Section : K19BY **Project No. : 2**

INT 213 : PYTHON PROGRAMMING

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

Submitted To : Ishan Kumar

Lovely Professional University

Jalandhar , Punjab , India

Project No: 2

Project Title:

Develop a GUI interface to convert number form one number system (binary, octal, decimal and hexadecimal) to other number system (binary, octal, decimal and hexadecimal) with explanation using Python**.**

Student Details:

Roll no.	Name	Registration no.
04	Aniket Kumar	11902717
05	Swaraj Senapati	11902728
55	Abhinav Yadav	11905597

There are infinite ways to represent a number. The four commonly associated with modern computers and digital electronics are: decimal, binary, octal, and hexadecimal.

Decimal (base 10) is the way most human beings represent numbers. Decimal is sometimes abbreviated as dec.

Decimal counting goes:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, and so on.

Binary (base 2) is the natural way most digital circuits represent and manipulate numbers. (Common misspellings are “bianary”, “bienary”, or “binery”.) Binary numbers are sometimes represented by preceding the value with '0b', as in 0b1011. Binary is sometimes abbreviated as bin.

Binary counting goes:
0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000, 10001, and so on.

Octal (base 8) was previously a popular choice for representing digital circuit numbers in a form that is more compact than binary. Octal is sometimes abbreviated as oct.

Octal counting goes:
0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, and so on.

Hexadecimal (base 16) is currently the most popular choice for representing digital circuit numbers in a form that is more compact than binary. (Common misspellings are “hexdecimal”, “hexidecimal”, “hexedecimal”, or “hexodecimal”.) Hexadecimal numbers are sometimes represented by preceding the value with '0x', as in 0x1B84. Hexadecimal is sometimes abbreviated as hex.

Hexadecimal counting goes:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, and so on.

All four number systems are equally capable of representing any number. Furthermore, a number can be perfectly converted between the various number systems without any loss of numeric value.

Binary Number Conversion

Binary to Octal

An easy way to convert from binary to octal is to group binary digits into sets of three, starting with the least significant (rightmost) digits.

Binary: 11100101 = 11 100 101

011 100 101 Pad the most significant digits with zeros if necessary to complete a group of three.

Then, look up each group in a table:

Binary:	000	001	010	011	100	101	110	111
Octal:	0	1	2	3	4	5	6	7

Binary = 011 100 101

Octal = 3 4 5 = 345 oct

Binary to Hexadecimal

An equally easy way to convert from binary to hexadecimal is to group binary digits into sets of four, starting with the least significant (rightmost) digits.

Binary: 11100101 = 1110 0101

Then, look up each group in a table:

Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	0	1	2	3	4	5	6	7
Binary:	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal:	8	9	A	B	C	D	E	F

Binary = 1110 0101

Hexadecimal = E 5 = E5 hex

Binary to Decimal

They say there are only 10 people in this world: those that understand binary and those that don't. Ha ha.

If you don't get that joke, you'll need a method to convert from binary to decimal. One method involves addition and multiplication.

1. Start the decimal result at 0.
2. Remove the most significant binary digit (leftmost) and add it to the result.
3. If all binary digits have been removed, you're done. Stop.
4. Otherwise, multiply the result by 2.
5. Go to step 2.

Here is an example of converting 11100000000 binary to decimal:

Binary Digits	Operation	Decimal Result	Operation	Decimal Result
11100000000	+1	1	× 2	2
1100000000	+1	3	× 2	6
100000000	+1	7	× 2	14
00000000	+0	14	× 2	28
0000000	+0	28	× 2	56
000000	+0	56	× 2	112
00000	+0	112	× 2	224
0000	+0	224	× 2	448
000	+0	448	× 2	896
00	+0	896	× 2	1792
0	+0	1792	done.	

Decimal Number Conversion

A repeated division and remainder algorithm can convert decimal to binary, octal, or hexadecimal.

1. Divide the decimal number by the desired target radix (2, 8, or 16).
2. Append the remainder as the next most significant digit.
3. Repeat until the decimal number has reached zero.

Decimal to Binary

Here is an example of using repeated division to convert 1792 decimal to binary:

Decimal Number	Operation	Quotient	Remainder	Binary Result
1792	÷ 2 =	896	0	0
896	÷ 2 =	448	0	00
448	÷ 2 =	224	0	000
224	÷ 2 =	112	0	0000
112	÷ 2 =	56	0	00000
56	÷ 2 =	28	0	000000
28	÷ 2 =	14	0	0000000
14	÷ 2 =	7	0	00000000
7	÷ 2 =	3	1	100000000
3	÷ 2 =	1	1	1100000000
1	÷ 2 =	0	1	11100000000
0	done.			

Decimal to Octal

Here is an example of using repeated division to convert 1792 decimal to octal:

Decimal Number	Operation	Quotient	Remainder	Octal Result
1792	÷ 8 =	224	0	0
224	÷ 8 =	28	0	00
28	÷ 8 =	3	4	400
3	÷ 8 =	0	3	3400
0	done.			

Decimal to Hexadecimal

Here is an example of using repeated division to convert 1792 decimal to hexadecimal:

Decimal Number	Operation	Quotient	Remainder	Hexadecimal Result
1792	÷ 16 =	112	0	0
112	÷ 16 =	7	0	00
7	÷ 16 =	0	7	700
0	done.			

