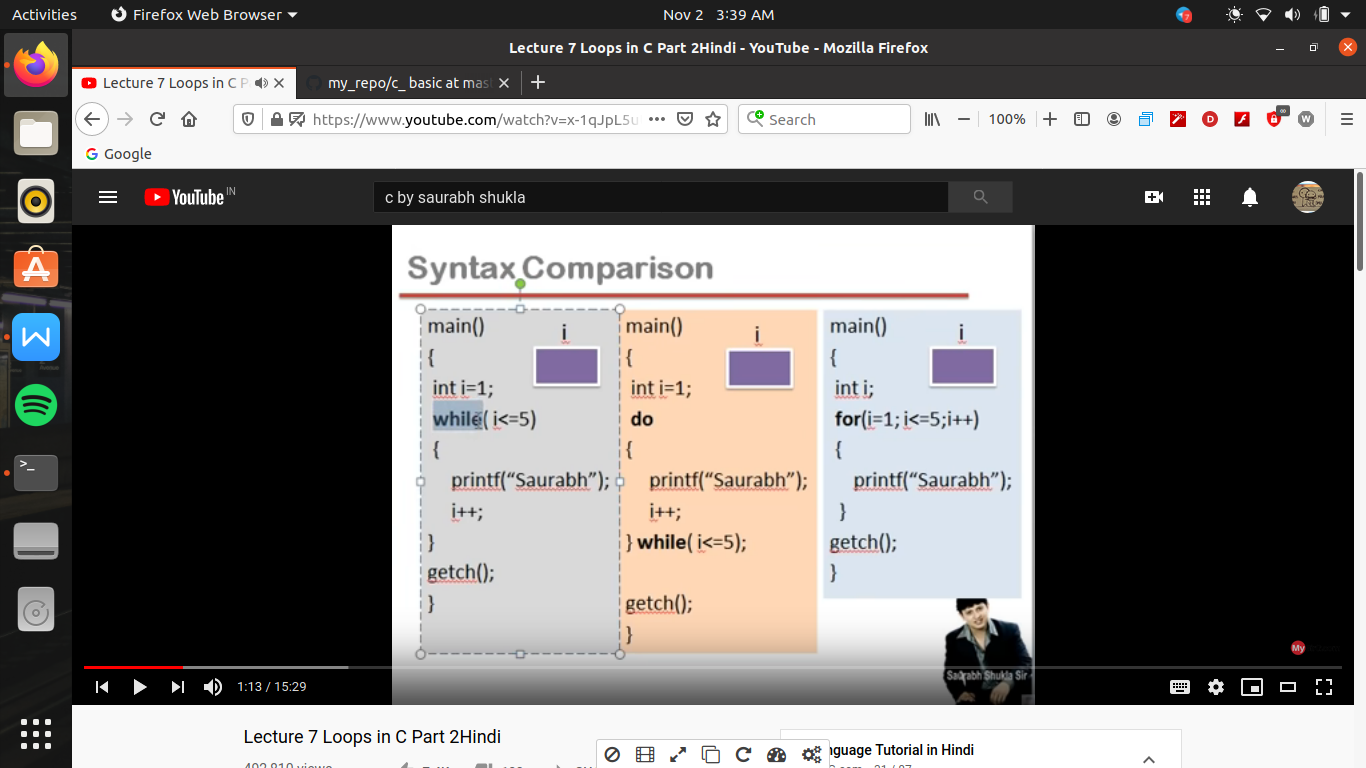
Loop in c language

It is also called iterative control instructions (iterative means repetative)

When we want to repeat some set of intruction in your program then we use loop

Their are three methods to apply loop :

1. while
2. do while
3. for



for ex-

syntax of while

while(condition)

{<this is while body what ever is intructions in give in the curly bracket(i.e while body) will repeat >

}

working of while when processor came to while then what ever be the condition is given was evaluated on the basis of true or false

and if statement is true then whatever is instruction is written in the while body will run

then after execution , then again processor go to the the while then again check the condition and if condition is true then it will again run the while body context or aisa jab tak hoga tab tak condition false nhi ho jati

so hame condition bhi bahot care se likhani hogi kyu ki agar wo condition constant rahegi toh , wo instruction never ending program ban jayega

so we can add atleast a variable in the condition

or while body me ek aisa instruction hona chahiye ki jab wo instruction ko dobara run kare toh variable ki value change ho jaye

#include<stdio.h>

int main()

{

int x=1; //counter variable ka initialization yaha se ho raha hai

while(x<=5) //here we use while

{

printf("hello \n"); //now here we want to print shashank 5 times for this either we have to write this statement 5 times or we use loop

++x; //thus x=x+1 we add this operator to increase the value of variables so that after a particular process the condition will yield false and the program will end , i.e jab dobara condition change ho tab x ki value change ho jani chahiye

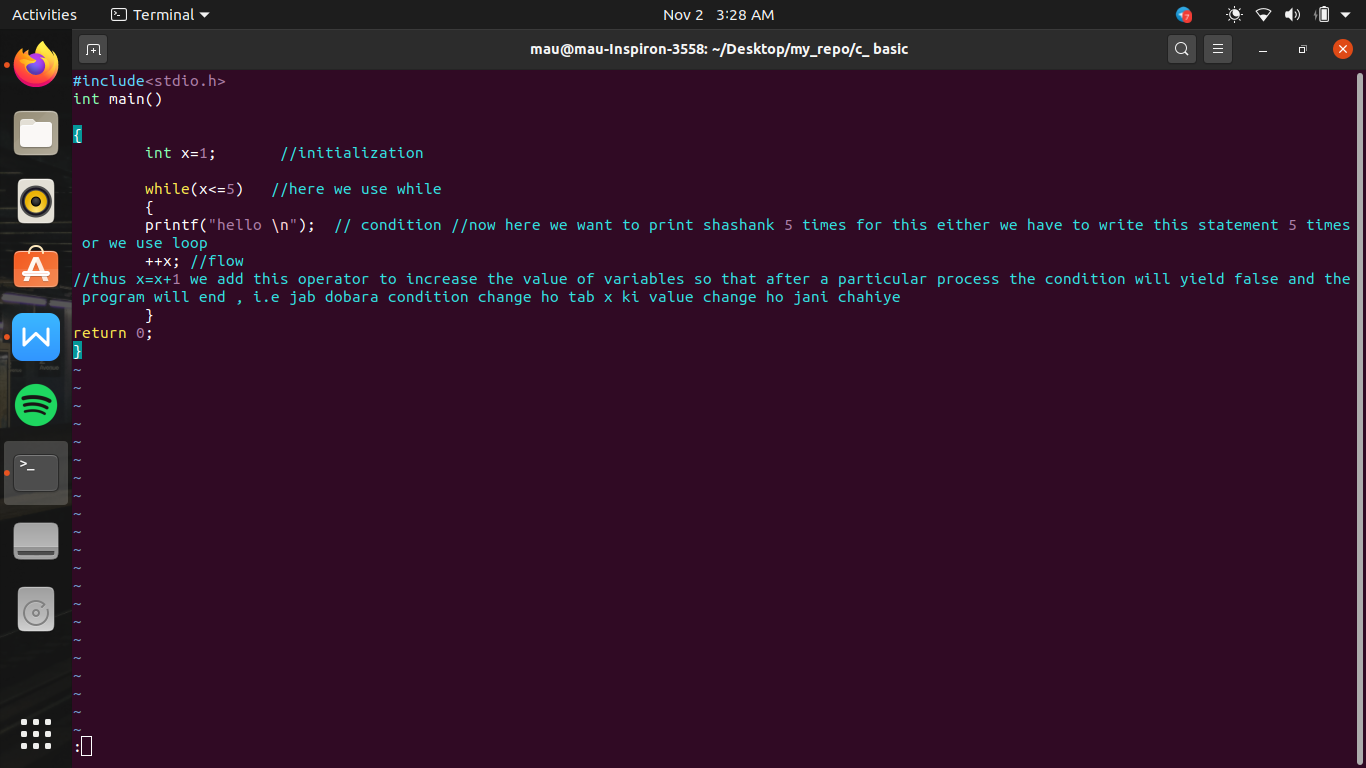
}

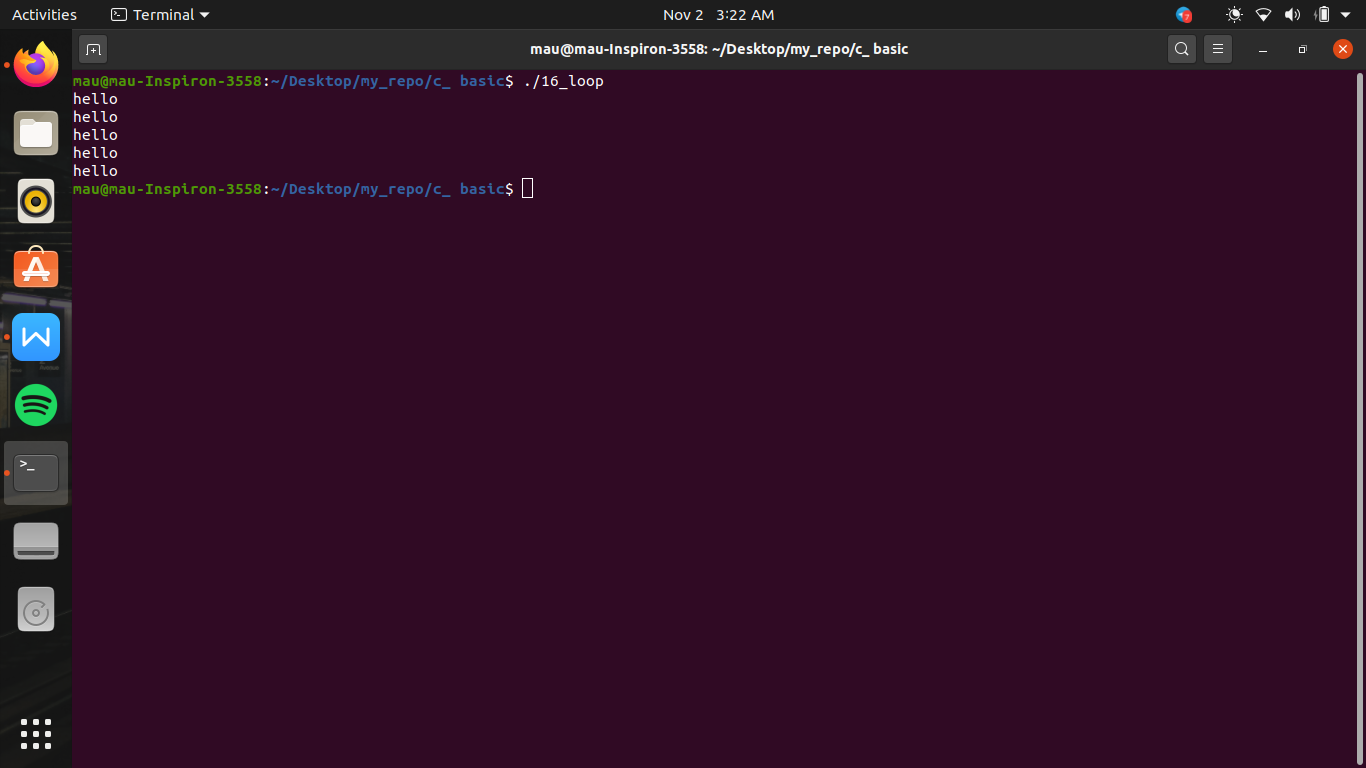
return 0;

