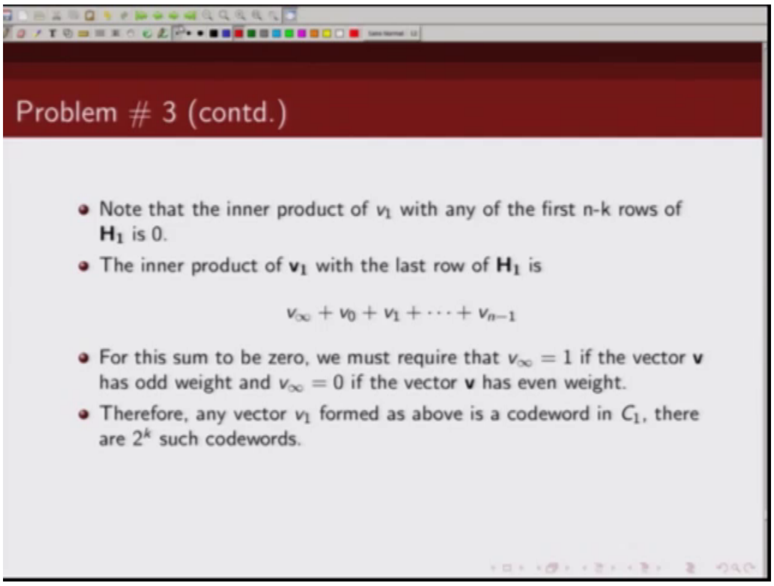
Now this should be equal to 0 if v 1 is a valid codeword, right? So

(Refer Slide Time 28:45)



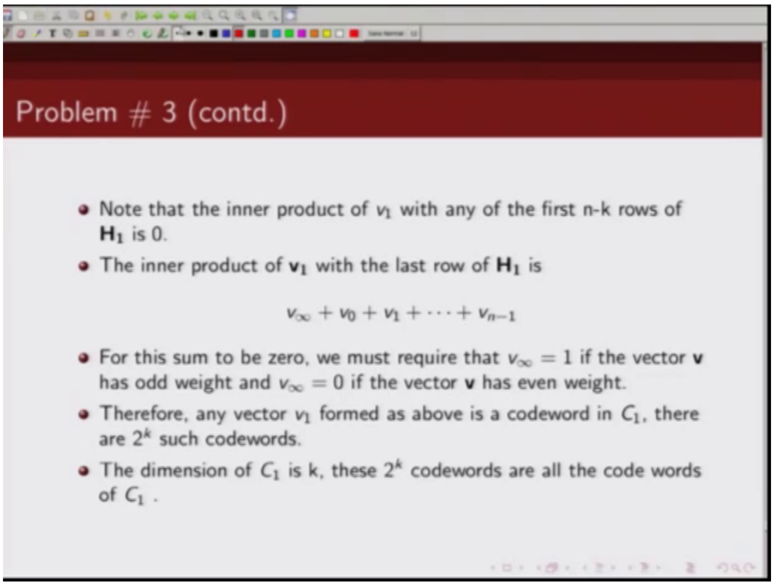
if the sum has to be 0, what do we need? See the original codeword is odd weight codeword. We need v infinity to be 1. And if the original codeword is even parity, then this new parity bit should be 0 and that's basically the proof how we can extend the original code to construct a new code. And this is basically,

(Refer Slide Time 29:18)



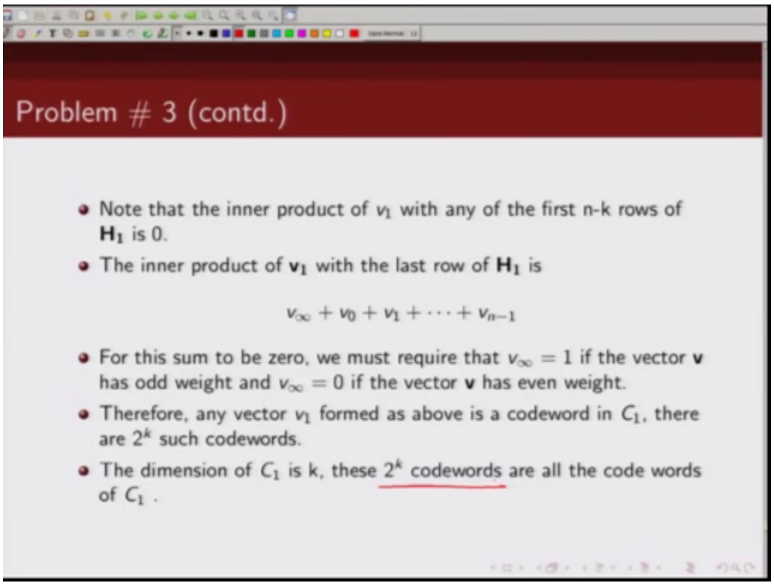
if this is equal to 0, we know that v H transpose, v 1 H 1 transpose is 0, so v 1 is the valid codeword in C 1.

