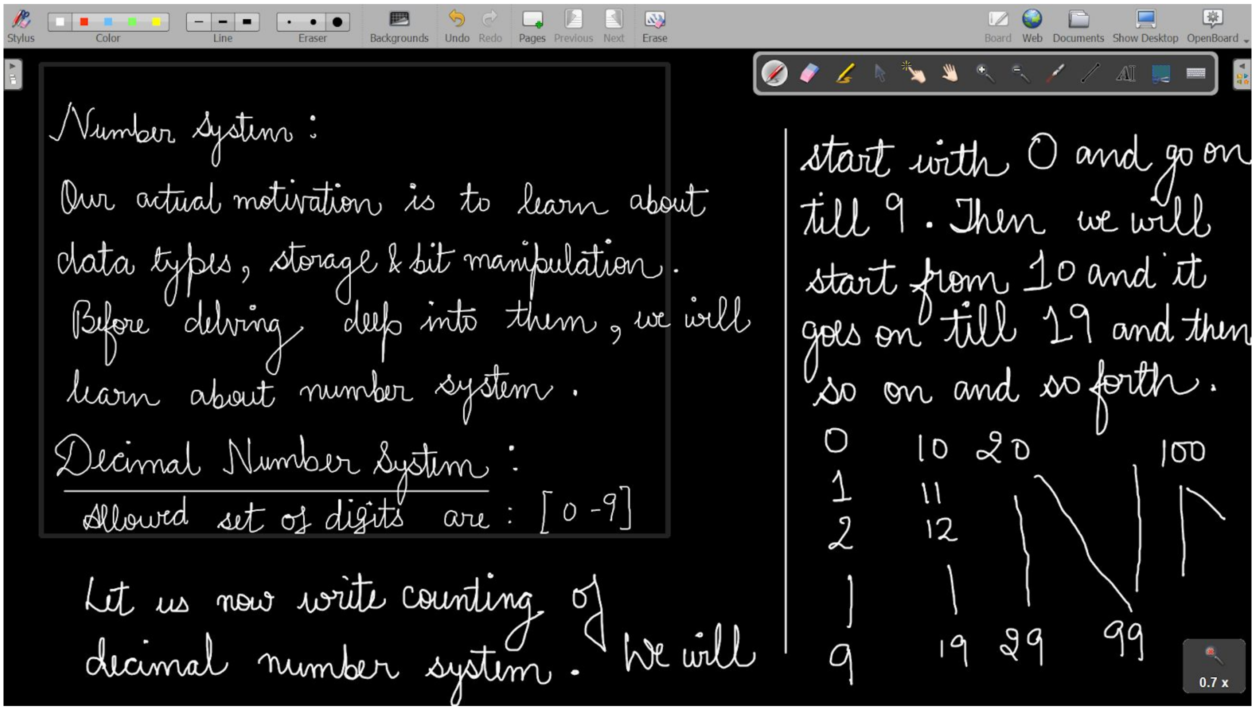
Decimal Number System



Number System :

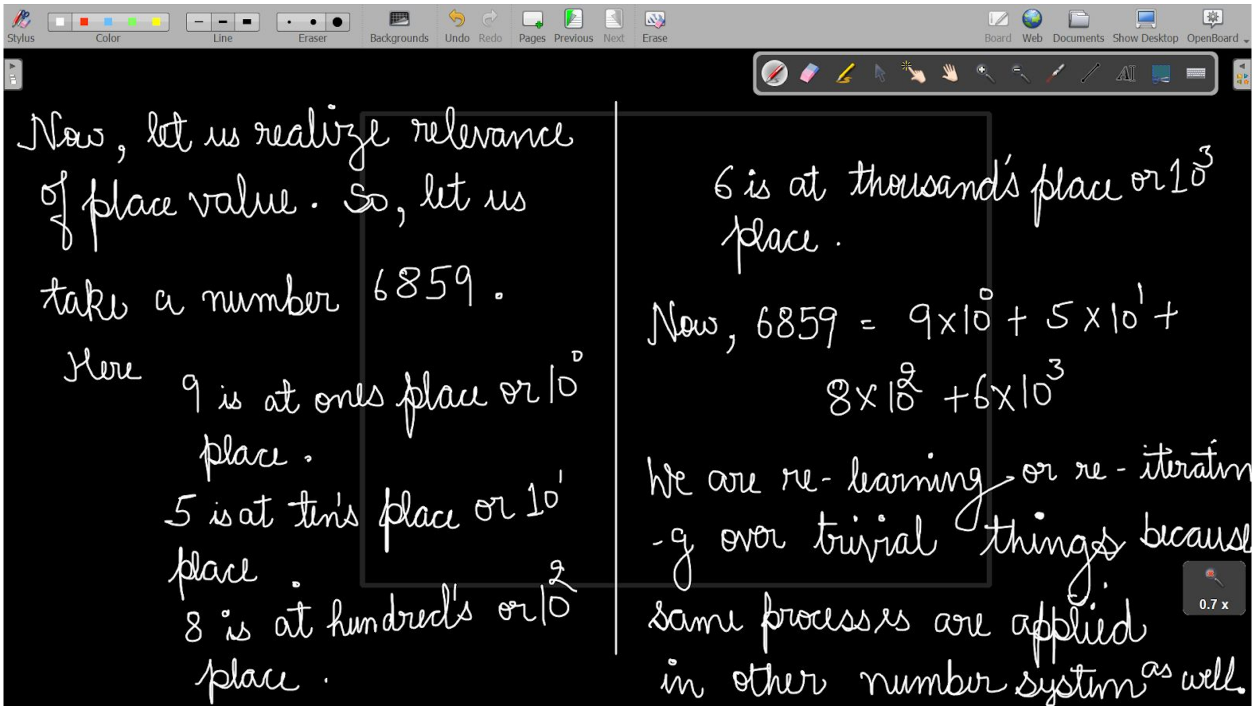
Our actual motivation is to learn about data types, storage & bit manipulation. Before delving deep into them, we will learn about number system.

Decimal Number System :

Allowed set of digits are : [0-9]

Let us now write counting of decimal number system. We will start with 0 and go on till 9. Then we will start from 10 and it goes on till 19 and then so on and so forth.

0 10 20 100
1 11
2 12
9 19 29 99



Now, let us realize relevance of place value. So, let us take a number 6859.

Here 9 is at ones place or 10^0 place.