}

(here in this while condition is checked by 6 times , 5 bar true and ek bar false)

while loop also called as entry control loop because block ke andar jayege hi kab jab condition true hogi





here x is a variable which control the loop thus it is also called as counter variable

dowhile ->

syntax

#include<stdio.h>

int main()

{

int x=3; //counter variable , initialise as value 1

do //by this processor directly go into the body of do and perform the intruction

{

printf("hello\n");

x++; //now after performing the instruction the value of variable is increased

}while(x<5); //here value of x become x+1 now here condition is checked if it is true then processorn then move to "do" the instruction again but if condition becomes false then the program will end

return 0;

}

(here the condition is checked by 1 less the condition checked by while loop)

here do-while loop exit control loop kahalata hai kyu ki block ke bahar a rahe hai tab condition check ho rahi hai es me agar condition starting me hi false ho toh bhi ek bar toh program kar dega

for loop ->

jayada programmer ko for loop hi pasand ata hai kyu ki operation ko control karane ke leye initialization , condition or flow yeh tino kam ek hi jagah ek sath kiye ja sakte hai

toh yeh readability ke hi sab se acha hai.

syntax

#include<stdio.h>

int main()

{

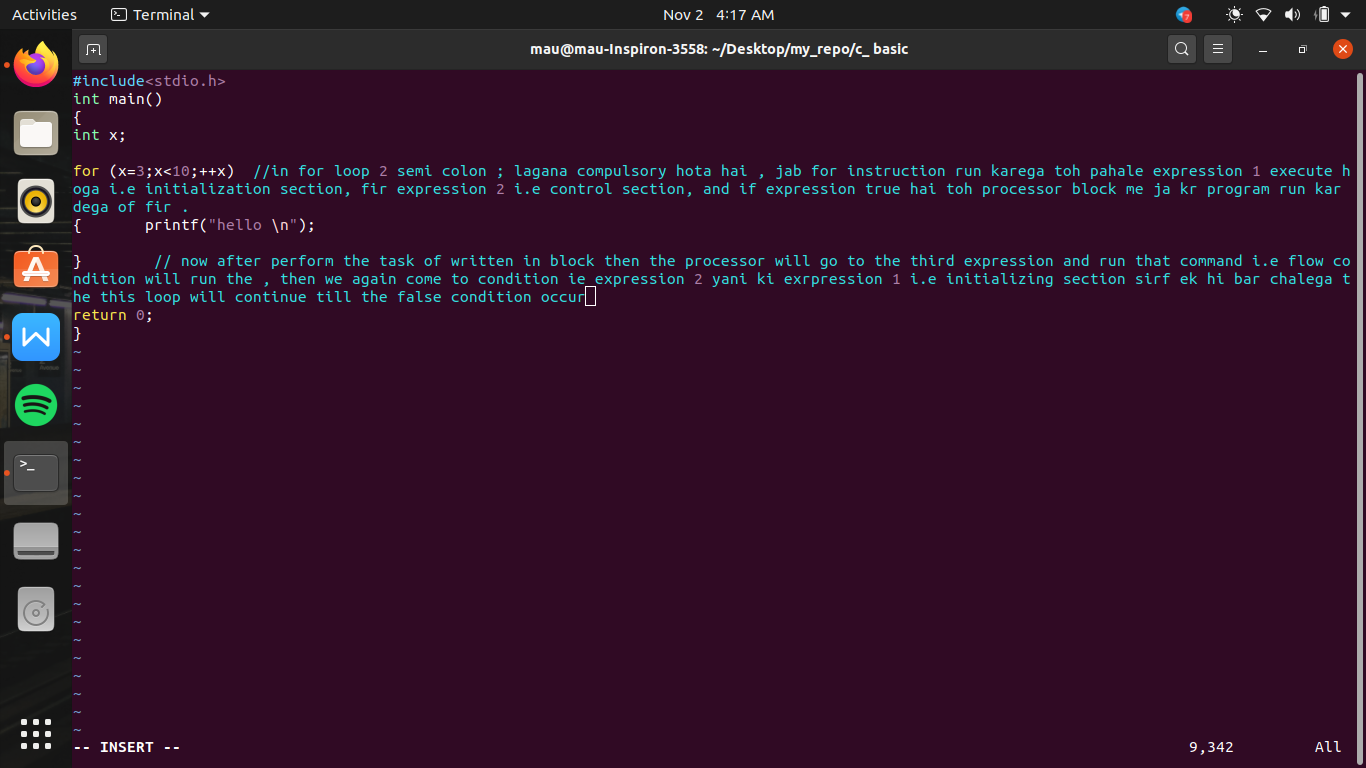
int x;

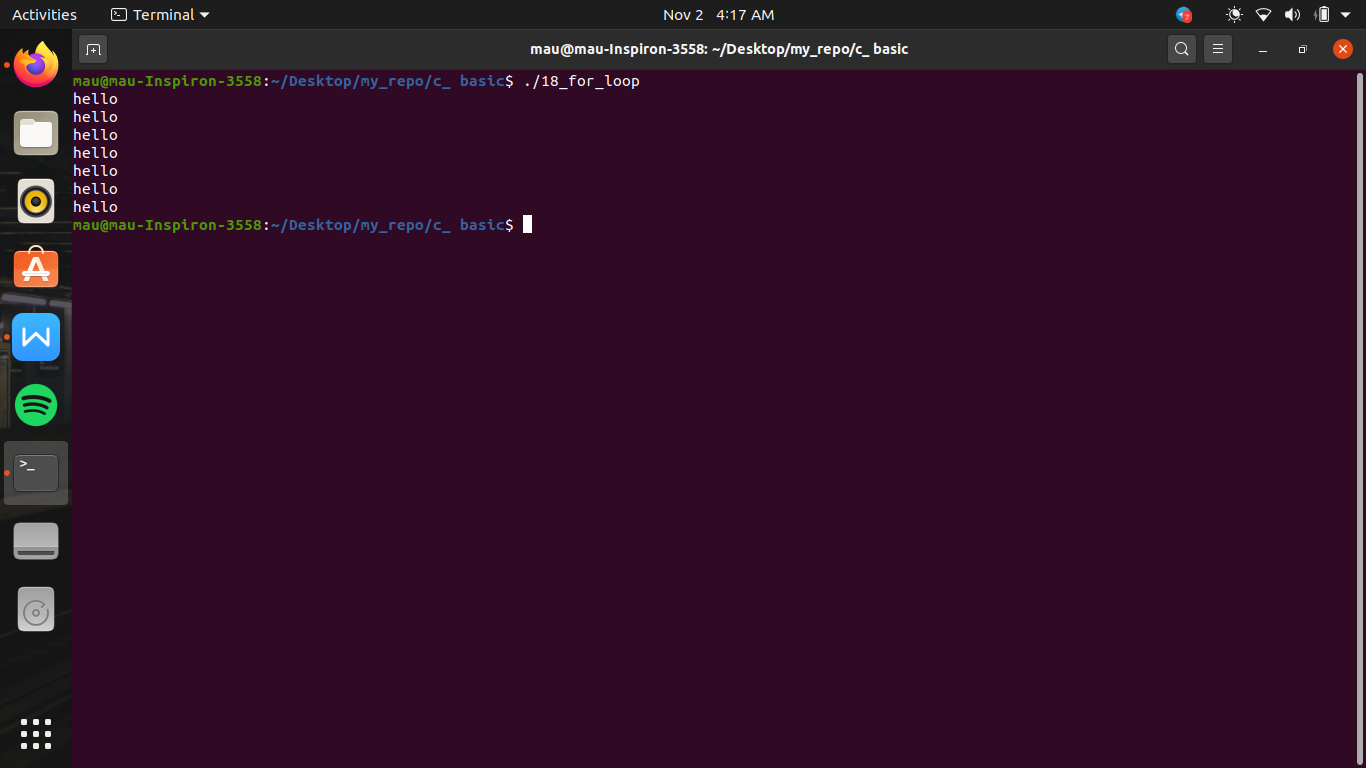
for (x=3;x<10;++x) //in for loop 2 semi colon ; lagana compulsory hota hai , jab for instruction run karega toh pahale expression 1 execute hoga i.e initialization section, fir expression 2 i.e control section, and if expression true hai toh processor block me ja kr program run kar dega of fir .