The only addition to the algorithm when converting from decimal to hexadecimal is that a table must be used to obtain the hexadecimal digit if the remainder is greater than decimal 9.

Decimal:	0	1	2	3	4	5	6	7
Hexadecimal:	0	1	2	3	4	5	6	7
Decimal:	8	9	10	11	12	13	14	15
Hexadecimal:	8	9	A	B	C	D	E	F

The addition of letters can make for funny hexadecimal values. For example, 48879 decimal converted to hex is:

Decimal Number	Operation	Quotient	Remainder	Hexadecimal Result
48879	÷ 16 =	3054	15	F
3054	÷ 16 =	190	14	EF
190	÷ 16 =	11	14	EEF
11	÷ 16 =	0	11	BEEF
0	done.			

Other fun hexadecimal numbers include: AD, BE, FAD, FADE, ADD, BED, BEE, BEAD, DEAF, FEE, ODD, BOD, DEAD, DEED, BABE, CAFE, CoFFEE, FED, FEED, FACE, BAD, FooD, and my initials DAC.

Octal Number Conversion

Octal to Binary

Converting from octal to binary is as easy as converting from binary to octal. Simply look up each octal digit to obtain the equivalent group of three binary digits.

Octal:	0	1	2	3	4	5	6	7
Binary:	000	001	010	011	100	101	110	111

Octal = 3 4 5

Binary = 011 100 101 = 011100101 binary

Octal to Hexadecimal

When converting from octal to hexadecimal, it is often easier to first convert the octal number into binary and then from binary into hexadecimal. For example, to convert 345 octal into hex:

(from the previous example)

Octal = 3 4 5

Binary = 011 100 101 = 011100101 binary

Drop any leading zeros or pad with leading zeros to get groups of four binary digits (bits):
Binary 011100101 = 1110 0101

Then, look up the groups in a table to convert to hexadecimal digits.

Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	0	1	2	3	4	5	6	7
Binary:	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal:	8	9	A	B	C	D	E	F

Binary = 1110 0101

Hexadecimal = E 5 = E5 hex

Therefore, through a two-step conversion process, octal 345 equals binary 011100101 equals hexadecimal E5.

Octal to Decimal

Converting octal to decimal can be done with repeated division.

1. Start the decimal result at 0.
2. Remove the most significant octal digit (leftmost) and add it to the result.
3. If all octal digits have been removed, you're done. Stop.
4. Otherwise, multiply the result by 8.
5. Go to step 2.

Octal Digits	Operation	Decimal Result	Operation	Decimal Result
345	+3	3	× 8	24
45	+4	28	× 8	224
5	+5	229	done.	

The conversion can also be performed in the conventional mathematical way, by showing each digit place as an increasing power of 8.

345 octal = $(3 * 8^2) + (4 * 8^1) + (5 * 8^0) = (3 * 64) + (4 * 8) + (5 * 1) = 229$ decimal

Hexadecimal Number Conversion

Hexadecimal to Binary

Converting from hexadecimal to binary is as easy as converting from binary to hexadecimal. Simply look up each hexadecimal digit to obtain the equivalent group of four binary digits.

Hexadecimal:	0	1	2	3	4	5	6	7
Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	8	9	A	B	C	D	E	F
Binary:	1000	1001	1010	1011	1100	1101	1110	1111

Hexadecimal = A 2 D E

Binary = 1010 0010 1101 1110 = 1010001011011110 binary

Hexadecimal to Octal

When converting from hexadecimal to octal, it is often easier to first convert the hexadecimal number into binary and then from binary into octal. For example, to convert A2DE hex into octal:

(from the previous example)

Hexadecimal = A 2 D E

Binary = 1010 0010 1101 1110 = 1010001011011110 binary

Add leading zeros or remove leading zeros to group into sets of three binary digits.

Binary: 1010001011011110 = 001 010 001 011 011 110

Then, look up each group in a table:

Binary:	000	001	010	011	100	101	110	111
Octal:	0	1	2	3	4	5	6	7

Binary = 001 010 001 011 011 110

Octal = 1 2 1 3 3 6 = 121336 octal

Therefore, through a two-step conversion process, hexadecimal A2DE equals binary 1010001011011110 equals octal 121336.

Hexadecimal to Decimal

Converting hexadecimal to decimal can be performed in the conventional mathematical way, by showing each digit place as an increasing power of 16. Of course, hexadecimal letter values need to be converted to decimal values before performing the math.

Hexadecimal:	0	1	2	3	4	5	6	7
Decimal:	0	1	2	3	4	5	6	7
Hexadecimal:	8	9	A	B	C	D	E	F
Decimal:	8	9	10	11	12	13	14	15

A2DE hexadecimal:
$= ((A) * 16^3) + (2 * 16^2) + ((D) * 16^1) + ((E) * 16^0)$
$= (10 * 16^3) + (2 * 16^2) + (13 * 16^1) + (14 * 16^0)$
$= (10 * 4096) + (2 * 256) + (13 * 16) + (14 * 1)$
$= 40960 + 512 + 208 + 14$
= 41694 decimal

Tools used in the project:

Tkinter :-

tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Import the Tkinter module.

