he and his wife very long four or five pages and the other one-on-one or two pages and you have to call them all them today and next week this is the plan we'll see OK discuss about the assembly language basics this chapter three from the book that is why it's called chapter 3 particular some of them but most of all we discuss in general the big difference is that you have to keep in mind something like that that in fact an assembler lation program has two main tasks and a separate checks the syntactic validity of your written code thank you and if everything is OK with it starts with generate funds generate it for you generated by so keep in mind this is the main task of generating bots generating box accordingly to the directives and instructions meant in your source code OK it's very important to keep in mind that generated by its generating funds also during this process during that's the possible invalid expressions the expressions that has been learned they must be computable and to simplify what is the expression is a combination of operands and operators operators are mechanism arithmetic that mechanisms applied to the operators that can be evaluated at assembly time this would be again something that you will see OK so the focus of this chapter three would be on the assembler on the directives one on the directives are indications given to the assembler meant to be very good rules or how to generate the corresponding 1 one of the instructions in contrast in contrast instructions are not directing to the assembly to this this reflation program you are selling your case right just to particular case that you're working it is possible to work possible to work with Microsoft assembler or even it's possible in some other universities somewhere else in the world to work under under a different operating system like Unix and you have some other assembly language directed to the Unix machines and so on but usually again you have any problem OK so during this argument as directed to the assembler instructions are the dressing themselves to the processor because these are commands given to the processor this is the big difference between these two in fact that form with the governments as you would see with which you have some words with the first three ones you met them and you met yourself because they are present at the level of a high level programming languages you have labels in C++ and some other languages that you learned about instructions directly to learn what you didn't have in high level programming languages this 4th element but which probably you Big Bang you did OK you didn't mention where you work with the location counter location counter this will be the focus for half an hour from now on but before that let's try to see what are the first three lines or 4 definitions here we give the definition of a machine language which is the native machine language is the native language to which the architecture reacts or with which or which the architecture understands OK is the set of the machine instructions which the processes directly reacts but this is nothing else the 00110 codes and so on there are nothing else that that yellow formulas yeah for every line which is the general form of the internal format of instruction right fixes then which is the only mandatory field that that has to appear in the internal format of instruction that you have what I mean if you have some operations that by you that you that you have memory from memory or registered if it is a register it starts if it is a memorial program you have to go further and you also will have at least other two bytes the same byte which says scale index space which is the reflection or the equivalent of the 2:00 AM in the morning office competition for and also you have that baby you will have also that what displacement part yeah displacement can be immediate but again these are not mandatory also the zip is mandatory displacement and constant it depends OK space you have to have so this is the machine language what is this table language the language is a very big step forward because it's the English version of the machine language which is really a human language it is the symbolic language right is a symbolic language because you as humans as programmers you are expressing yourself starting bids there 011001 but you will label your repeat strings you will label them with symbols this is true also for the instructions because you put their name on it which are denominations for instructions move add salt and so on the names of the instructions on from them on mix yeah OK Nate all of the instructions and also you have also inform your variables these are also symbols from the point of view of the processor can be assembled doesn't care about your needs I mean it does care in the first place because it has to be polite with you and has to translate all this garbage that you put there names for the assembly from the processor these names are garbage because they have nothing to do with what they are they understand it yeah from their point of view a sample and the processor they acted only numbers these aren't the family language and the sea programming language being the most language yeah everything that's flows there it is you bet you can the answer to an American bank yeah so from December there has to the IT has to translate everything that you write in fact in numbers and everything that can be evaluated at the same time as we will see you cannot develop everything at the same time for example you cannot evaluate the value from whatever you can only runtime for instance with the really resident in your running yeah but there are many many things that can be verified everything that can be very assembly time will be elevated and the corresponding by then will be generated by the assembler to obtain the exact image and this ex image will be safe this and you have to get them there those parts together with the data part that is why the hold thing that you are writing in US alone will be the components of the exit why data segment and the code segment both of them yeah we'll be there and like we discussed last week and two weeks ago about these officers specification formula any options that is issue oh it is accessed as a direct addressing mode I mean every time you put the name of variable every variable name will be by your center to its hey which will be from the point of view of the sampler will be that constant part thereafter constant determination simply time OK today we'll make a little bit of difference there OK so this is the assembly language my mornings labels and so on OK let's go to the location counter did you use location counter now probably not been every week in what context and what did you understood until now and what is this good for and which is the context in which is probably recommended to use the location counter from the point of view that done when you have what at least an array OK it's the logical structure from our point of view yeah I mean the assembly language doesn't help the understanding of the right but OK this means that I have a defined by problem many lines this is my