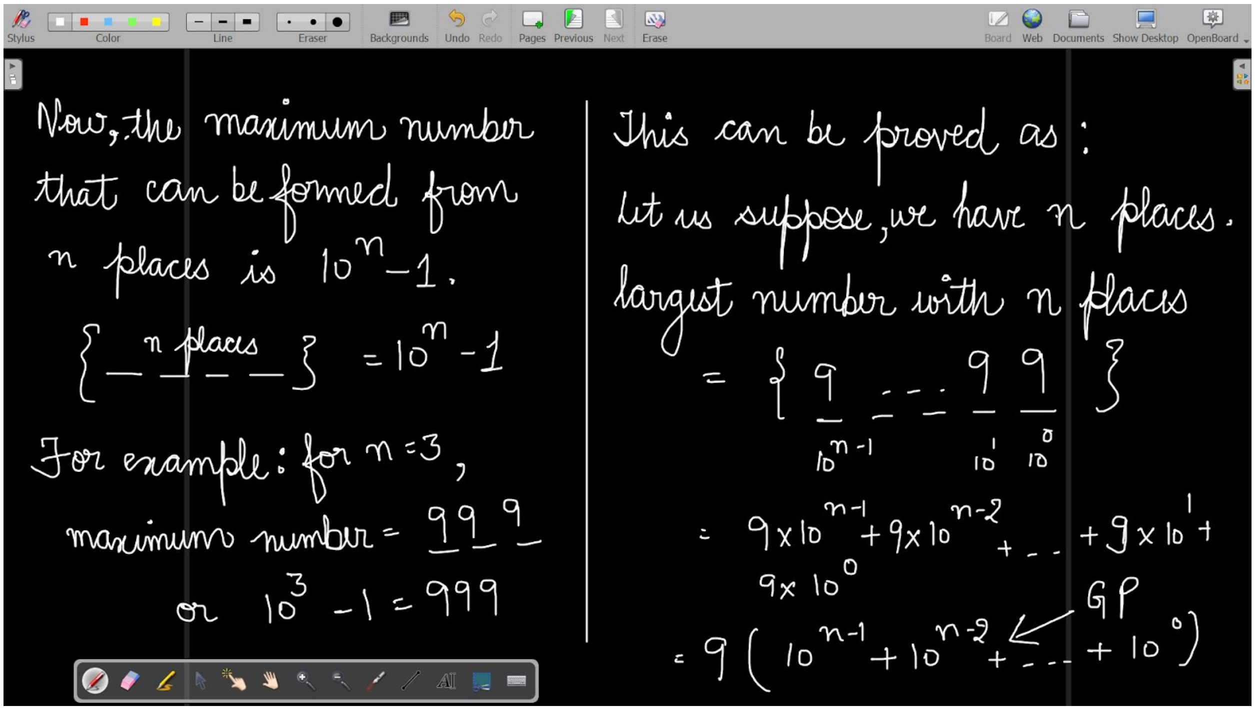
5 is at ten's place or 10^1 place.

8 is at hundred's or 10^2 place.

6 is at thousand's place or 10^3 place.

Now, $6859 = 9\times10^0 + 5\times10^1 + 8\times10^2 + 6\times10^3$

We are re-learning or re-iterating over trivial things because same processes are applied in other number system as well.



Now, the maximum number that can be formed from n places is $10^n - 1$.

$$\{\overset{n\ places}{\_\ \_\ \_\ \_}\} = 10^n - 1$$

For example: for $n = 3$,

maximum number $= \underline{9}\,\underline{9}\,\underline{9}$

or $10^3 - 1 = 999$

This can be proved as:

Let us suppose, we have n places.

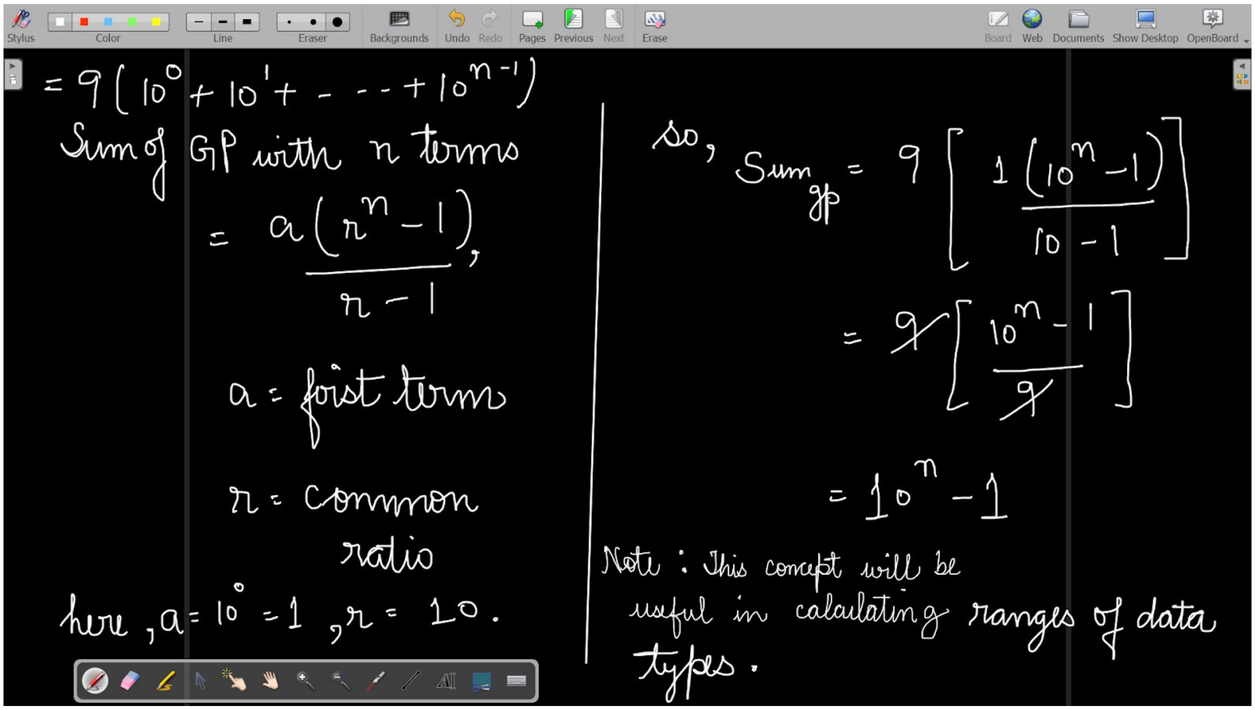
largest number with n places

$$= \{ \underset{10^{n-1}}{\underline{9}}\ \_\ \_\ \_\ \underset{10^1}{\underline{9}}\ \underset{10^0}{\underline{9}} \}$$