{ printf("hello \n");

} // now after perform the task of written in block then the processor will go to the third expression and run that command i.e flow condition will run the , then we again come to condition ie expression 2 yani ki exrpression 1 i.e initializing section sirf ek hi bar chalega the this loop will continue till the false condition occur return 0;

}

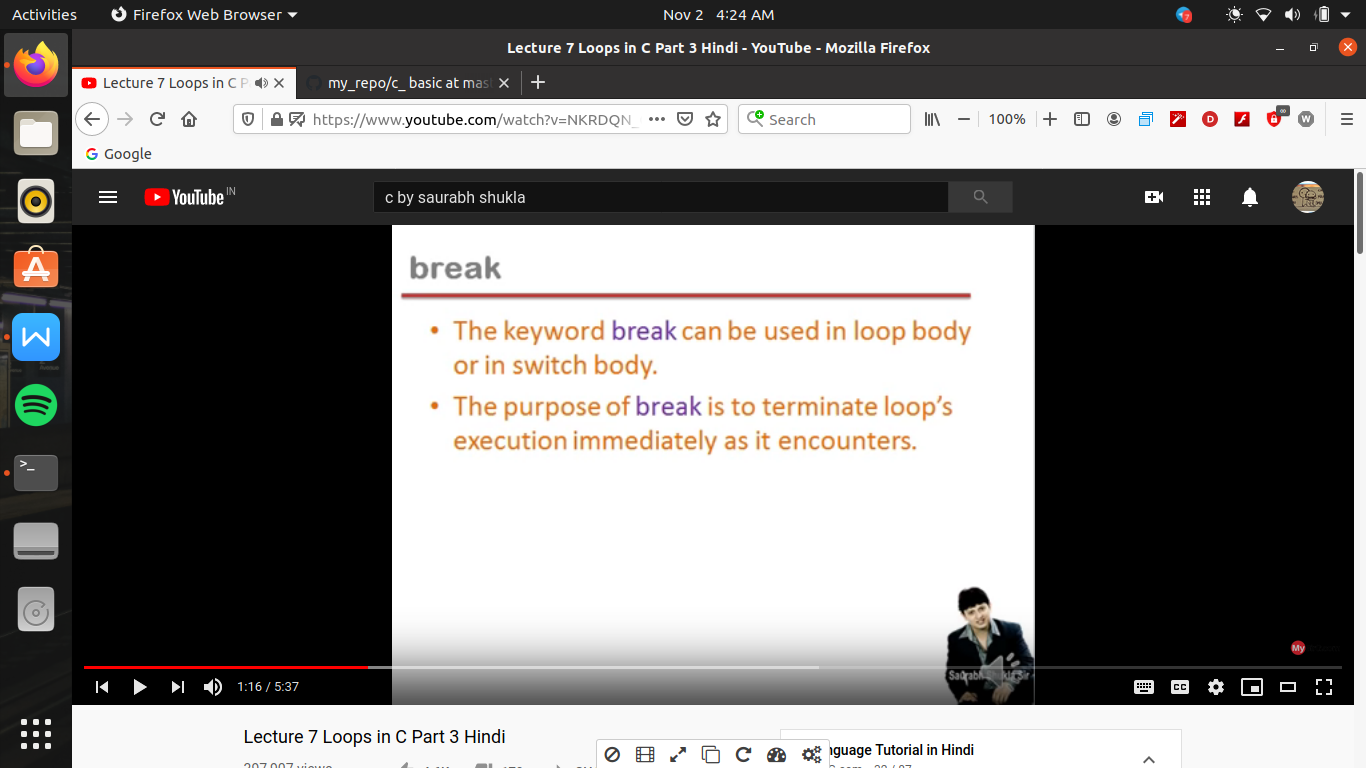




break keyword ->

break ka use can be possible only either in loop body or in switch body , inspite of it their is no other place to use it

break in loop word work as to terminate the work of loop as incomplete i.e beech me hi terminate kar deta hai



#include<stdio.h>

int main()

{

int x=1,y;

while(x<=5)

{

printf("enter a number \n");

scanf("%d",&y);

if(y<4) // break ka use waha hota hai jab hum pata toh ho kitani bar instruction run karwana hai but hum yeh bhi chahate ho kuch kuch special jagah pr loop ruk jaye

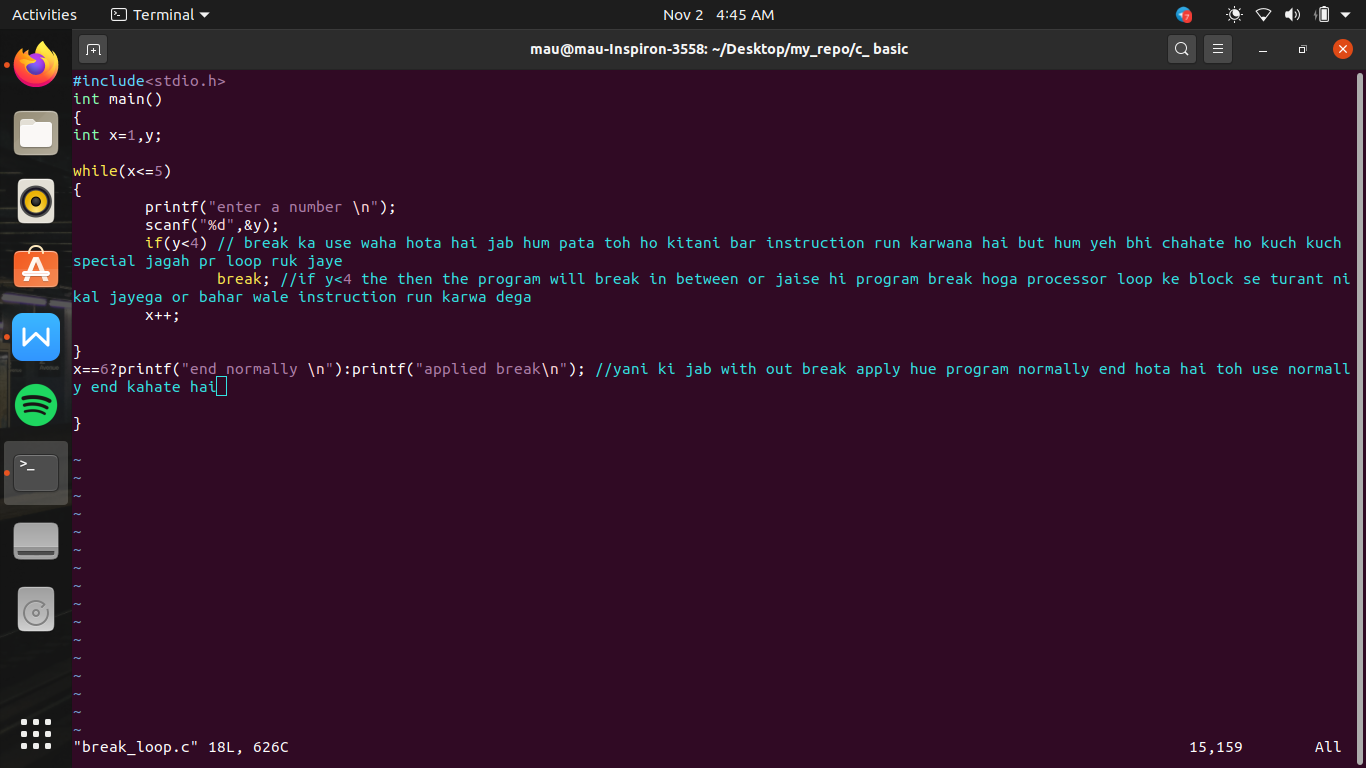
break; //if y<4 the then the program will break in between or jaise hi program break hoga processor loop ke block se turant nikal jayega or bahar wale instruction run karwa dega

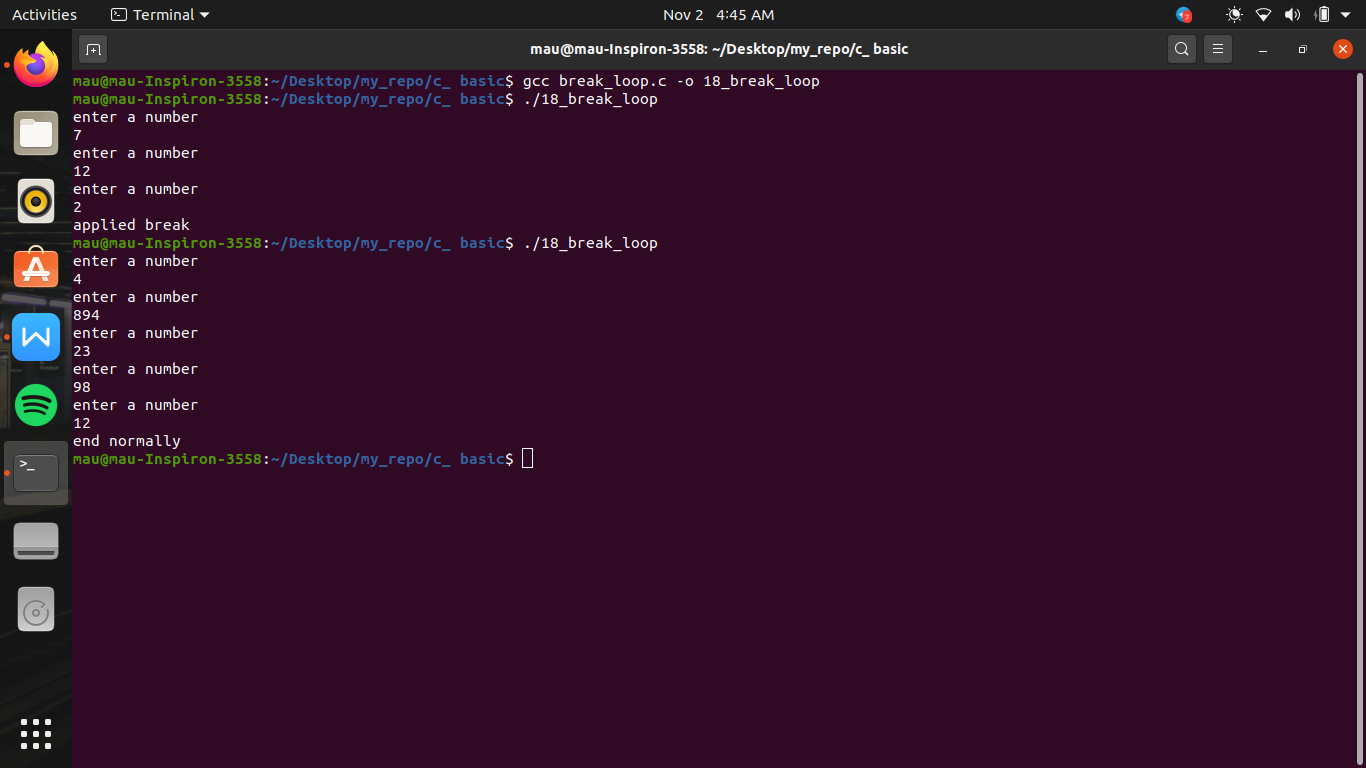
x++;

}

x==6?printf("end normally \n"):printf("applied break\n"); //yani ki jab with out break apply hue program normally end hota hai toh use normally end kahate hai

}





switch control ->

agar hamare pass bahot sare options ho then if-else or nasting of if-else suitable na ho toh waha switch control kam ate hai

switch control instruction ->

switch(expression)

{

case constant : code;

case constant : code;

case constant : code;

case constant : code;

default:code;

}

here we use 3 keywords switch, case, default.

in “if” expression written paranthis is called as condition and us ka evaluate true or false ke term me karate the

but in switch the expression written in the paranthis were not evaluated as true or false , balaki expression a ek result nikala jayega , calculate ho kar koe constant wale result mil jayegi

switch means , processor will jump to the location or case inside the block whose constant value of the case matches with the expression constant of switch and that code will run which is written with that case this code can be of single or also can be of multiple lines after executing that case then it will come to the next sequence case from top to bottom or agar hum chahate hai ki sirf wohi case run ho toh har case ke code ke bad we have to write break