OK and then OK so exactly what would you write what just for three you're in the first row so probably OK bye each understood what why did you make this projection so you didn't you're my guy there because This is why I'm present here to try to explain to you why location I know why are you OK what is your understanding about where you OK and if the location counter tells us how many bytes were allocated until that moment for our data the consequence is that OK but why did you wrote all the miners 1 off the memory of the beginning of the memory area to which I'm referring to again OK so why because at the end we will obtain as a result of subscript traction 1 that this memory has OK so you subtract 1 quantities as elements you subtract 1 addresses very good this is explanation is is for everyone everybody understood that OK but because you subtract to address it my question for you is in a little bit larger from we have to understand what in computer science is called pointer arithmetic I mean my question is which are the operations allowed performed with pointers in generally computer science it's not something specific to assembly language because what we will discuss in the next 15 minutes it's something that is true in the whole computer science from 70 years ago starting and even today these operations are the same seven years and independently of the operating system independently on the aperture independently of the language what we will discuss today is really true Inc for some people she works with the dresses being the most lovely language that is why we have this reflection also like in assembly language so please tell me let's try clear numerate which are the 7th grade or how many operations do you think that are capable to exist are able to exist in computer science again as pointer arithmetic acting as operations with pointers yeah how many do we have and let's try to write on this part of the whiteboard and to see how many operation itself pointer arithmetic how many operations we have let's see let's innovate what do you think how grateful that you think that it's OK and will be useful for something no OK I say pointed arithmetic means computations mathematical arithmetic computation the references it's OK something that you can do with the pointer but everything is made by applying the certain operator so because I'm I'm I'm writing there pointer arithmetic I can show you what there is nothing is point point I think it's that as in can we extract the square root or seven or the roof from point there's a point of arithmetic they should be appointed arithmetic yeah mathematical operations with pointers that is why I'm asking so which arm can anybody tell me which is the least except it the staff of these perceptions has points in computer science in general it has nothing to do in the first place also applies that implements them but any leverage that does that OK OK multiplication division increment is dictation is something 4 something that treats you 132.3 hours in your opinion why are accepted operations arithmetic operations important what will be the output of such an attempt everything that flows inside the computer there's a number but numbers representing what from a semantic point of view because I'm starting to use operants addresses operands and I want to be yeah representing some other point in memory so let's agree that point arithmetic has the task of using some conference acquiring some mathematical operations point arithmetic operations and in the end we're looking also appointed so all The thing is to go into memory from right to left from left to right starting from one or two addresses to take the final risk of something this is the task this is what we have in mind now let's find out which are the means for doing that an address is always of what kind of number always so square root but you can't do you know something confidential OK so it happens so but if it does not happen oops but I have to give a list of operations that must be widely general computer science so they don't to be applied on any kind of address only this kind of addresses can be let's say you used as a pointer or something that you can apply that and you can work with the square root from one address if you take an address maybe you know the variable or in a register and then perform let's say with the mathematical coprocessor or system addresses you can call functions from system calls and you make the square roots from a point of if you want that's not considered as a pointer that's considered as an integer number which has nothing to do with the address point of view as the data OK so we are talking here about pointer types from the point of view of the processor in the assembler what we will work with here in our discussion are pointers as data types the data type of the occurrence that we start with are pointers so this is my question again what conference is the operations can we do this pointers this is the general question the general answer is anything that does make sense right one of reasons can you do on which I am based in computing another address and I cannot express from a semantic point dividing 2 pointers vision remains what can I keep in mind like there's no download to divide the point for that of theory not to divide the pointer what good what good it is to take the start of the break of a memory area and divide it to some other point I'm taking one use the based on which I'm moving into memory it's no semantics that can be associated with that make sense so the vision eliminated can we multiply the pointers doesn't make sense because let's suppose you take the last two pointers from memory what do you obtain by multiplying the last two addresses from the memory from that four gigawatts you'll take something you will not pay for that so multiplication of the pointers doesn't make sense for any target for any goal that we get every month division of the poisons we don't have tools or similar positions is shrinking more and more so do we really have some point or you just messing around here