Defining a Function :-

You can define functions to provide the required functionality. Here are simple rules to define a function in Python. Function blocks begin with the keyword def followed by the function name and parentheses (( )). Any input parameters or arguments should be placed within these parentheses.

Try and except :-

The try block lets you test a block of code for errors. The except block lets you handle the error. The finally block lets you execute code, regardless of the result of the try- and except blocks.

Entry widget :-

The Entry widget is used to accept single-line text strings from a user.

Label widget:-

The Label is used to specify the container box where we can place the text or images. This widget is used to provide the message to the user about other widgets used in the python application. There are the various options which can be specified to configure the text or the part of the text shown in the Label.

Radio button :-

The Radiobutton is a standard tkinter widget used to implement one-of-many selections.

Frame:-

Python Tkinter Frame widget is used to organize the group of widgets. It acts like a container which can be used to hold the other widgets

Work Done By Aniket Kumar (Roll no. 04) :-

- Written the code for the conversion i.e. from one number form to another one.
- Created the structure of the GUI.
- Written the code of how input are taken from the user to perform the task.
- Created the Radio button and join it to the result part.
- Given the description of the report.

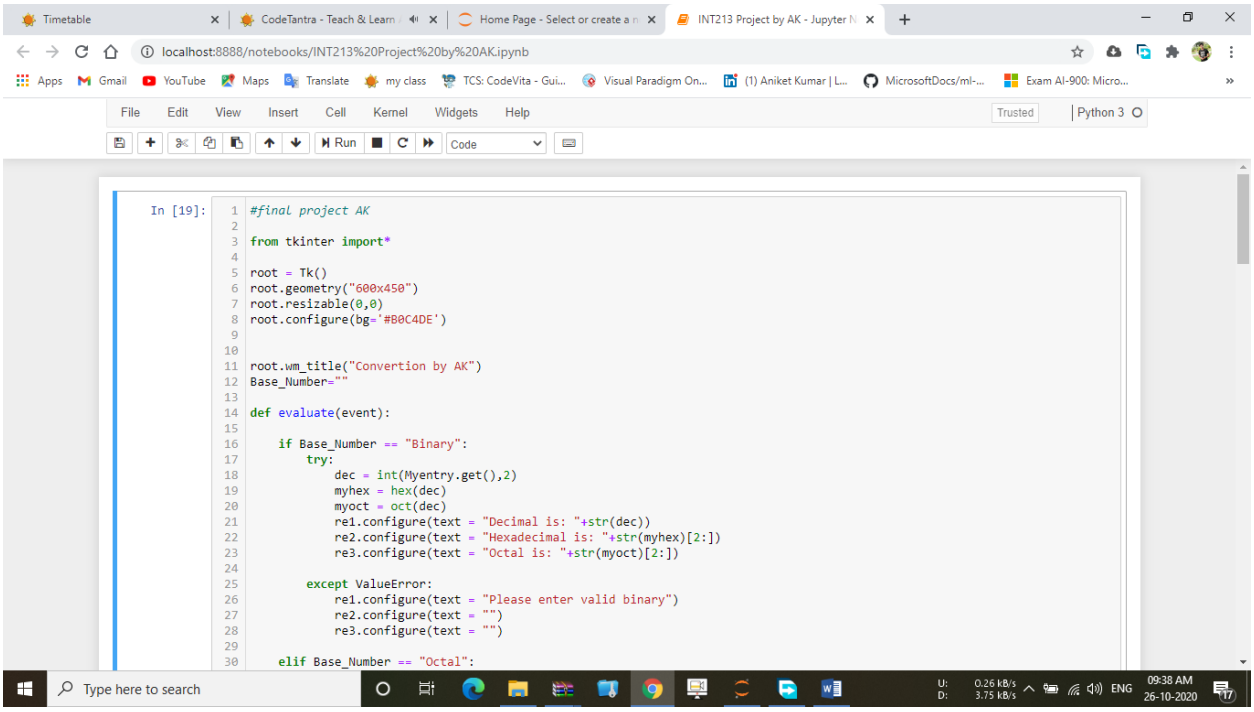
Work Done By Swaraj Senapati (Roll no. 05) :-

- Help in the coding of the labelling of the GUI.
- Given some description in the coding part.
- Written some part of report