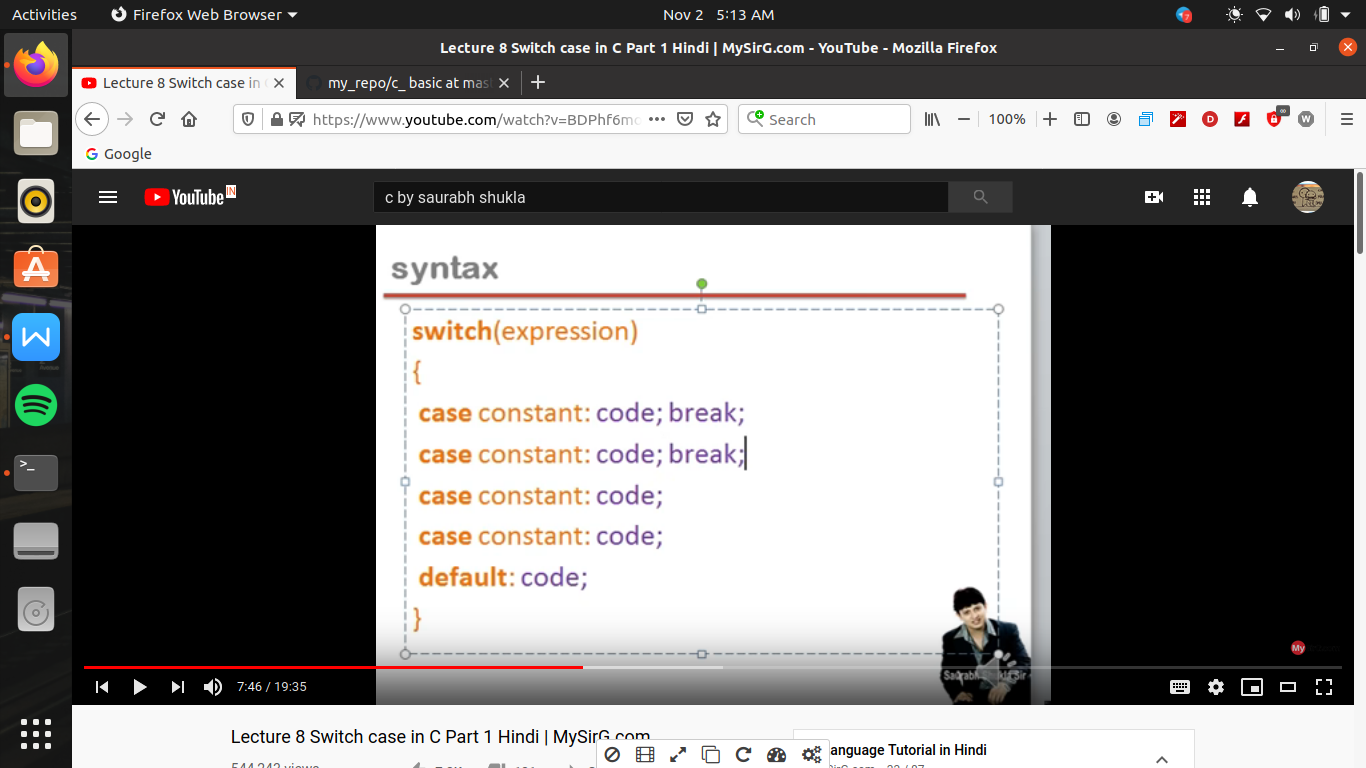
which take the processor to out of the switch block immediately

or job koe esa expression hota hai ki wo block me likhe kisi code se match nhi karata toh wo default ke pass ata hai or default ka code run kar deta hai

and after that control will come out of of switch block (as because it is not like a loop)only run one time

rule ->

1. har ek case me jo constant likha hai wo unique hona chahiye yani same nhi hona chahiye
2. this real value can be integer constant or a character constant but cannot be a real value
3. yeh zaruri nhi hai case me likhe constant sequence me ho



now here we making a menu driven program

in which we gave a menu to the user want ask to select their choice

jise me menu ke roop me 3 options hoge jaise

1. addition
2. odd -even
3. printing first n natural no.

#include<stdio.h>

int main()

{

int choice,x,y,z,p,q,b;

printf("select your choice \n 1. addition \n 2. odd-even \n 3. naural numbers\n");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("enter two numbers \n");

scanf("%d%d",&x,&y);

z=x+y;

printf("the sum of %d and %d is %d \n",x,y,z);

break;

case 2:

printf("enter a number \n");

scanf("%d",&p);

if(p%2==0?printf("the no. is even \n"):printf("the no. is odd \n"));

break;

case 3:

printf("enter a number \n ");

scanf("%d",&q);

for(b=1;b<=q;b++)

printf("%d \n",b);

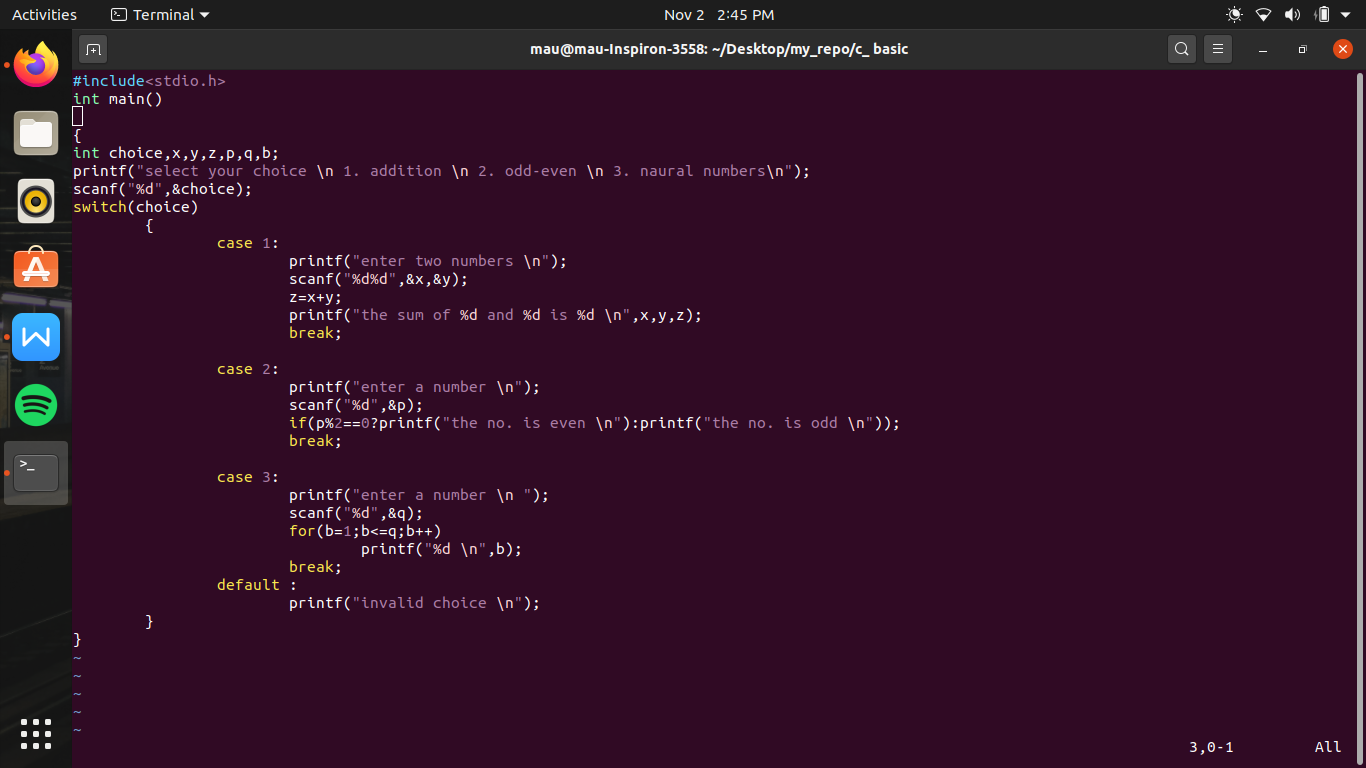
break;

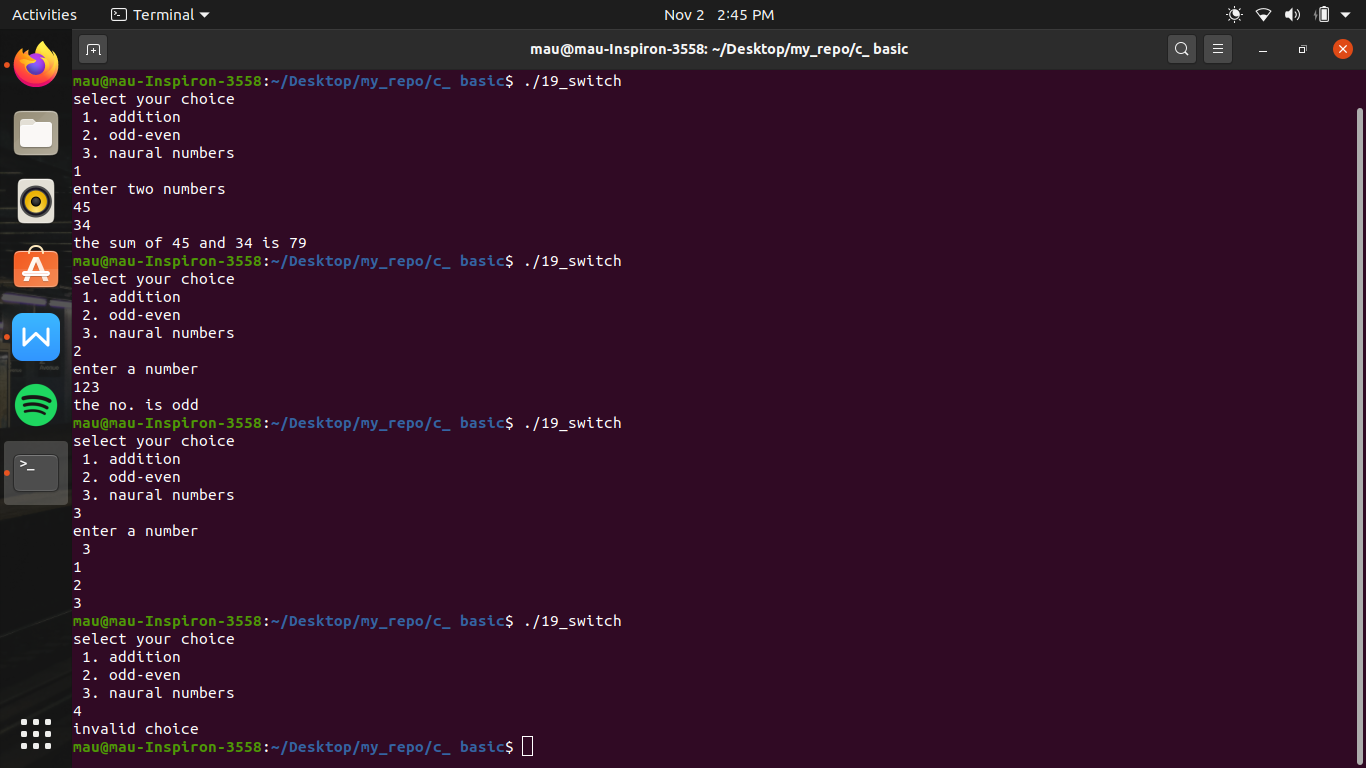
default :

printf("invalid choice \n");