messenger yes no but it's secure can see C++ I got articulated language which is the most language OK basically that is why we have tries to correct in a way OK so we don't have pointers multiplication we don't have pointed division we don't have square roots what the hell is still there addition and subtraction but they will disappoint you let's take again the last two pointers from memory let's watch the hill OK something for sure you don't have addition of the pointers yeah you don't get that according to exactly like you did when using this location what you did it is to use the only operation capable of performing a mathematical operations on two points the only operation possible and the language is subtracting 2 pointers because there's a logic and like I said anything that does make sense make sense to subtract the pointers because this will show me how many bytes of memory are between the beginning of that memory area and the end of the memorial if I have P&Q that's two pointers and if I am performing Q - P I will obtain exactly the number of bytes between P and Q do you understand that is why also what you did with dollar minus 8 is really subtracting pointers download means exactly the point where I am with the generating files so you can you can think like that what is what the hell is this location counter it's like the cursor in Word you have a blank page start write something you are generating by generating by generating vices like this ever and any moment with this generation of white OK you are exactly the point the cursor the position the current position where you are and that's memory segment you have to think that any page like the memory segment any segment has its own location counter right now because I don't know it's very clear what you discussed at 7:00 on Sunday so the location down but it's not like the general variable we have only one location cargo per last any program but instead any segment any memory segment from your program its own location counter location so the first pointer arithmetic operation that is allowed is subtracting 2 pointers because it does make sense subtracting two points and like I said if you have area this is big and Q these are the pointers these the pointer that means that it's the beginning of the memory area cube is the end of the memory area if you perform a kill minus B you're really opening the number of bytes between P and Q so it does make sense is a useful information it's a dynamic information in a way because you always can have from one run to another you can have a dynamic I don't know length for something so it's very good to have a a general expression that independently of how many bytes do you put there you will have always an expression that depth and will compute for you compute for you really the like this is in fact the main role from the location council to dynamically affect to wonderful length wants to put there for array for logical array you have do they have still something in addition do we have in addition something I mean do we still have something in addition please value addition OK so adding a constant to appointments it's something distinct from what is about adding a constant to appoint as is radically different from adding two points why he suggested there that I told you what is really happening in sea and also in assemble when you write yourself this is the contents from a + 7 so this is a arithmetic this process is a point of ethnic and this point arithmetic as an operator is generated by this square breakfast yeah this is a operator this is going to some plus seven it's OK because this allows you to go into parts an array it acts as an index this constant that you add that as an index in trying to parse a logical grid or a certain memory area from a practical point of view inside the memory inside memory OK do you have something else stop tracking add constant from a point of course the P - 7 also makes sense P + 7 means go here 79 but also you can go backwards in memory with the minus seven constant but anyone of them is meant to the same go it allows you to go to the memory to the left to the right and you are able to parse without going of course one question can be and what if I go out of it because I can go out of it of course but you can do that of course first of all if you take the last possible memory address from which is representative word on the on the double word of course that if you take the last possible number there you will go outside from a double word and the question is and it depends on the of the assembly first of all this is a constant expression if it is not a constant expression and it's something that can be very digit runtime only it depends on the perception that but usually the figures like that probably it will tell you memory access violation or something like that or in some situation it can accept that depends so don't know exactly probably depends on the make that to go from the beginning of memory again yeah you see it depends but usually this company going forward here and something that it's not really case so a reaction will be possible OK the 4th one louder please come back to the right it is an operation that can be done to check in it's in a valuation like for example it's threatening the value from a point to again that these are not the reflecting corporations yeah open up 75 can be done but it's not going to the written they're arithmetic operations something else the gap she movies that this put it down check OK can I continue which is the phone operation possible with pointers there is no 4th operation these are the only three operations possible pointers nothing else will be accepted ever everything else that you might want to right mathematical operations with pointers will be rejected as syntax after pretty until I killed myself five years ago when I first met when accidentally I write the cube plus B instead of Q - b and I saw the December didn't report operation baby sampler this expression is supported fortunately after two weeks of discussing with the defender the guys from there keep in touch they didn't know something like that they did much more people that I could do at that moment and explain to me via something that didn't find on the Internet very clearly even today what the people and unfortunately it is accepting unfortunately it doesn't mean additional I will explain you next week OK don't people around because they check out it's a little fresh point no OK to see how important these kind of said was appointed for fees and which are the implications of such concept so important can we agree that