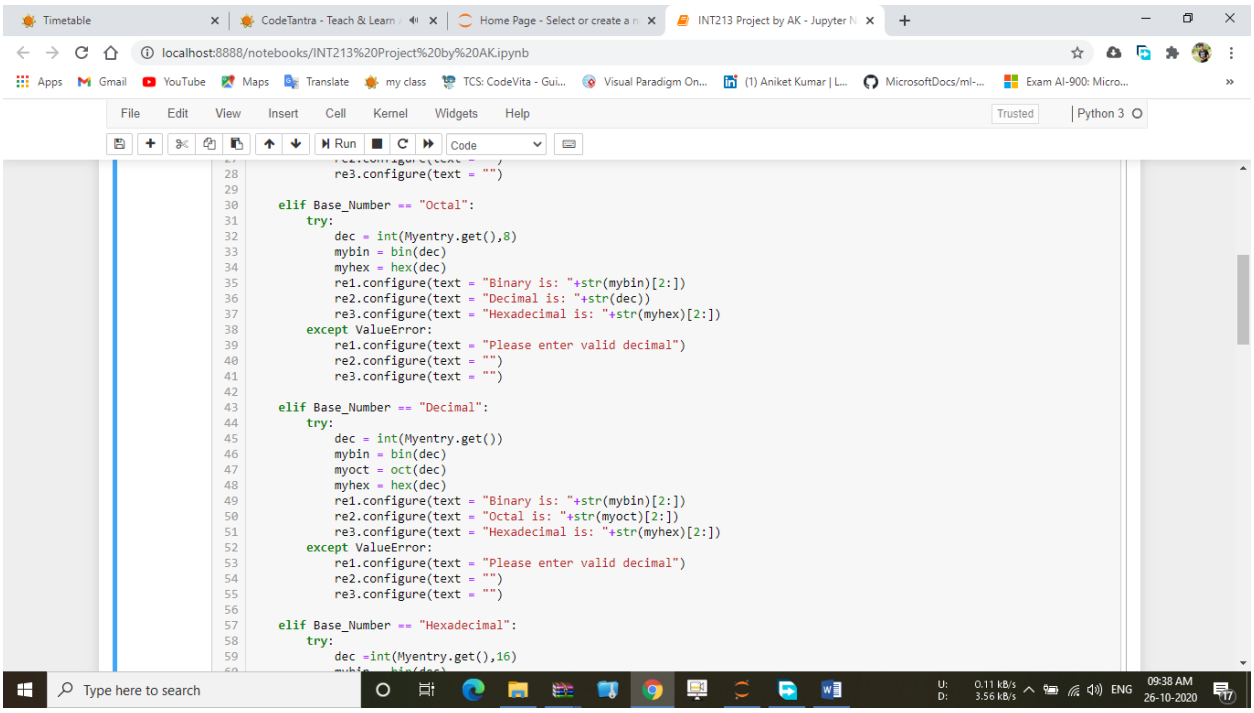
Work Done By Abhinav Yadav (Roll no. 55) :-

- Given some description in the coding part.
- Help in writing some part of report.