$$= 9\times 10^{n-1} + 9\times 10^{n-2} + \dots + 9\times 10^1 + 9\times 10^0$$

GP

$$= 9(10^{n-1} + 10^{n-2} + \dots + 10^0)$$



$$= 9(10^0 + 10^1 + \dots + 10^{n-1})$$

Sum of GP with n terms

$$= \frac{a(r^n - 1)}{r - 1},$$

a = first term

r = common ratio

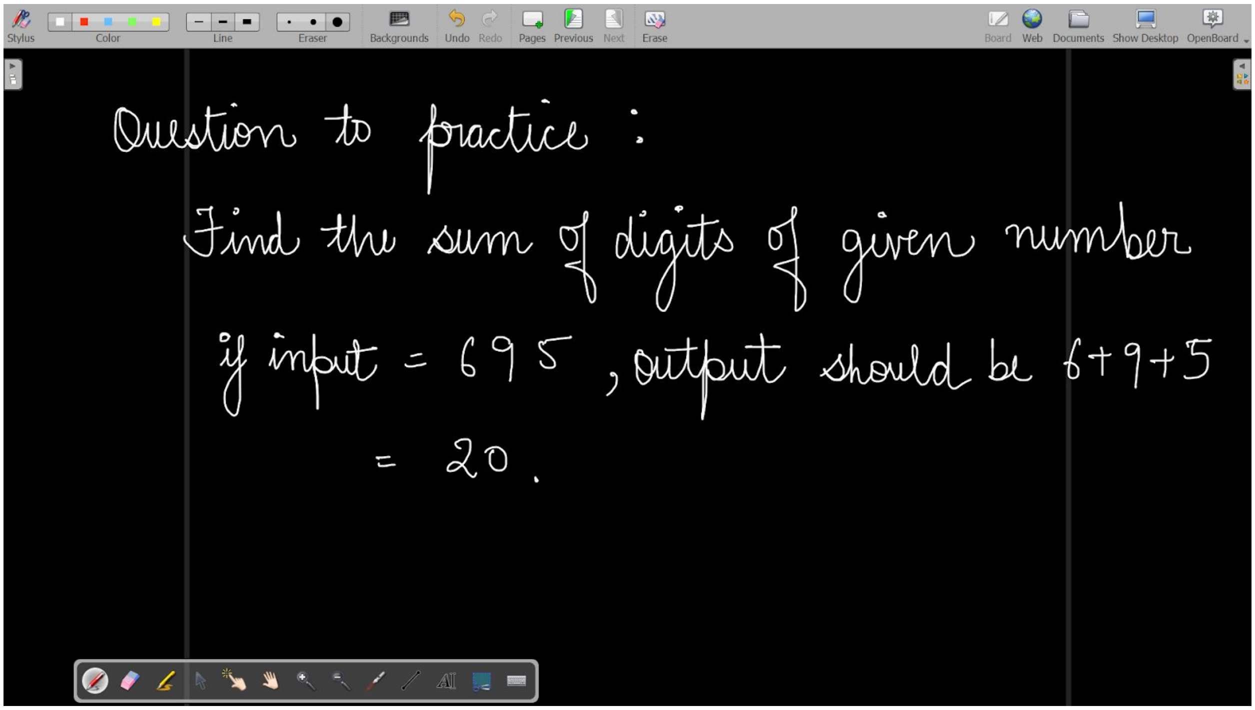
here, $a = 10^0 = 1$, $r = 10$.

So, $$Sum_{gp} = 9\left[\frac{1(10^n - 1)}{10 - 1}\right]$$

$$= \cancel{9}\left[\frac{10^n - 1}{\cancel{9}}\right]$$

$$= 10^n - 1$$

Note: This concept will be useful in calculating ranges of data types.



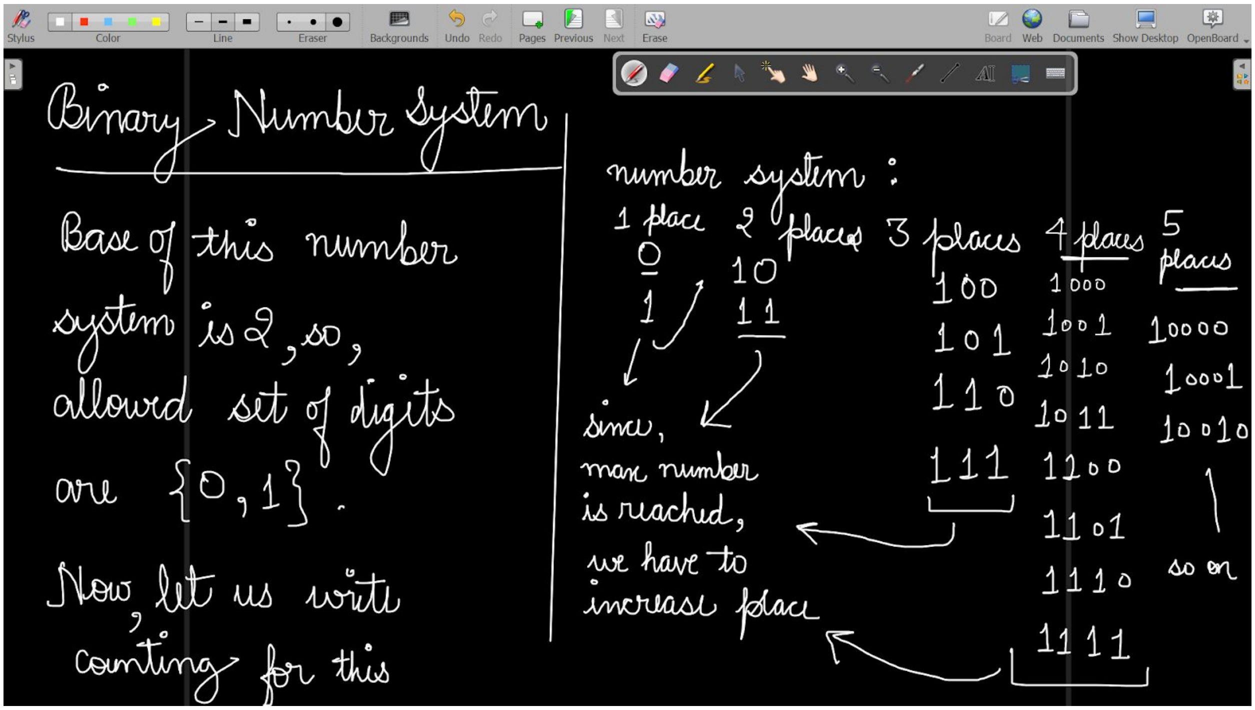
Question to practice :

Find the sum of digits of given number

if input = 695 , output should be 6+9+5

= 20.

Binary Number System



Binary Number System

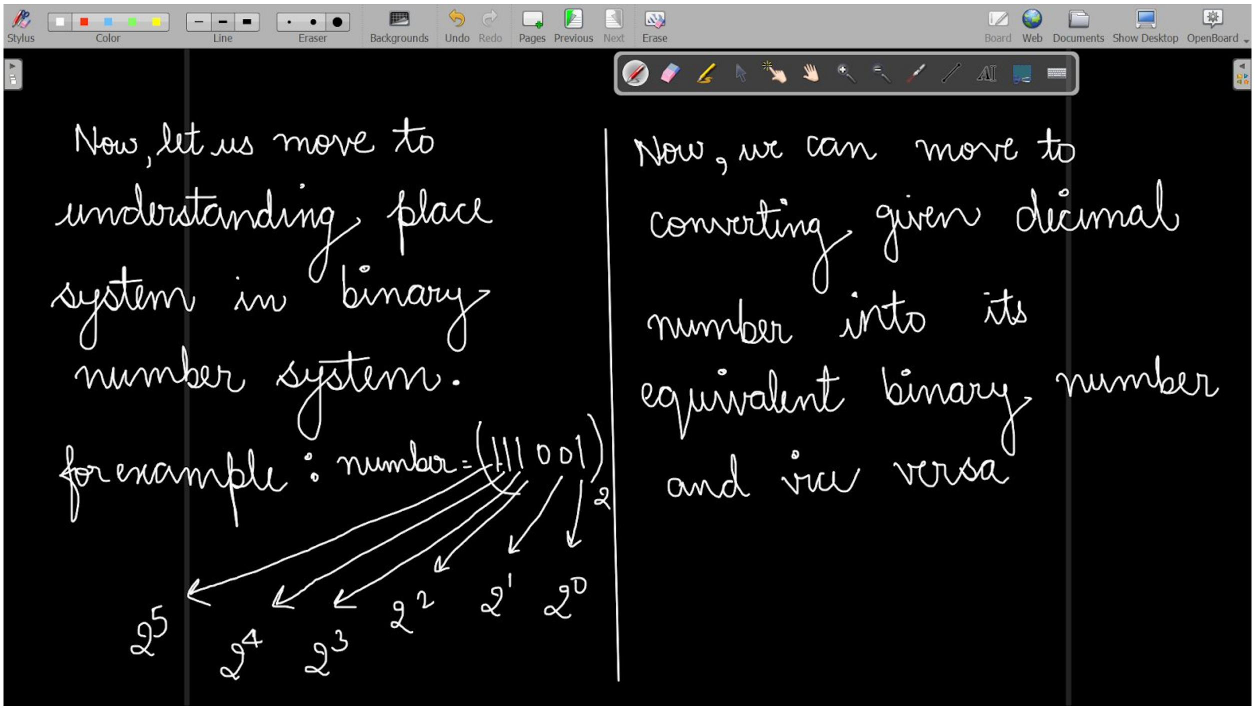
Base of this number system is 2, so, allowed set of digits are {0, 1}.