}

}



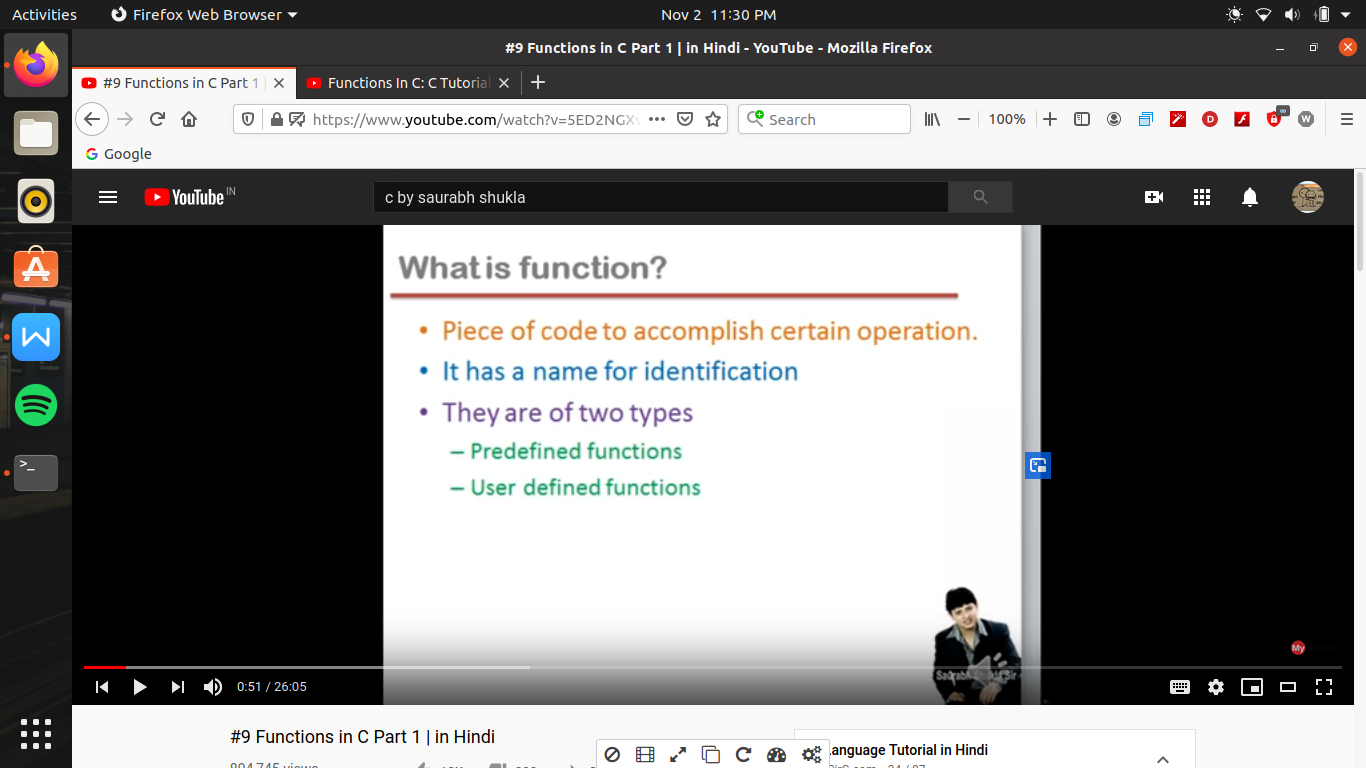


now here if we want that after selecting ones choice the program will again ask for another choice,

so for this we apply while loop in the whole program

Function in c ->

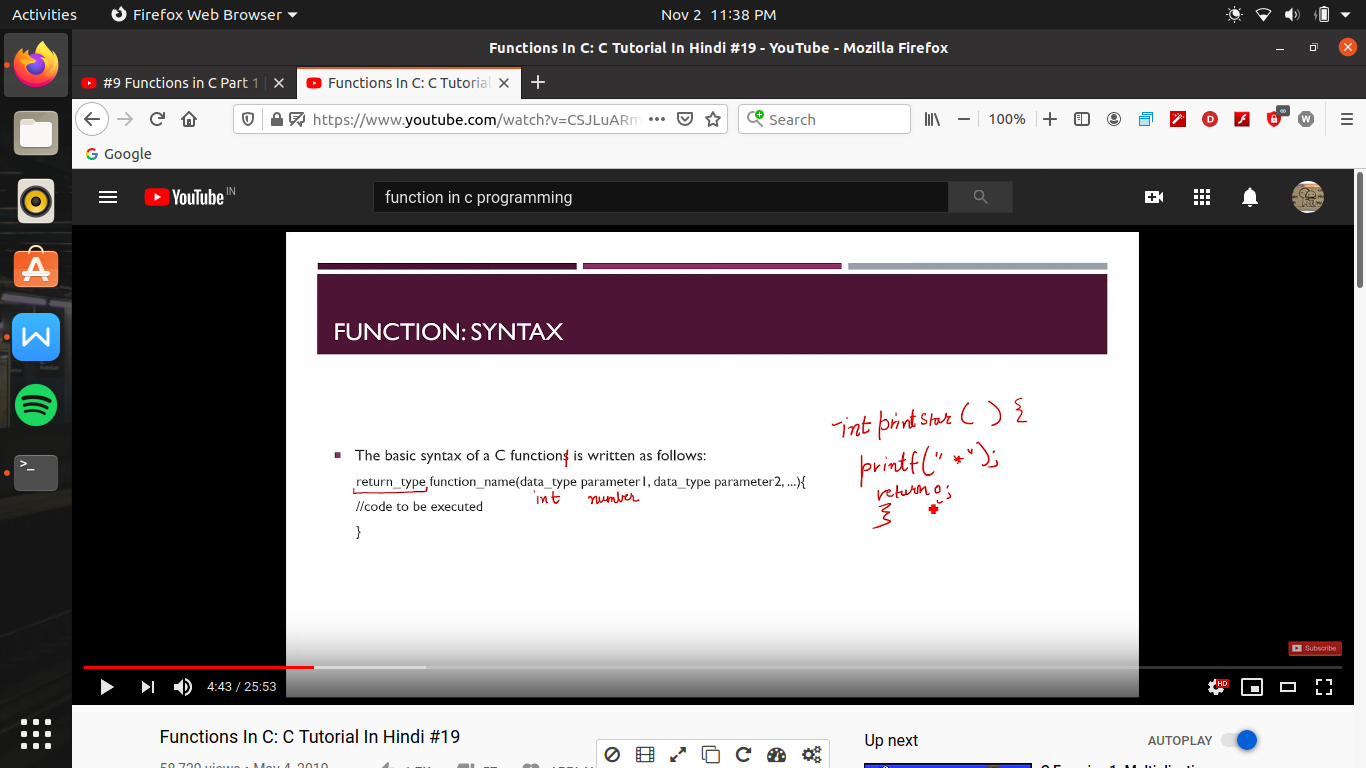
function basically is a piece of code or block of code jis ka ki kuch nam hota hai or es ka fayada yeh hai ki hum sirf us block ke nam se block ke andar ko represent kar sakate hai



function can be used to divide the large c program into pieces

or agar aisa ho ki ek program me 10 chije ho rahi hai or 10 chije me es 7 chij aisi h jo reapeat ho rahi hai toh hum us program ko 7 bar nhi likege sirf ek bar likh kr jab jarurat padegi toh use call kr lege

syntax



return\_type\_function\_name(data\_type\_parameter1, data\_type\_parameter2...)

{

//code to be executed

}

return<int>

ex- int print<name>(int x,1)

syntax

#include<stdio.h>

int main() //it is just the name of block which run

{

sum(); //here by only writing the name of the block , whole instruction in the block will run

even(); // here it is a user defined funtion as we defined the function //here we call the written funtion so we will say its is a function call

}

int sum() // here sum() represent the whole instruction of sum block

{

int x,y,z;

printf("enter two number \n");

scanf("%d%d",&x,&y);

z=x+y;

printf("sum of number is %d \n",z);

}

int even() //here even() represent the even block // a name given to function is known as function definition

{

int x,y;

printf("enter a number \n");

scanf("%d",&x);

if (x%2==0)

printf("the no. is even\n");