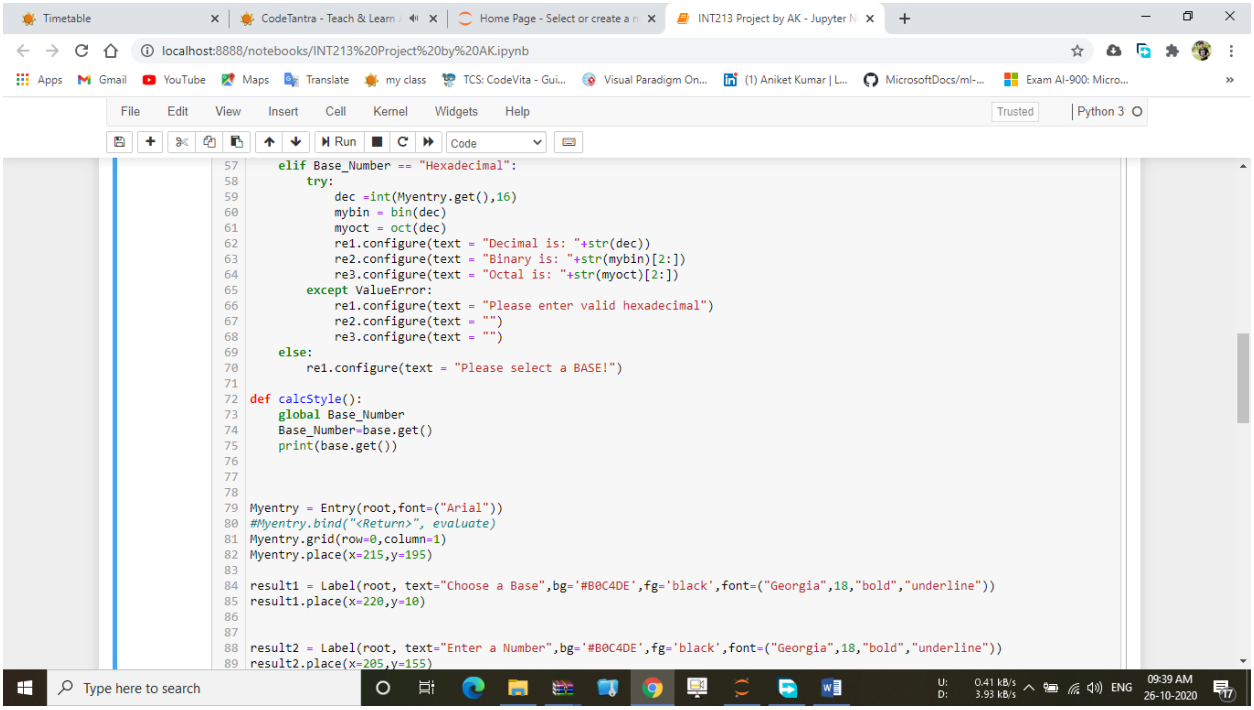
SNAPSHOTS OF THE CODING:-



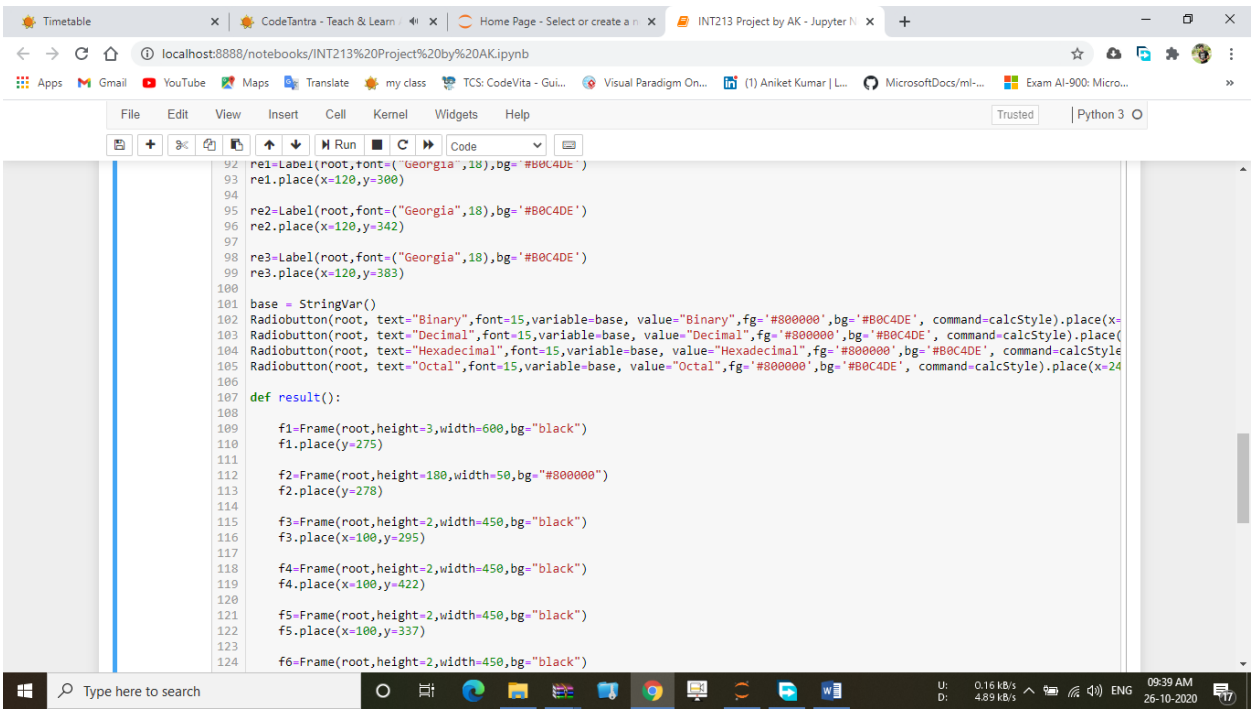
```
In [19]:
#final project AK

from tkinter import*

root = Tk()
root.geometry("600x450")
root.resizable(0,0)
root.configure(bg='#B0C4DE')


root.wm_title("Convertion by AK")
Base_Number=""

def evaluate(event):

    if Base_Number == "Binary":
        try:
            dec = int(Myentry.get(),2)
            myhex = hex(dec)
            myoct = oct(dec)
            re1.configure(text = "Decimal is: "+str(dec))
            re2.configure(text = "Hexadecimal is: "+str(myhex)[2:])
            re3.configure(text = "Octal is: "+str(myoct)[2:])

        except ValueError:
            re1.configure(text = "Please enter valid binary")
            re2.configure(text = "")
            re3.configure(text = "")

    elif Base_Number == "Octal":
```