Now, let us write counting for this

number system :

1 place	2 places	3 places	4 places	5 places
0	10	100	1000	10000
1	11	101	1001	10001
		110	1010	10010
		111	1011	so on
			1100	
			1101	
			1110	
			1111	

since, max number is reached, we have to increase place

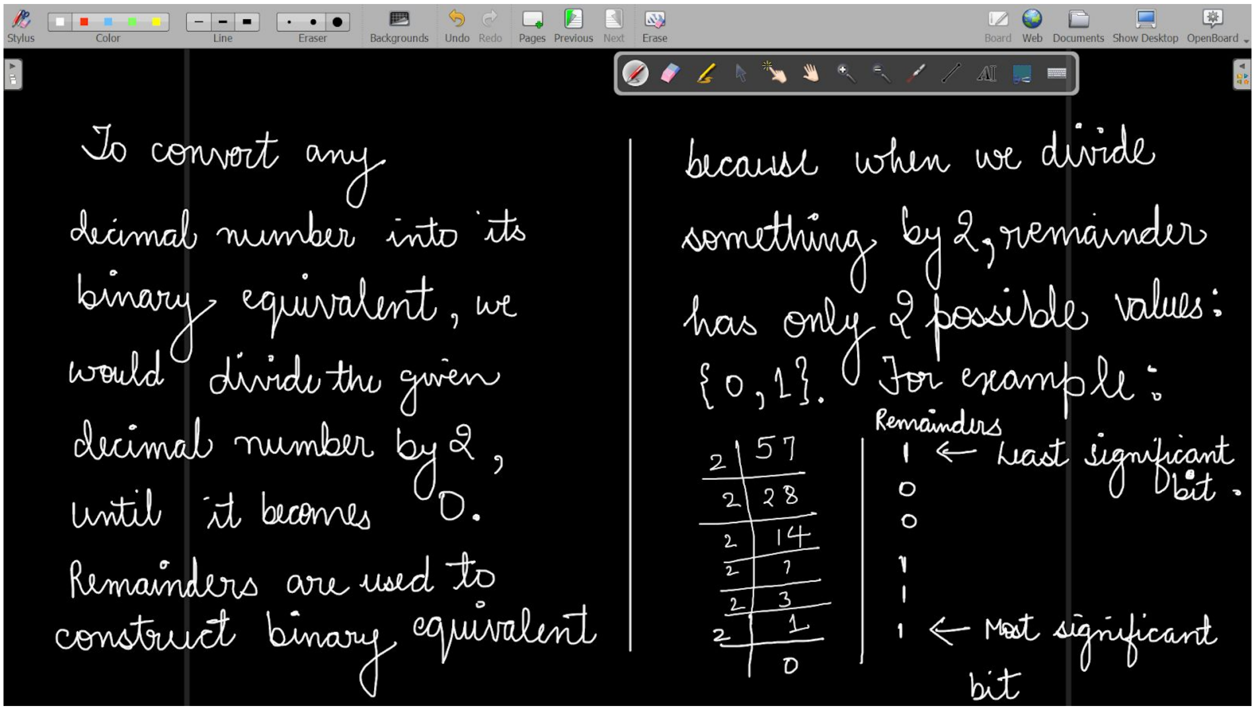


Now, let us move to understanding place system in binary number system.

for example : number = $(111001)_2$

2^5 2^4 2^3 2^2 2^1 2^0

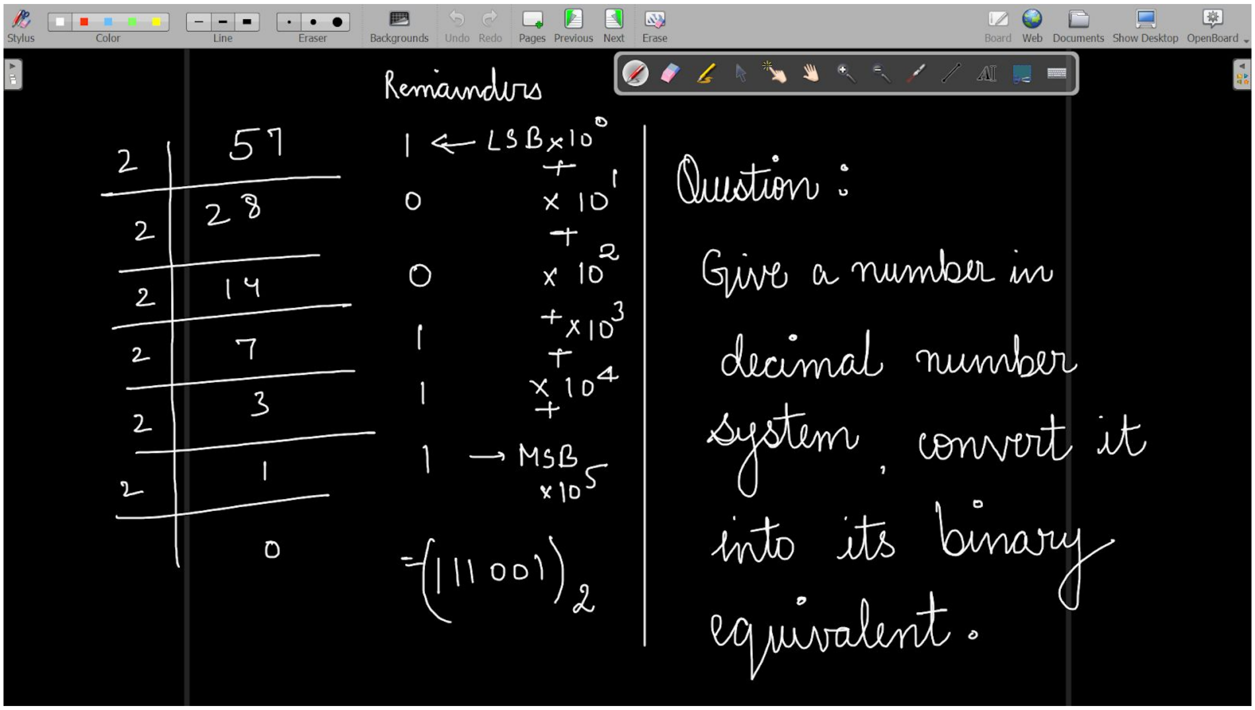
Now, we can move to converting given decimal number into its equivalent binary number and vice versa



To convert any decimal number into its binary equivalent, we would divide the given decimal number by 2, until it becomes 0. Remainders are used to construct binary equivalent

because when we divide something by 2, remainder has only 2 possible values: {0, 1}. For example:

		Remainders
2	57	1 ← Least significant bit.
2	28	0
2	14	0
2	7	1
2	3	1
2	1	1 ← Most significant bit
	0	