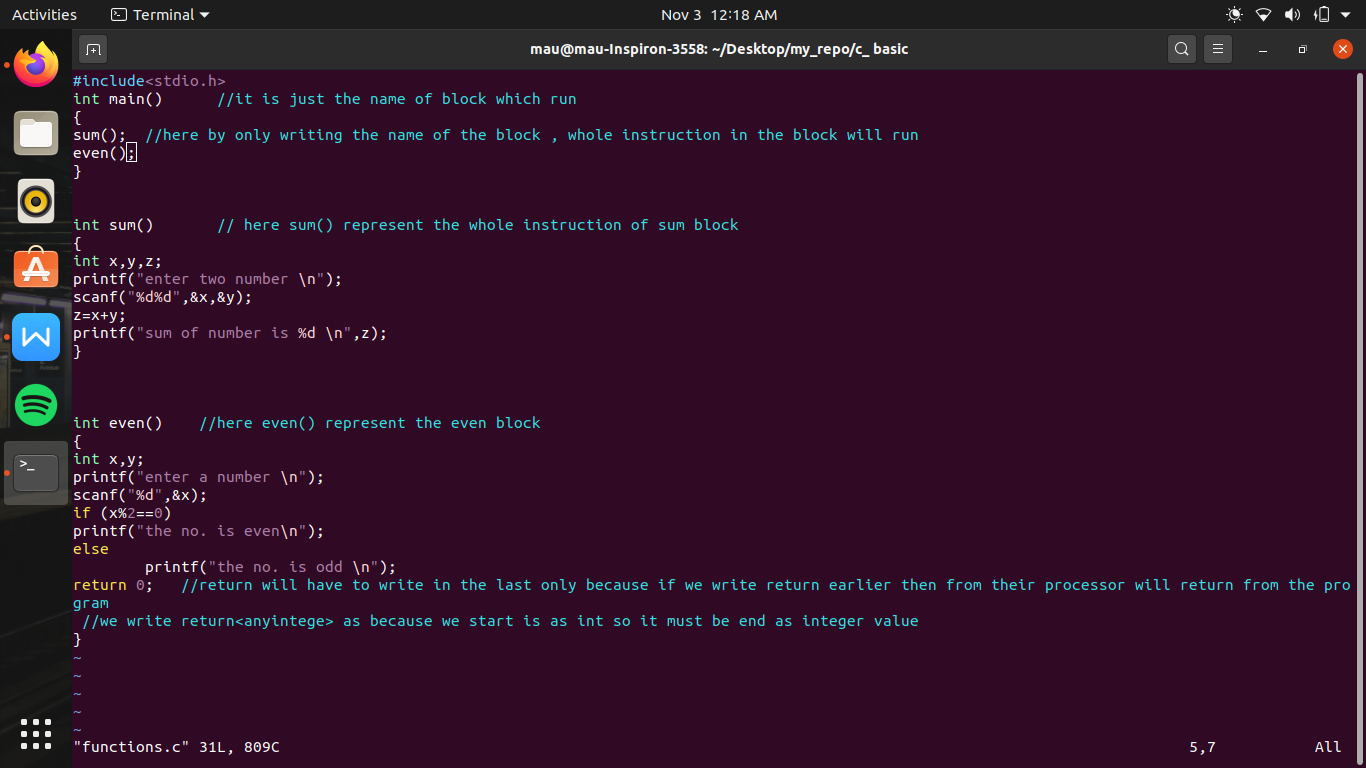
else

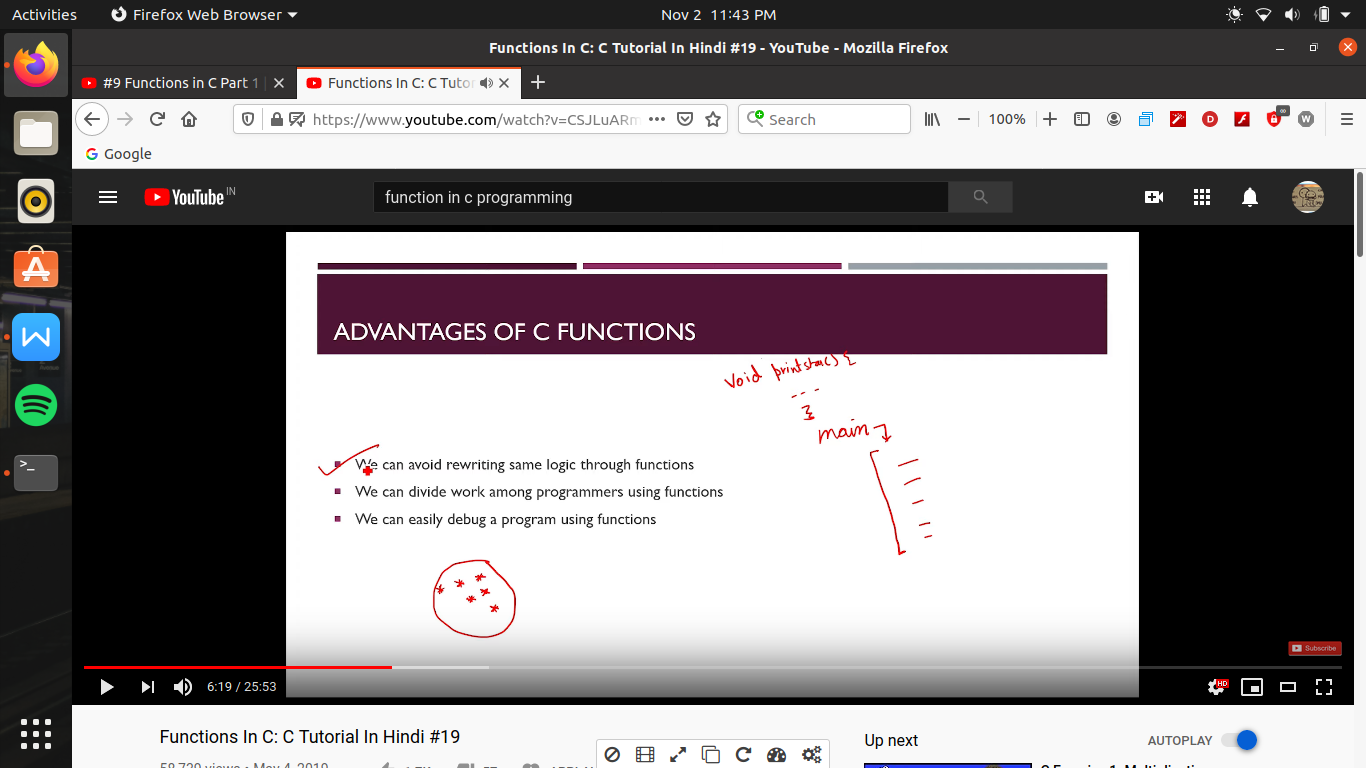
printf("the no. is odd \n");

return 0; //return will have to write in the last only because if we write return earlier then from their processor will return from the program

//we write return<anyintege> as because we start is as int so it must be end as integer value

}

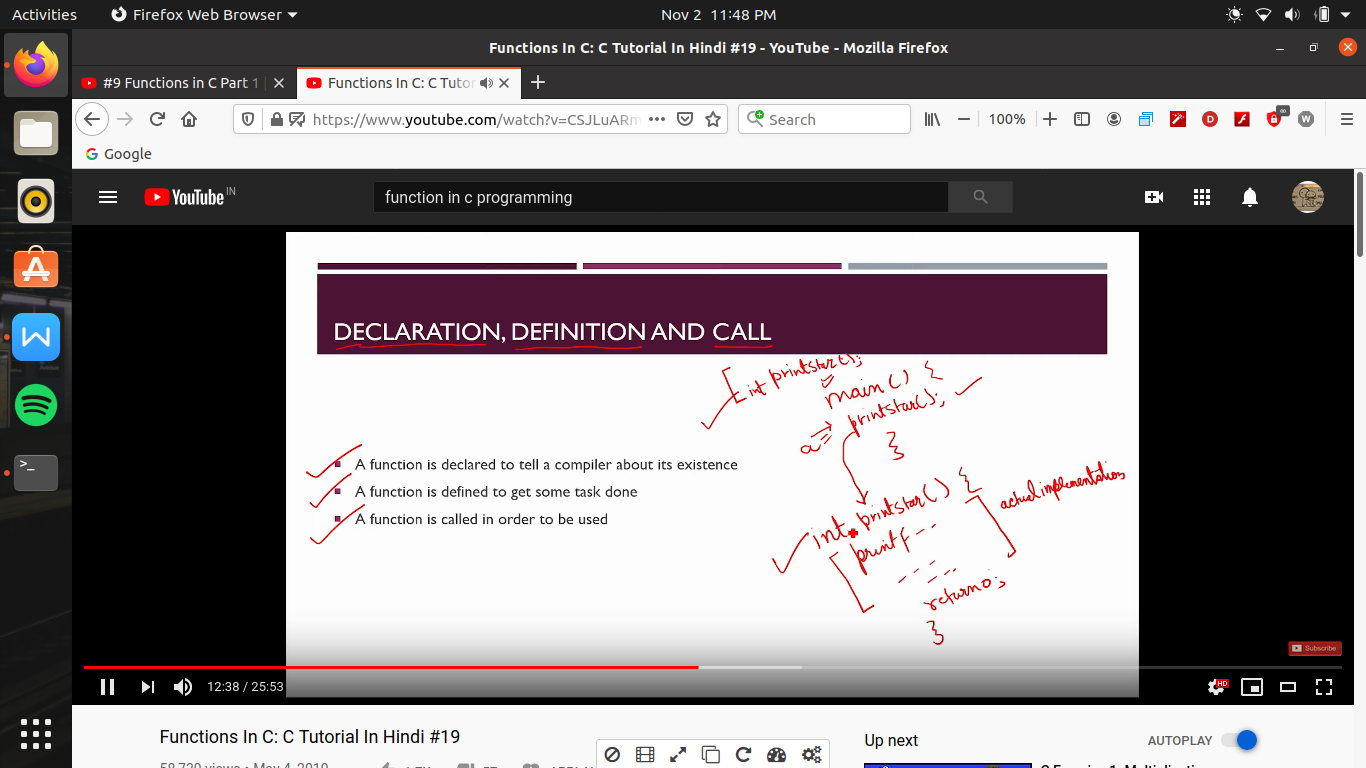




funtion definition ->

funtion declaration(funtion prototype) ->

and function call->



types of function :-

1. library function -> function included in c header files
2. user defined function -> functions created by c programmer to reduce complexity of a program

function code examples ->

1. without arguments and without return value
2. without arguments with return value
3. with arguments and without return value
4. with arguments and with return value

working of a program which execute more than one function

note -> ek program me chahe jitane function banaye main(); function se hi program ka execution hota hai