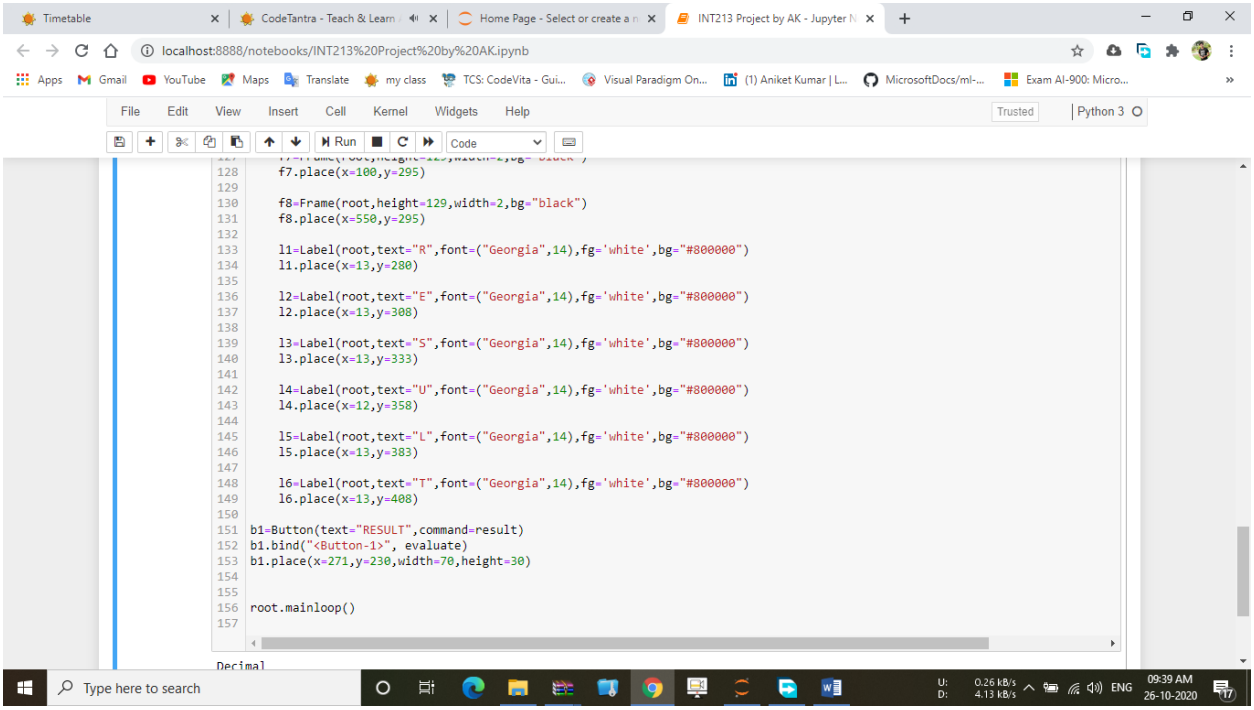
```
            re3.configure(text = "")

    elif Base_Number == "Octal":
        try:
            dec = int(Myentry.get(),8)
            mybin = bin(dec)
            myhex = hex(dec)
            re1.configure(text = "Binary is: "+str(mybin)[2:])
            re2.configure(text = "Decimal is: "+str(dec))
            re3.configure(text = "Hexadecimal is: "+str(myhex)[2:])
        except ValueError:
            re1.configure(text = "Please enter valid decimal")
            re2.configure(text = "")
            re3.configure(text = "")

    elif Base_Number == "Decimal":
        try:
            dec = int(Myentry.get())
            mybin = bin(dec)
            myoct = oct(dec)
            myhex = hex(dec)
            re1.configure(text = "Binary is: "+str(mybin)[2:])
            re2.configure(text = "Octal is: "+str(myoct)[2:])
            re3.configure(text = "Hexadecimal is: "+str(myhex)[2:])
        except ValueError:
            re1.configure(text = "Please enter valid decimal")
            re2.configure(text = "")
            re3.configure(text = "")

    elif Base_Number == "Hexadecimal":
        try:
            dec =int(Myentry.get(),16)
```



```
    elif Base_Number == "Hexadecimal":
        try:
            dec =int(Myentry.get(),16)
            mybin = bin(dec)
            myoct = oct(dec)
            re1.configure(text = "Decimal is: "+str(dec))
            re2.configure(text = "Binary is: "+str(mybin)[2:])
            re3.configure(text = "Octal is: "+str(myoct)[2:])
        except ValueError:
            re1.configure(text = "Please enter valid hexadecimal")
            re2.configure(text = "")
            re3.configure(text = "")
    else:
        re1.configure(text = "Please select a BASE!")

def calcStyle():
    global Base_Number
    Base_Number=base.get()
    print(base.get())



Myentry = Entry(root,font=("Arial"))
#Myentry.bind("<Return>", evaluate)
Myentry.grid(row=0,column=1)
Myentry.place(x=215,y=195)

result1 = Label(root, text="Choose a Base",bg='#B0C4DE',fg='black',font=("Georgia",18,"bold","underline"))
result1.place(x=220,y=10)


result2 = Label(root, text="Enter a Number",bg='#B0C4DE',fg='black',font=("Georgia",18,"bold","underline"))
result2.place(x=205,y=155)
```