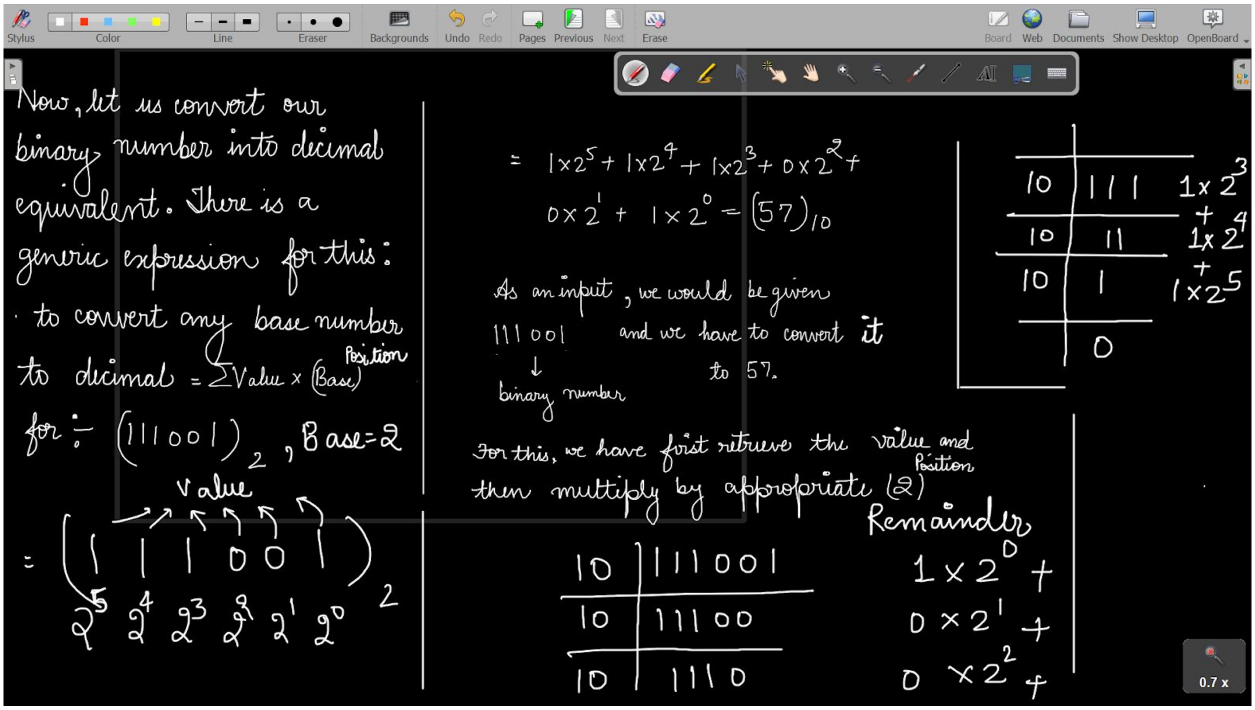


		Remainders
2	57	1 ← LSB $\times 10^0$ +
2	28	0 $\times 10^1$ +
2	14	0 $\times 10^2$ +
2	7	1 $\times 10^3$ +
2	3	1 $\times 10^4$ +
2	1	1 → MSB $\times 10^5$
	0	

$= (111001)_2$

Question:

Give a number in decimal number system, convert it into its binary equivalent.



Now, let us convert our binary number into decimal equivalent. There is a generic expression for this:

· to convert any base number to decimal $= \sum \text{Value} \times (\text{Base})^{\text{Position}}$

for :- $(111001)_2$, Base = 2

Value

$$= \left(\underset{2^5}{1}\ \underset{2^4}{1}\ \underset{2^3}{1}\ \underset{2^2}{0}\ \underset{2^1}{0}\ \underset{2^0}{1}\right)_2$$

$$= 1\times2^5 + 1\times2^4 + 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0 = (57)_{10}$$

As an input, we would be given 111001 and we have to convert it to 57.

↓

binary number

For this, we have first retrieve the value and then multiply by appropriate $(2)^{\text{Position}}$

$$\begin{array}{r|l} 10 & 111001 \\ \hline 10 & 11100 \\ \hline 10 & 1110 \\ \hline 10 & 111 \\ \hline 10 & 11 \\ \hline 10 & 1 \\ \hline & 0 \end{array}$$