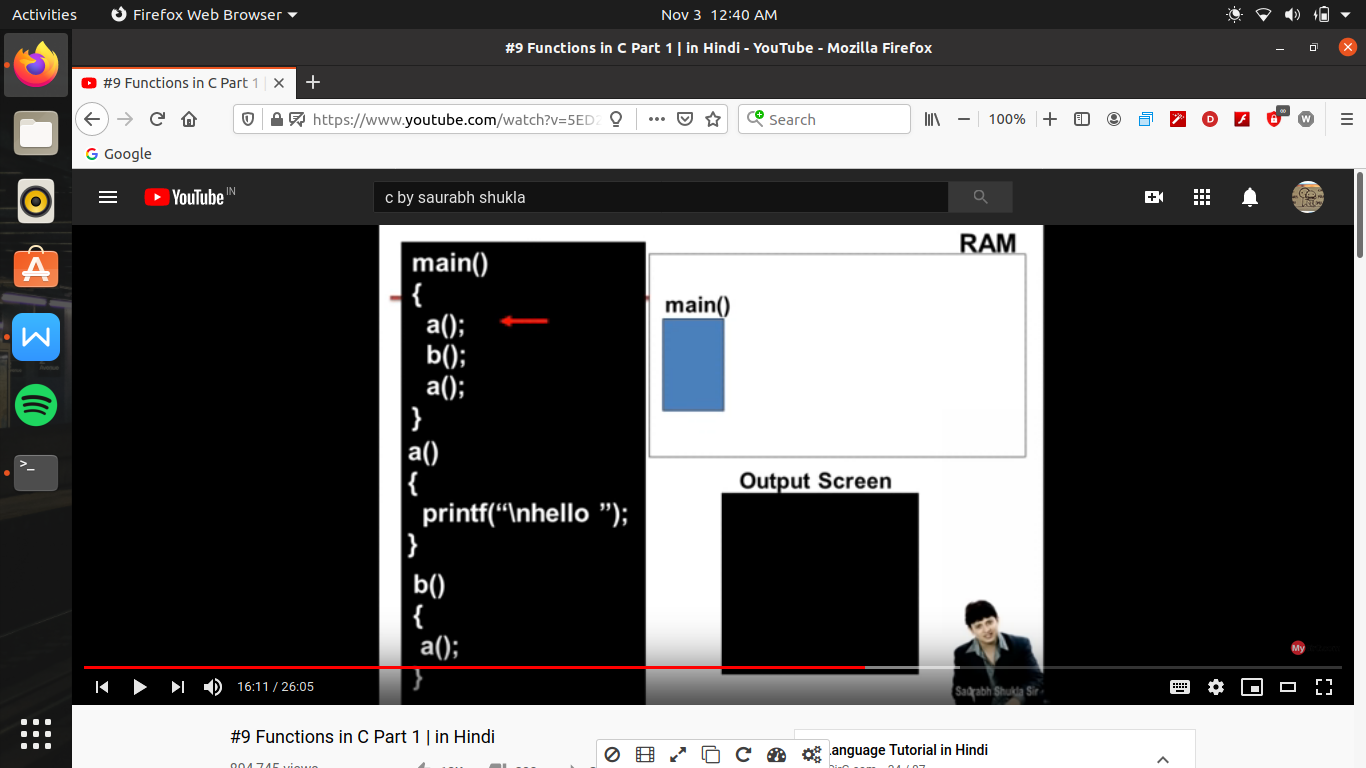
yani program me ek function ka nam main jarur rakhana hoga or har function ka nam alag alag rakhana bhi jaruri hai

and it is not compulsory that we have to defined the function in a particular sequence , yani koe bhi function kahi bhui likha ja sakata hai also main() function bhi

toh pahale main ko call kiya jayega operation system ke dawara and then a memory main function ko ram me jagah allocate kr di jayegi

or ram me memory 2 tarike ke leye milati hai ek toh jab use instruction rakhane ke leye or dusara jo data variables ko rakhane ke leye

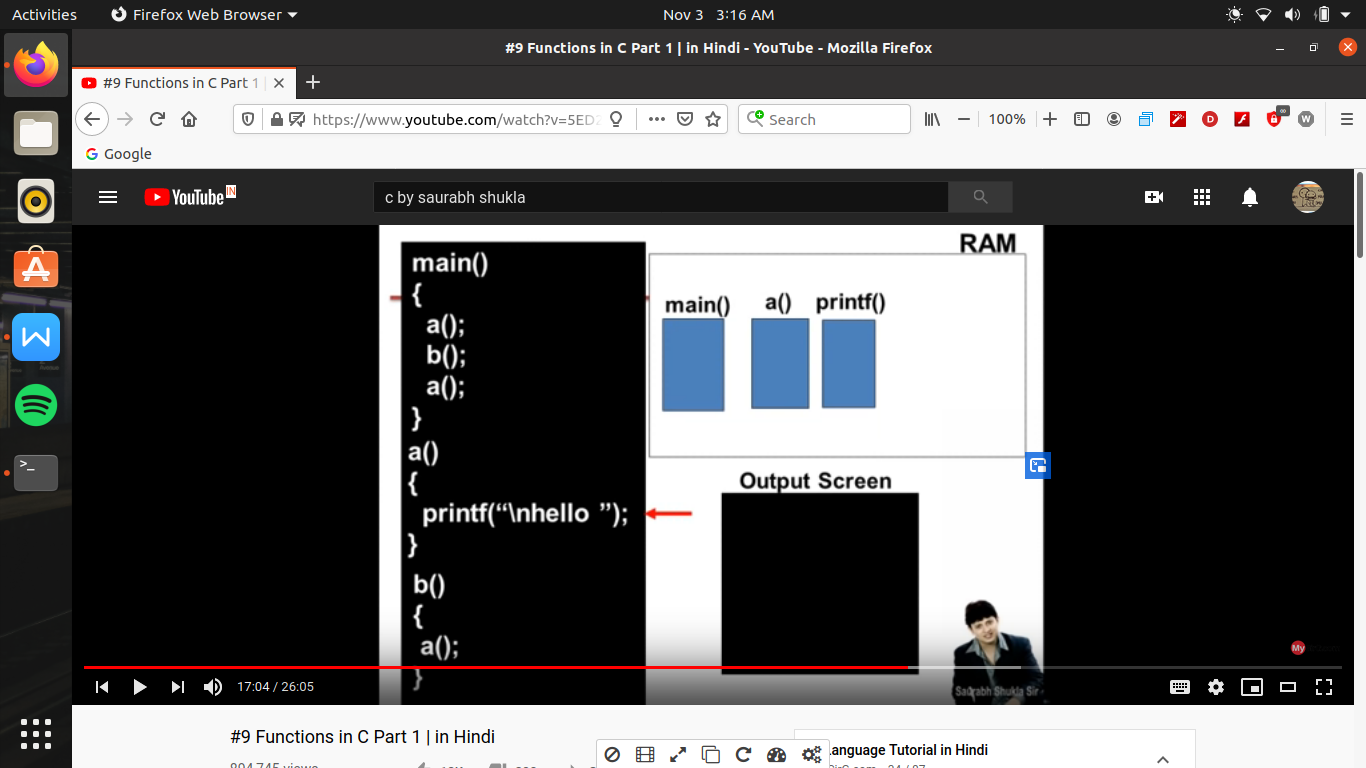
as because every function scope is different so ek function me bane variable dusara function use nhi kr sakata



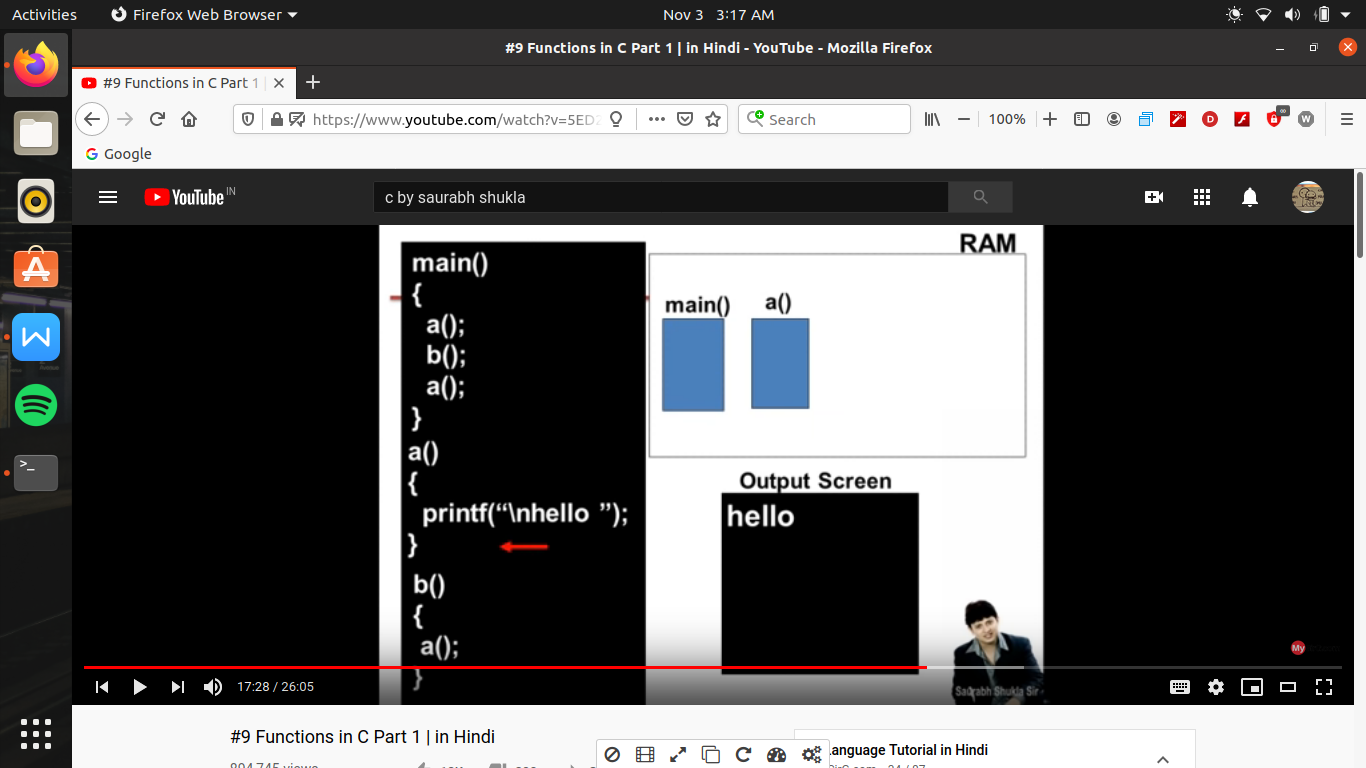
now when the function run and controls enter the body of the function then its run the first function written in the sequence it call the function and controller will directly jumpt to the block of definition of the function then that function will load in the ram

here main() function is pending jis ki coding abhi chal rahi hai us ki ek line abhi chal rahi hai jis ke karad a() function bhi ram me a gaya hai or us ki con\ding bhi abhi chl rahi hai

now a() call printf() function , printf function ek predefined function hai , toh ab printf() function bhi ram me a gaya