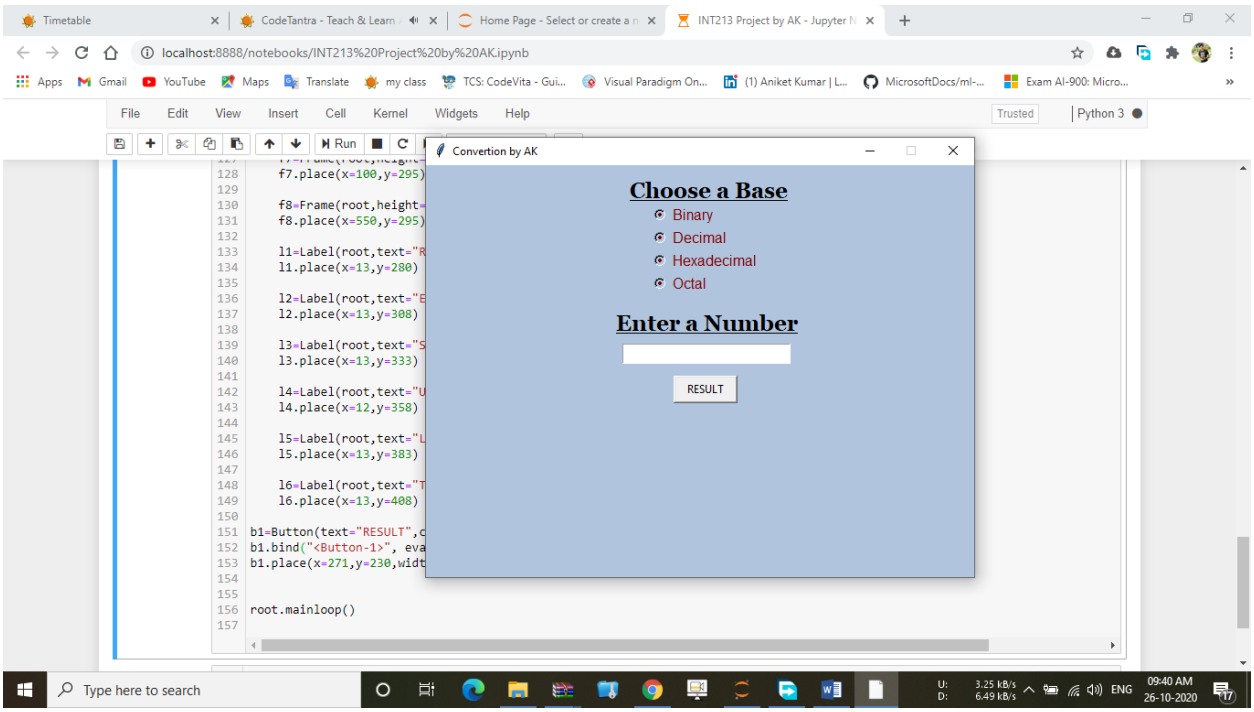


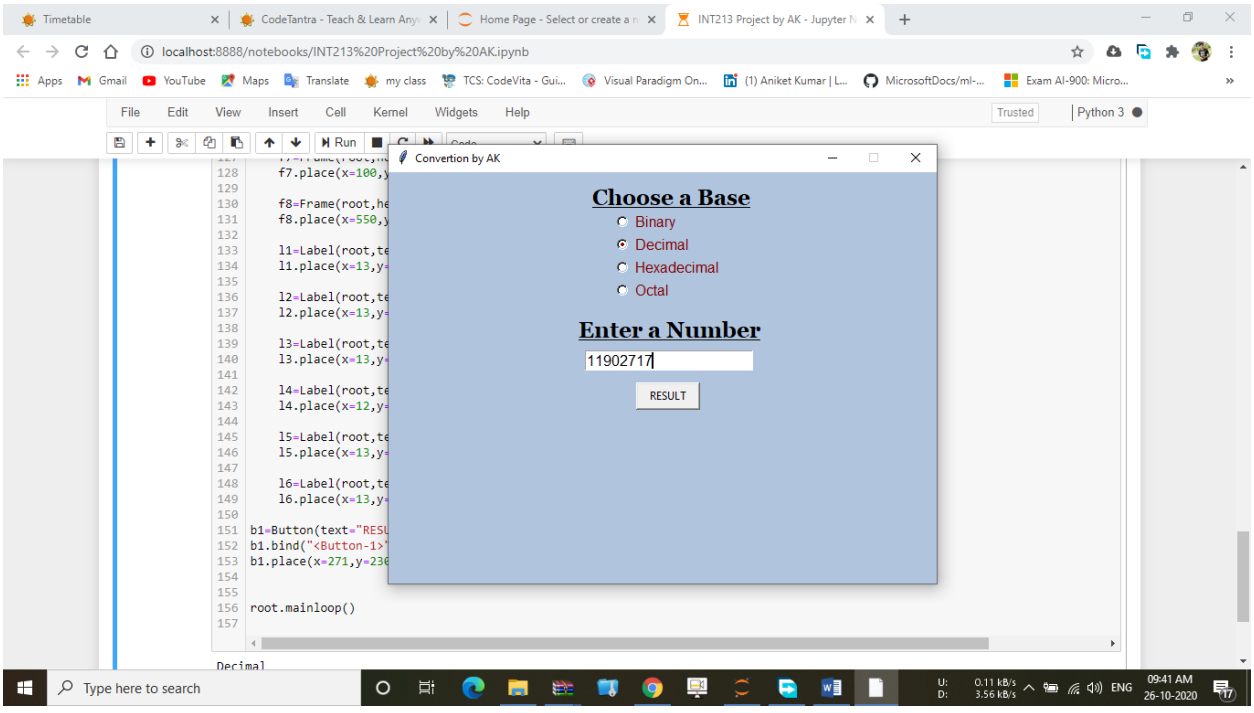
```
re1=Label(root,font=("Georgia",18),bg='#B0C4DE')
re1.place(x=120,y=300)

re2=Label(root,font=("Georgia",18),bg='#B0C4DE')
re2.place(x=120,y=342)

re3=Label(root,font=("Georgia",18),bg='#B0C4DE')
re3.place(x=120,y=383)

base = StringVar()
Radiobutton(root, text="Binary",font=15,variable=base, value="Binary",fg='#800000',bg='#B0C4DE', command=calcStyle).place(x=
Radiobutton(root, text="Decimal",font=15,variable=base, value="Decimal",fg='#800000',bg='#B0C4DE', command=calcStyle).place(
Radiobutton(root, text="Hexadecimal",font=15,variable=base, value="Hexadecimal",fg='#800000',bg='#B0C4DE', command=calcStyle
Radiobutton(root, text="Octal",font=15,variable=base, value="Octal",fg='#800000',bg='#B0C4DE', command=calcStyle).place(x=24

def result():

    f1=Frame(root,height=3,width=600,bg="black")
    f1.place(y=275)

    f2=Frame(root,height=180,width=50,bg="#800000")
    f2.place(y=278)

    f3=Frame(root,height=2,width=450,bg="black")
    f3.place(x=100,y=295)

    f4=Frame(root,height=2,width=450,bg="black")
    f4.place(x=100,y=422)

    f5=Frame(root,height=2,width=450,bg="black")
    f5.place(x=100,y=337)

    f6=Frame(root,height=2,width=450,bg="black")
```