addition is commutative no for thousands of years yes design no yeah it works because this means you have to be avoided arithmetic this is something that of course in assembler works but it's not astonishing what can be astonishing that it was for me at the time that I found out is that something like that it's possibly see if it's possible in see also because C is the most so it reflects what the point of arithmetic means the contents from a memory location which is obtained by applying a method of performing an arithmetic operation with pointers or with constant pointers keep in mind something like that is possible is valid it's no syntax error you can use it multiple we want just because you really perform a pointer here and the second language and also see accepts that and this is the understanding OK some other important aspect here he's going to stand what is the output of each of these operations from the point of view of the data back when I'm ending OK and starting with P&Q as pointers always appointment take the time is an address when I'm at the constant to a pointer I will obtain as an output of P + 7 is also a point of data type 3 - 7 again would be a point of data type Q - P is what it's a so-called scalar value immediate value scalar value it is something that it is computable at assembly time by the assembler I mean all of them are computing computed by December but it's not the pointer type it's a scalar value it is important to understand that because they are regions regarding the output of some expressions I mean operators can work only with some that you will see even today that it's important to see on what kind of data types can we apply some operation some operations can be applied to pointers some of them are valid for scalar types OK why would you stress about two reasons first of all I want to discuss the interactions of the 2:00 AM in the morning for the line with the offset computation formula do we use pointer arithmetic in the opposite computation formula this is the opposite specification formula look here these plus must be a plus and not minus this is a plus and must be a plus another minute is this point arithmetic yes and no I mean it's not a point arithmetic because you don't really have pointers there or anyway it is a special for you add it is not really a point here but you have the restriction that you must use the addition sign and never subtraction if you will try to put there an expression which you will use subtraction it will this impact because you just use your values from register which is really a special case which only in the end after all the formula is evaluated and computed will be interpreted as an obstacle but in the first place once you find there is not an address it's really the contents of register multiplied by an index and so on so they are really values that only in the end after the painting the final value will be interpreting all this stuff as an address so it's not really a point originating here in the Oxford pacification formula these two classes like you have to keep in mind that they really must be glasses not mines OK so this is the first remark oh but if this is so let's try to give some examples here to see if they are allowed and what do you think about this example so for instructions here I have this declarations that they should be defined by 1785 words once you see the device our word one field and then I will write it this instructions let's see if they are locked they are correct not correct cancel the first one 8 plus edgings it's OK this is the basis this is the index this is the scale 3 - 7 is what it's the constant it's the constant computable at assembly time but also we have to look at that interaction because it is a pointer graphic you take the opposite of me you subtract the seven and you will obtain as the database point but it is a direct address I mean it's not the point of that is called indirect so a direct address point during is always a quantity computability assembly, This is why it is correct thank you OK so it is correct the second one correct because you can't put the minus in front of that I mean you cannot stop you have to put the plus here so if you put the plus here how can you write that you cannot write anything else because if you put the plus with it right which is not this offset must be always a positive balance you cannot afford to the negative value of an office so this is syntax if you want to know exactly what this is written pretty probably pass it this time turn this then it's something like that syntax error invalid effective address impossible segment based multiplier this is what it will be reported here the third one the same you don't have point of addition it's not an operation with pointers along it doesn't make part of the pointer to arithmetic list of operations possible now you see how important is to understand and to know exactly what the pointers arithmetic is in computer science in general because you have to in valuate some expressions if you don't know that you clear based on which you must evaluate the correctness of this expression you're lost episode #2\*\*\*\*\* pointed the last one louder please I don't understand 8 minus this video OK so I'm not talking about base class this is true and I just put them in oh excuse me right you're true in the first yeah OK so suppose second now can you tell me if this is correct or not from the point of view of this pression because I suppose that still believe that didn't item that can be presented correct or not greater than something like that wouldn't be allowed we cannot know that until runtime getting the money because I don't know exactly how that is from there so at the assembly time to be very clear the assembler can evaluate this line only from the point of view of the syntactic correctness it will perform the subtraction and the operations and of course that only the processor in the end will decide based on the values from every sidechains from the exact moments at runtime decide if you this a memory violation error or not the correct computation correct address from the point of view of the computational so penalty side because even here if you obtain something very big probably will be let's say cancelling away by the values of being values that can be found there so we cannot know that people exactly say that so from technical point of view it's correct because here we have a pointer and this is a scalar value of course that we are referring exactly to the above this will be a negative value yeah so this is something negative in here let's say you can write something