Remainder

$1\times2^0 +$

$0\times2^1 +$

$0\times2^2 +$

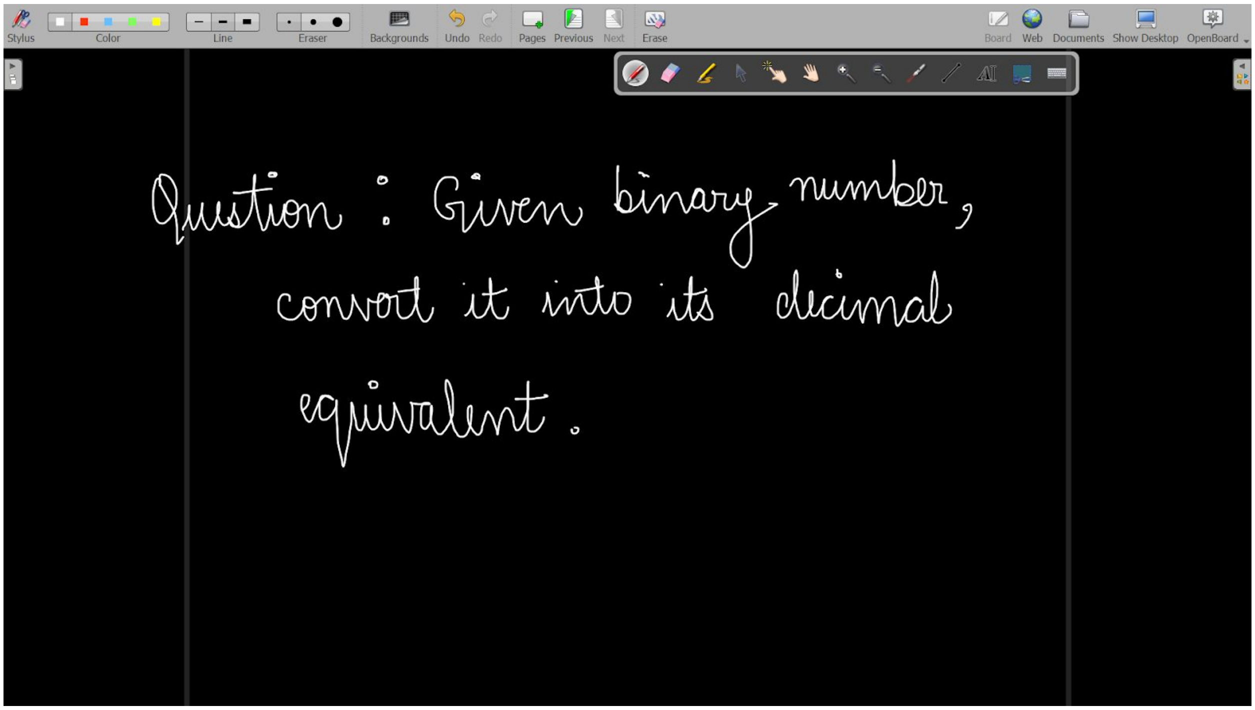
1×2^3

$+$

1×2^4

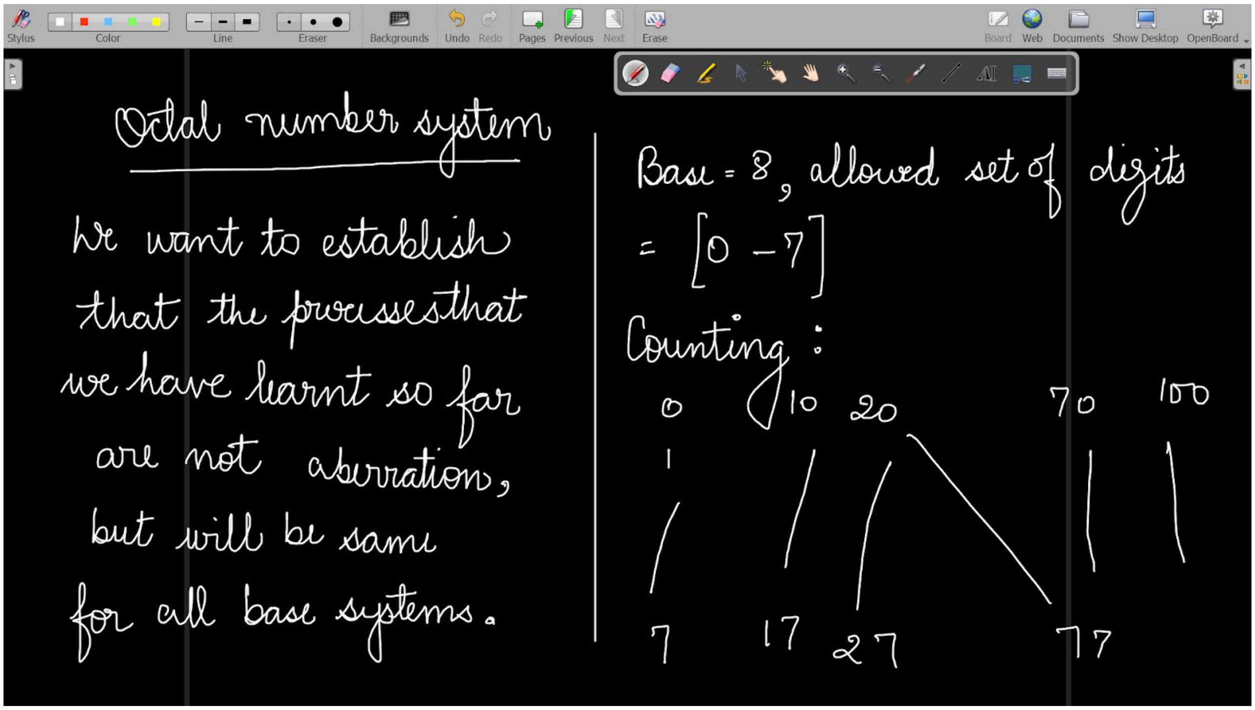
$+$

1×2^5



Question : Given binary number, convert it into its decimal equivalent.

Octal Number System



Octal number system

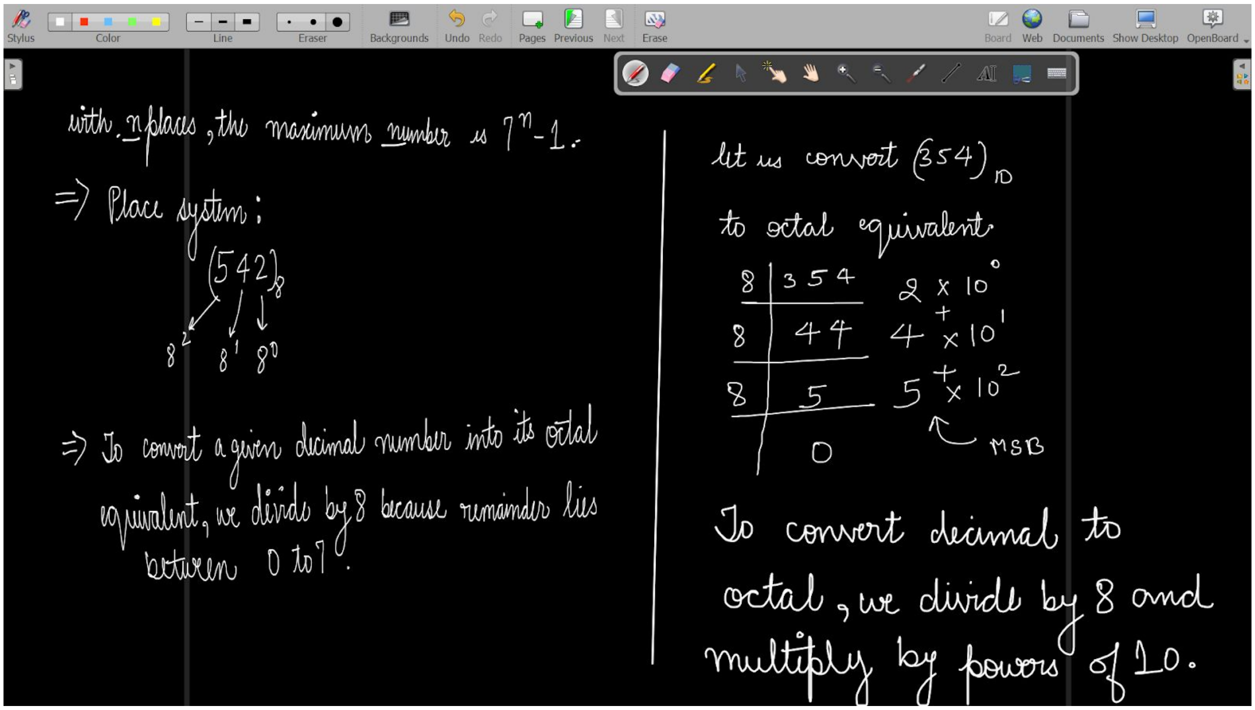
We want to establish that the processesthat we have learnt so far are not aberration, but will be same for all base systems.

Base = 8, allowed set of digits = [0 – 7]

Counting :

0   10   20   70   100

7   17   27   77



with n places, the maximum number is $7^n - 1$.

⇒ Place system :