(Refer Slide Time 29:31)



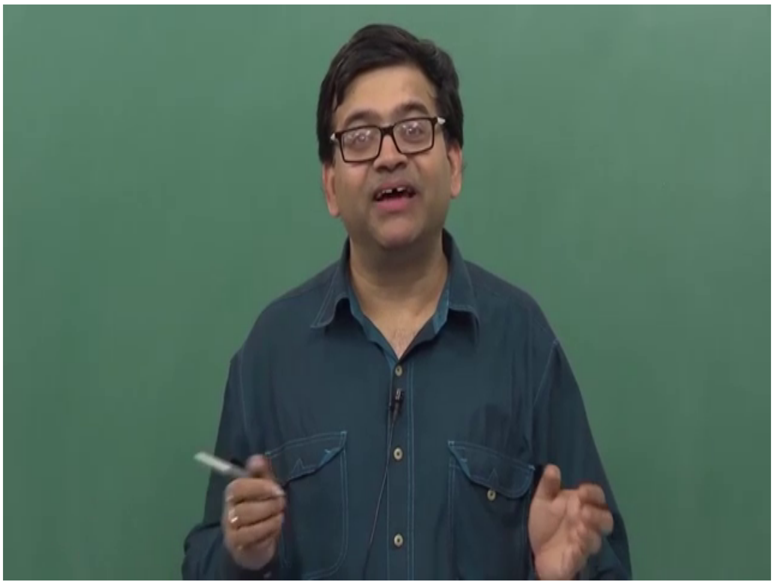
And total there are 2 k codewords. This we have already proved in the first part

(Refer Slide Time 29:37)



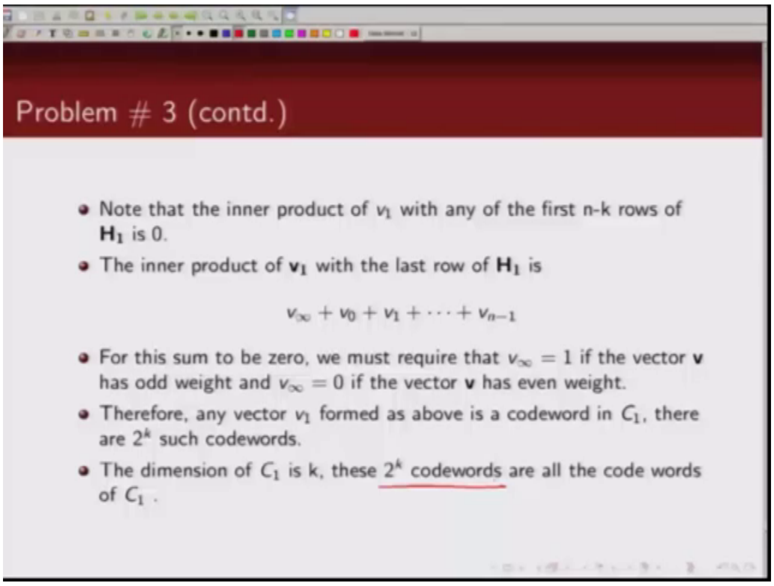
that there are total

(Refer Slide Time 29:39)



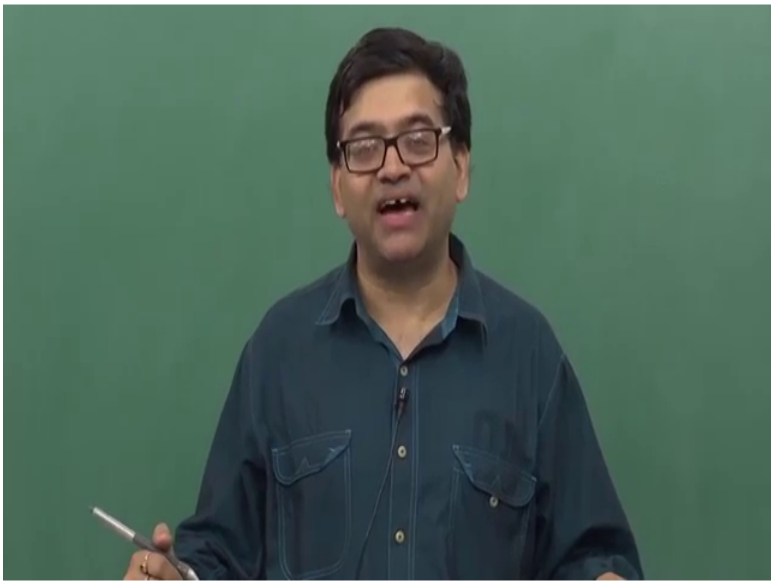
2 k codewords of length n plus 1,

(Refer Slide Time 29:43)



Ok. So with this,

(Refer Slide Time 29:46)



I will conclude this lecture. Thank you