like this and it's OK because it's still a plus from the point of view of the formula and also it can be a - 7 because subtracting a constant from a pointer is a pointer prismatic operation accepted so from a syntactical point of view this is correct and accepted by the assembly OK stop stop why appointment still remains still remains this is not accepted mine is really not necessary very good question so the question is even from a point of view of a syntactic aspect OK is it correct to perform an A - b between award and number of this order question OK what do you think so even if you define a memory and then one based on words and then another people your Facebook double words anyone of these three memory areas they do have an address and their addresses are always and for all three of them quantities represented on 32 bits so AB and really from the point of view as an address as addresses they are all three double words this was the discussion with doctor specification formula what the hell is constantly in the case of contents that is why it's still be nominated because you cannot have it's a conference the constant variable is underrated I think OK right that is why the office of constant development assembly time for that that we are the whole run of the program OK it's OK pause I want to kiss the first material to try to discuss with you because they are so I don't think and I don't know if you recall I was telling you two or three weeks ago about June 4999 machines during one is the machine from the point of view it's really the practical and the realities of faith the architecture that we can from 70 years ago on all the machines constructed in computer science all computers are nothing else on my machines scientists are very good scientists that it's 1956 I think publish could not be go bold power electrical computer must be built or something like that it's from a conceptual point of view easy I don't know in the future some sometimes something like that would be possible this was the real idea of this article electrical computer possibility this way and this worked the first published and official ideas of how an electrical computer must be built I mean it's a he said there must be something called a processor central processing unit this was CPU this a central processing not the processor the central processing unit oh where the operations are known must be the commands are being huge from a CPU OK so you have some operands and you must work on some data but where the health data is so that is why the concept of having a memory arises yeah you have to have a memory but again if you have a memory and you want to access something for comment was the guy that introduced the concept of an address because you have to put there in some very unknown very old manner yeah to put somewhere in you put a dress as they like to set about it you put 7 you put there OK thanks like that to be organized so you have to have an internal memory also this internal memory probably print on electrical power to one will be political ones so you need a stable memory which is called an external memory and so on OK so this was the second thing but the main idea is that in the end and I don't want to disappoint you but from this perspective in fact what is any program that was written in history you know for membership because this kind of seeing the computer means that you have that memory you access some things there and then you have to put some things there yeah OK so in the end independently of how many IPS and why and repeats and forced and control structure do we have in a programming language and data types complex object oriented language and in functional problem in the logic language and so on one of them you have to have a bigger semantical very dense structures in the end you will end all this stuff we want with the very primitive thing this operations everything is based on the cycle segment I think that the four white even the check or check thought yesterday so any program written until now in the history of the computer science I don't want to disappoint you but it's nothing else that the poor sequence of that's all yes I would computer insistence that to change the values of some memory cells that's old 70 years of computer science that's all that is why because we have only for 900 machines from a practical point of view fortunately from logical point of view we can go in and the nature from some other things but we don't have from practical point to do some other kind of computer but we must keep in mind that everything OK so this languages that are really built around the assignment statement which is really the practical thing I've called rebadging therapy people that will get it's not questionable he's 17 to me it's not something questionable two that it's imperative to do that and this kind of architecture languages are called for my 1 languages imperative languages which are built around the assignment state OK question is now well thank you do you see a difference between this eye this eye London think from while from the right or if there from a semantic point of view they are expressing different things OK this is what we do the point of view of what quantity means the item the left compared to the quantity that displaced from the IT doesn't matter given that 4th plus one or +7 if we are talking about the left value of I and the right value of I is there a difference between these two or they expressed the same thing the same but I've never in fact if you write something like that there's an imperative operation imperative command processor do with this this she stop in the first place when it sees the left yes and that is only thing that it has to do the processor has to go in memory to the address of iron starts and stops there OK identify the location when I have to kill somebody but I identify and I stop it there so I'm interested and select values means that address and stop it doesn't mean the contents it doesn't mean the value from there there is no difference in performance unless value the left there is an address and there is very different between the left there you can right there that is why these are concepts in computer science when you refer to the value of an assignment or the right right yeah right left hand side size my versus the right hand side of it over the side different things this is an address stop this is the contents from an address so this has to be dereferenced that is why because some languages must be very clear on that they do not make the implicit delete the reference dictation on the right hand side for