$(542)_8$

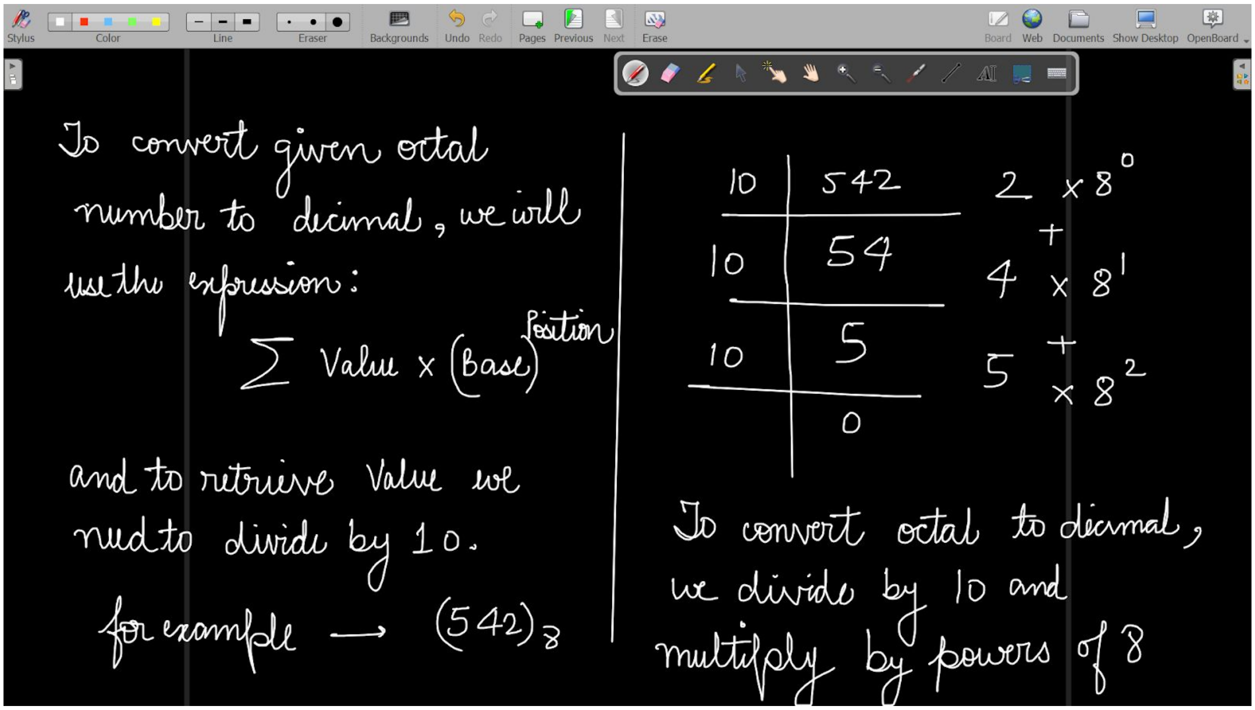
8^2 8^1 8^0

⇒ To convert a given decimal number into its octal equivalent, we divide by 8 because remainder lies between 0 to 7.

let us convert $(354)_{10}$ to octal equivalent

8	354	2×10^0
8	44	$+ \; 4 \times 10^1$
8	5	$+ \; 5 \times 10^2$
	0	MSB

To convert decimal to octal, we divide by 8 and multiply by powers of 10.



To convert given octal number to decimal, we will use the expression:

$$\sum \text{Value} \times (\text{Base})^{\text{Position}}$$

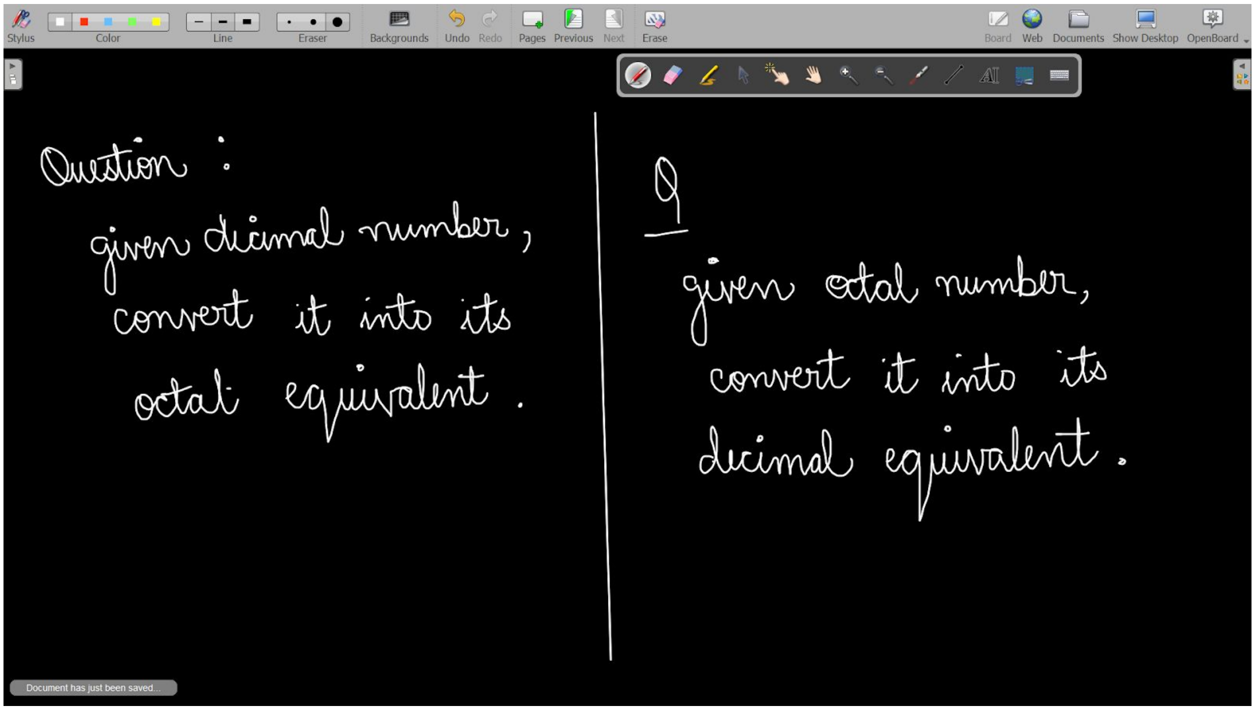
and to retrieve Value we need to divide by 10.

for example → $(542)_8$

10	542
10	54
10	5
	0

$$2 \times 8^0 + 4 \times 8^1 + 5 \times 8^2$$

To convert octal to decimal, we divide by 10 and multiply by powers of 8



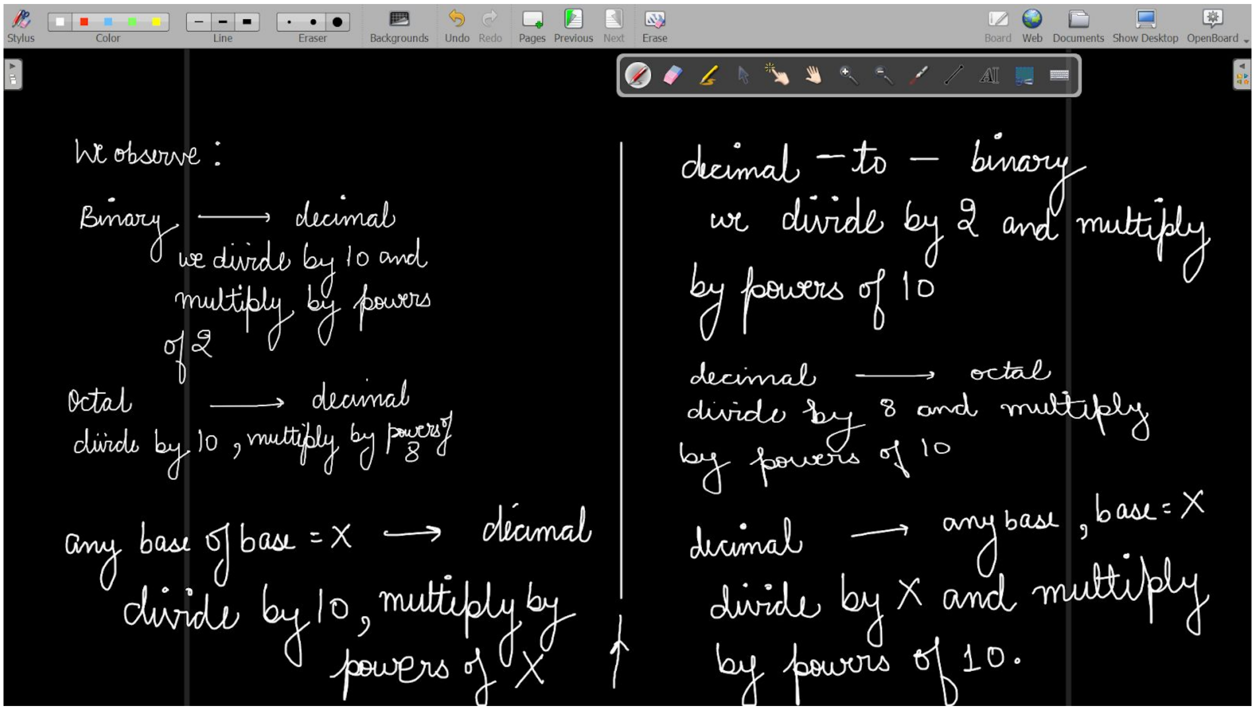
Question :

given decimal number, convert it into its octal equivalent.