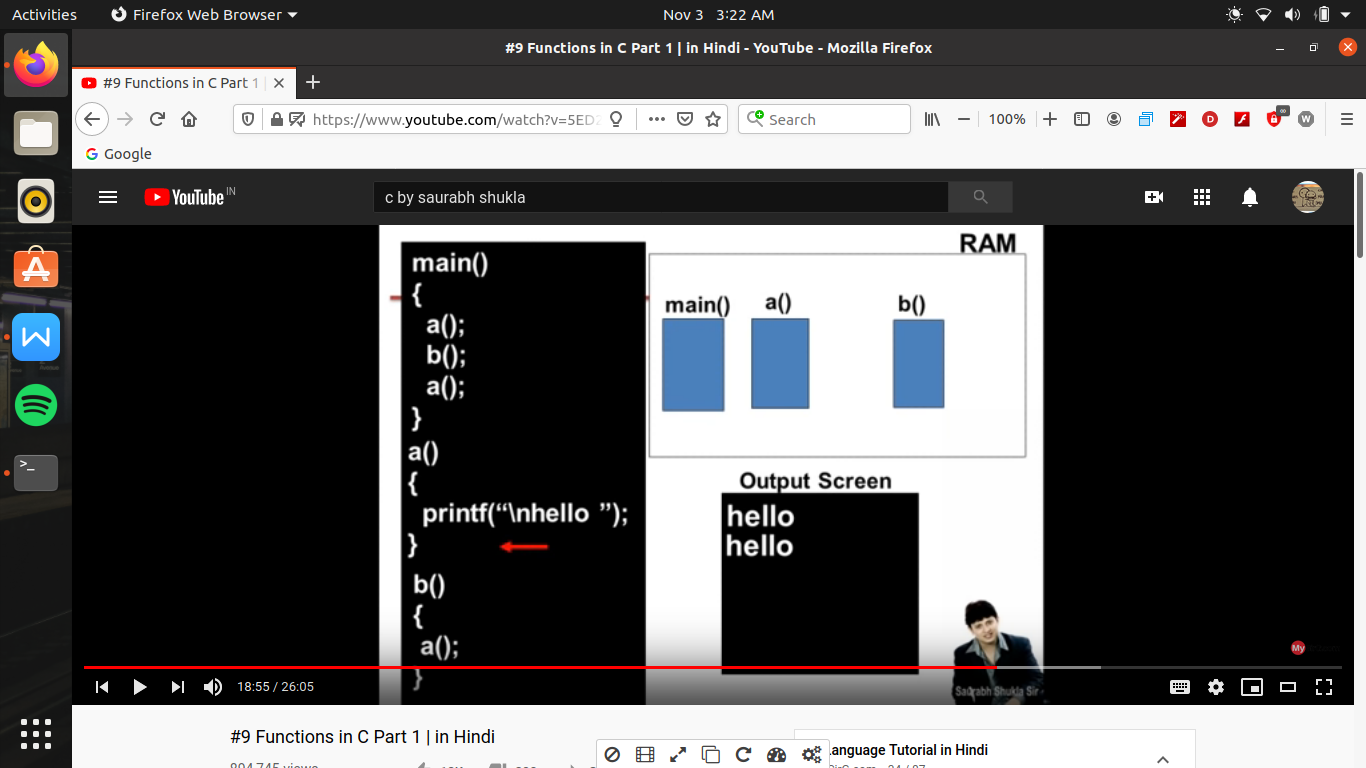


or jaisi hi prinf() execute ho jayega tab printf function ki memory release ho jayegi

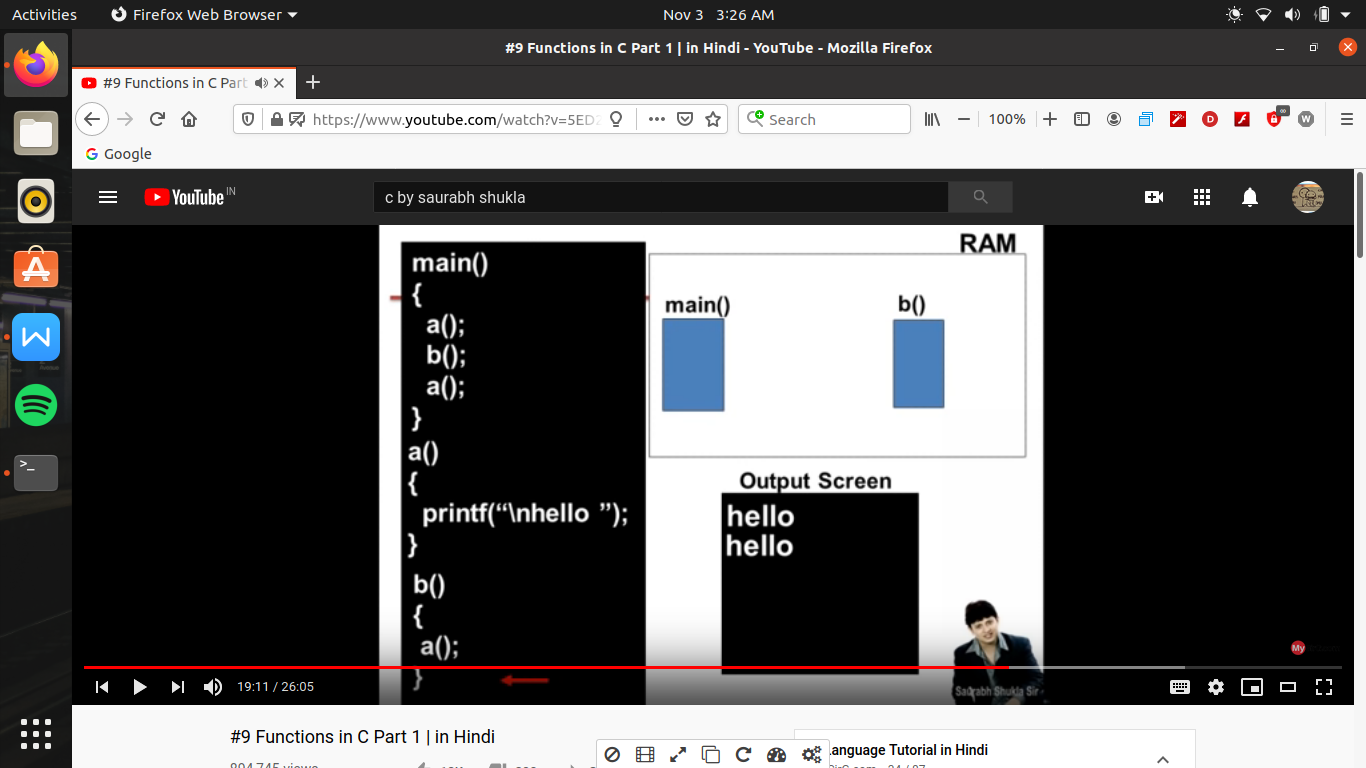


now a function end ho jayega or b() function ko call karega

ab jab b function execute ho raha hai toh fir b() , a () function ko call kar waha hai , koe bhi function kisi bhi function ko call kr sakata hai yeh jaruri nhi hai ki main hi sirf koe function ko call kare



toh ab a() function execute hoga fir controller b() ke end me jayega

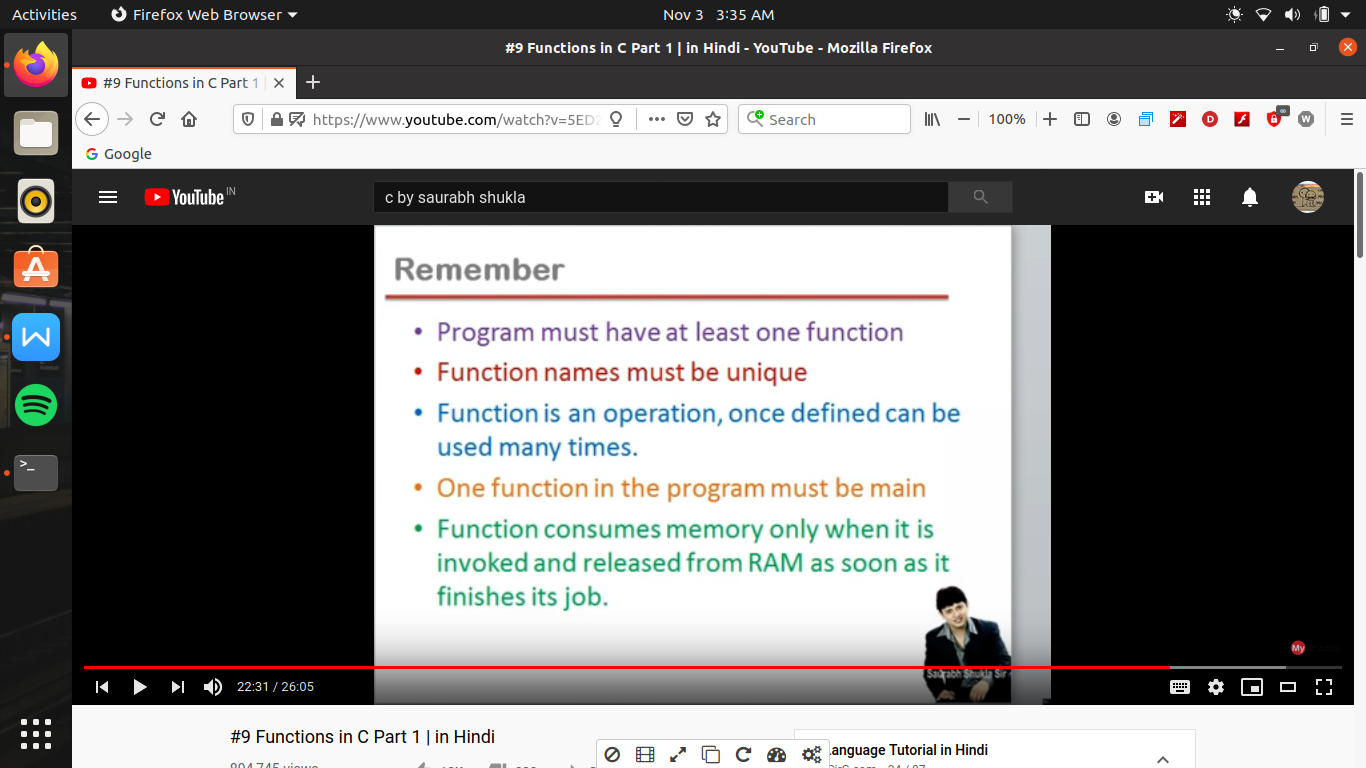


as b () function execute ho gaya now toh b() ki memory release ho jati hai and controller fir main () pe jayega jaha se wo b() me aya tha

agar yeh sab code hum sirf main me likh dete toh pura code ek bar me ram ki memory leleta jo ki function banane se ram ki usage optimize ho rahi hai , ram me sirf utani hi memory allocate hogi jo function() actaul me run kar raha hoga

point to remember ->

1. program must have at least one function
2. function name must be unique
3. function is an operation(i.e ek function main toh hona hi chahiye ), once defined can be used many times
4. one function is the program must be main
5. function consumes memory only when it is invoked and release from ram as soon as it finished its jobs



benefits of function->

1. modularization -> ek bade task ko several sub task me distribute kr dete hai
2. easy to read
3. easy to debug
4. easy to modify
5. avoids rewriting of same code over and over
6. better memory utilization