example I have here as I know the language and that's all that I know that language it is called police language where the reference is not making implicitly at all I mean if you want to write something like that in please you must write like that from a syntactical point of view I mean I is always an actress you want to reference it then to say don't go there and take the value you must use the referencing operators and now comes from a logical point of view your question don't you know the actor recently right something like that and my answer will be pointer arithmetic write something is adding a constant to appointment point that incremented by 1 it's a very big difference between these two both of them are possible in place I don't know anything about this but I know that first microphone whatever I want to explain to you at this late hour in the evening OK as of course yeah and I cannot bring them in an official way from that you know dramatic yes so OK it's very important to understand OK simple OK you are plus plus very good plus plus do you know why steam plus plus has two pluses and not only one the first class is from the second class is that in addition subtraction in addition to see has some other many features that is you weren't present and they consider that she must be extended to completed with some other features for example you don't have any see the possibility of passive just by reference they extended and introduced as seen by reference just passed by reference in C++ nothing to do with North Korea intention and they are really some other things for example you can not define the constant in C there is no contest reservoir you have only this preprocessor directive defined and so on which is something else in C++ you have corners preserve word for that nothing to do this is nothing to do with object orientation and another very interesting thing that I will teach you today thoughts and you will find really also page new things about even though that is that one has a special kind of variables called C++ reference values this is really by updating things as plus plus about them I'm sure that you use then in one way I mean C++ reference very obviously have three usages the first usage is to define this means you can define just to fool around the guys that are reading the listings you can do that by using and this is the moment in which you will record that C++ reference variable is identified by the you use the & the & is really so you can write like that in drew Jay this is the name for you define you didn't reference means he has the same address so you see you work with addresses special kind of of variables called reference OK why did I discuss that because you will tell me that what you know about you assignment is like that and you will tell me single receipts expression yeah more clearly received this pressure do you agree this is the general form of an assignment in your languages not at all this is not the general form this is a poor instantiation of the general the general form is that that we learn in this course two weeks ago opposite specification formula because you have to have a less value in the left part which is not always an identifier but this is the general case would calculate their address and expression representing a company and address computation this is the general conclusions it shows me C++ reference variables they do below you to write expressions in the letter and parts and not only the name of available this is an expression OK what is that this is an expression which must be evaluated and can issue or less value or the right value so again this is usable in see not only C++ it's true so look that you haven't your disposal of high level programming language which you have most of this language in which you can write an expression in the left hand part as a general case you see so it's not good and not correct you tell everybody it is identified with some expression because you lie you tell lie if you if you tell something like that the general form is address computation formula is expression and what is really spectacular in this Bing OK the second which is really to pass by reference some yeah but you know that I mean probably the single thing that you know about CNC OK so passing by reference parameters the third way which you you can you can write functions like that insert & when Jesus preach about pardon return and when you apply that is something like that if they are at seven we received 49 you right here not in the right hand side but you put an A function call you put in the left hand side because using & it means that that function which is a simple cross reference variable we refund you left values a reference to other predators it's an exact computation there and it's still that is why you can write that and the real effect is what because it returns the of I it means that we really don't the left value of POI that you want to carry baby what is really happening here is what of seven we received 49% effect of applying dysfunction now you understand because it is a function that returns and left them why did I put you in front of you these things because C++ has nothing to do with you plus plus has nothing to do with language like in the perspective of knowing what the left value versus the right value in specially that's important importance computer science in general the eye from the left is that address that I from the right is the conference and this is through the whole computer science is nothing to do with the processor and the young facture but he has to keep addresses with computational progress Cortana everything that we discussed what are the operations that are present anything that does make sense yeah yeah any operation that makes sense I say this makes sense it doesn't make sense I think the point this doesn't make sense it's not a long subtracting a 77 of eight you see that OK so keep in mind that regarding the point of only three operations are possible those that you see on the left most whiteboard yeah these outputs are pointer and pointer sometimes people ask you is allowed even ask remove budget you will see not now it's it's not really a point to the dishes so I myself in the fence OK this is what I see appointments this comes the examples that we do here on the OK left value versus white the right value what means you have an assignment and what means to discuss about the and the press in the left hand side and the contents in the right hand side the different question but it's not the general case the general case address computation expression receipts and expression