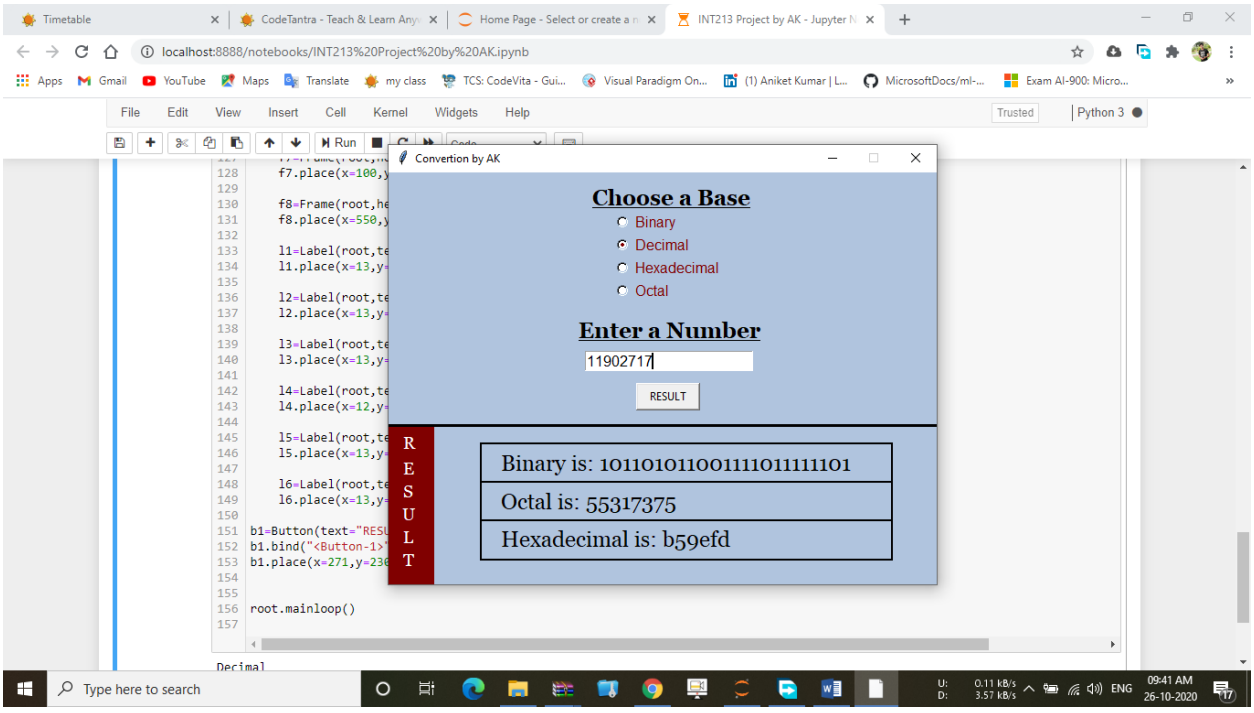


```
    f7.place(x=100,y=295)

    f8=Frame(root,height=129,width=2,bg="black")
    f8.place(x=550,y=295)

    l1=Label(root,text="R",font=("Georgia",14),fg='white',bg="#800000")
    l1.place(x=13,y=280)

    l2=Label(root,text="E",font=("Georgia",14),fg='white',bg="#800000")
    l2.place(x=13,y=308)

    l3=Label(root,text="S",font=("Georgia",14),fg='white',bg="#800000")
    l3.place(x=13,y=333)

    l4=Label(root,text="U",font=("Georgia",14),fg='white',bg="#800000")
    l4.place(x=12,y=358)

    l5=Label(root,text="L",font=("Georgia",14),fg='white',bg="#800000")
    l5.place(x=13,y=383)

    l6=Label(root,text="T",font=("Georgia",14),fg='white',bg="#800000")
    l6.place(x=13,y=408)

b1=Button(text="RESULT",command=result)
b1.bind("<Button-1>", evaluate)
b1.place(x=271,y=230,width=70,height=30)


root.mainloop()
```

SNAPSHOTS OF THE CODING RESULT:-




Convertion by AK
Choose a Base
Binary
Decimal
Hexadecimal
Octal
Enter a Number
11902717
RESULT
root.mainloop()


Convertion by AK
Choose a Base
Binary
Decimal
Hexadecimal
Octal
Enter a Number
11902717
RESULT
RESULT
Binary is: 10110101100111011111101
Octal is: 55317375
Hexadecimal is: b59efd
root.mainloop()