Q

given octal number, convert it into its decimal equivalent.

Document has just been saved...



We observe :

Binary ⟶ decimal
we divide by 10 and multiply by powers of 2

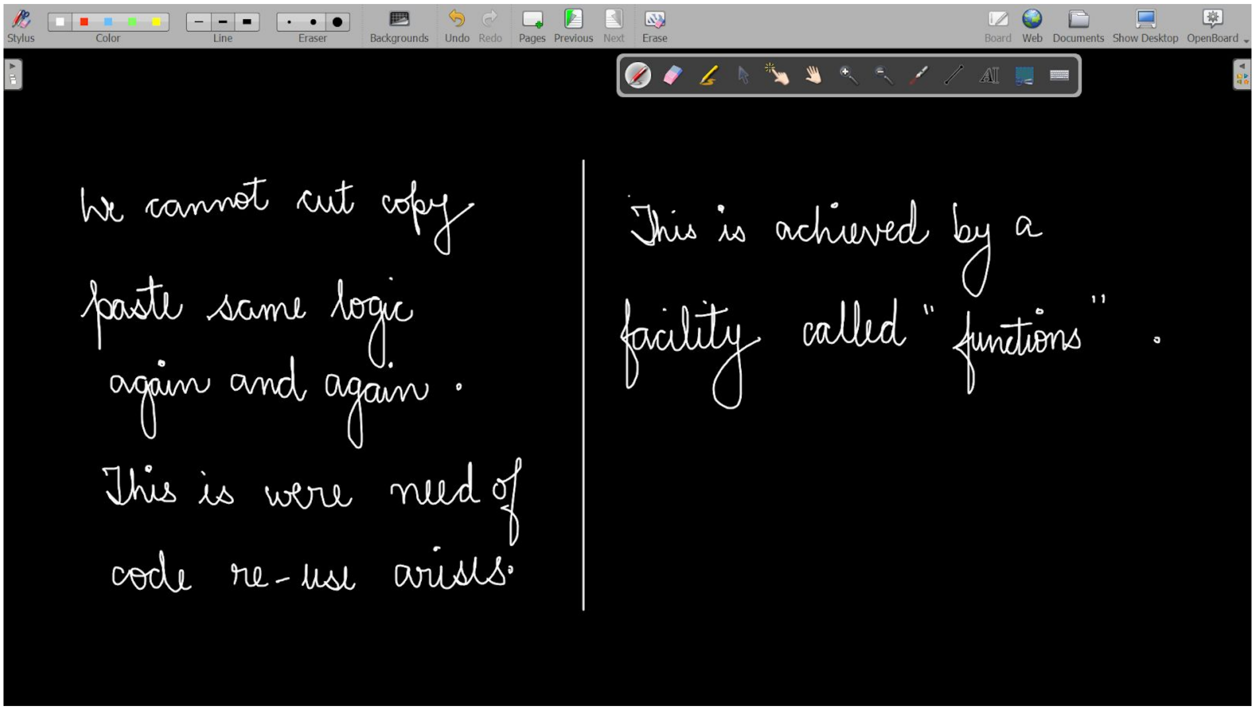
Octal ⟶ decimal
divide by 10 , multiply by powers of 8

any base of base = X ⟶ decimal
divide by 10, multiply by powers of X

decimal – to – binary
we divide by 2 and multiply by powers of 10

decimal ⟶ octal
divide by 8 and multiply by powers of 10

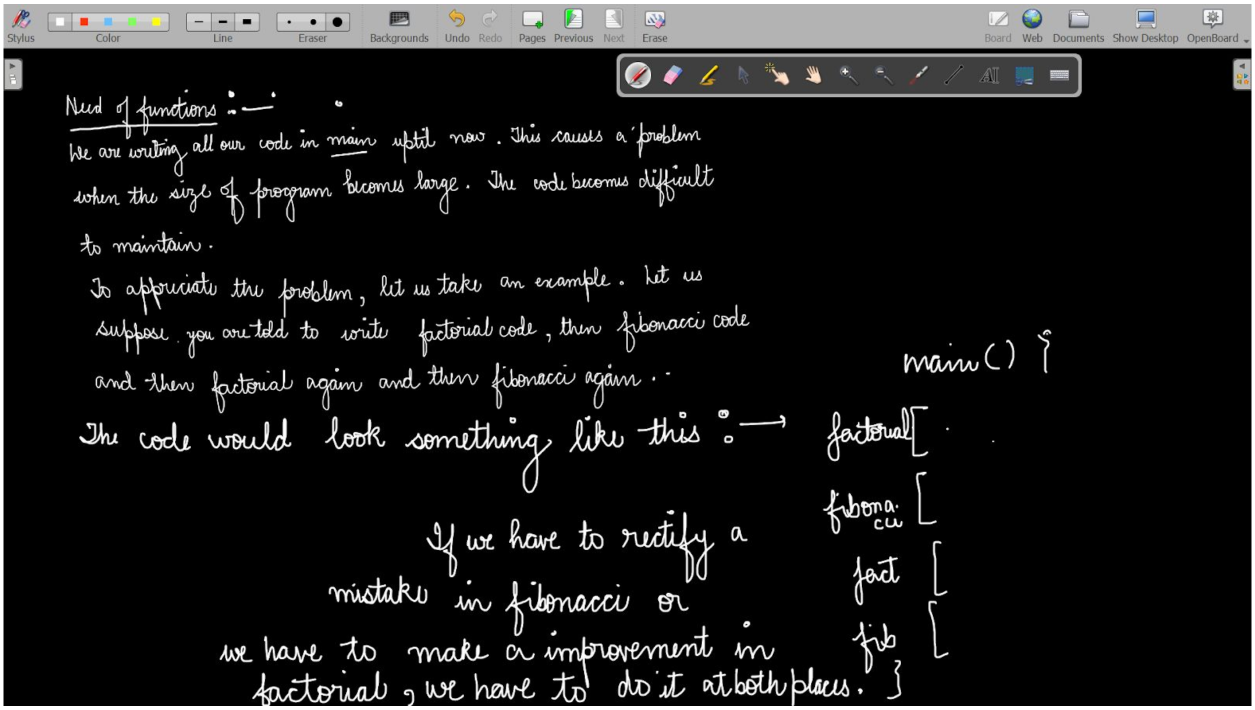
decimal ⟶ any base , base = X
divide by X and multiply by powers of 10.



We cannot cut copy paste same logic again and again. This is were need of code re-use arises.

This is achieved by a facility called " functions" .

Functions



Need of functions :—

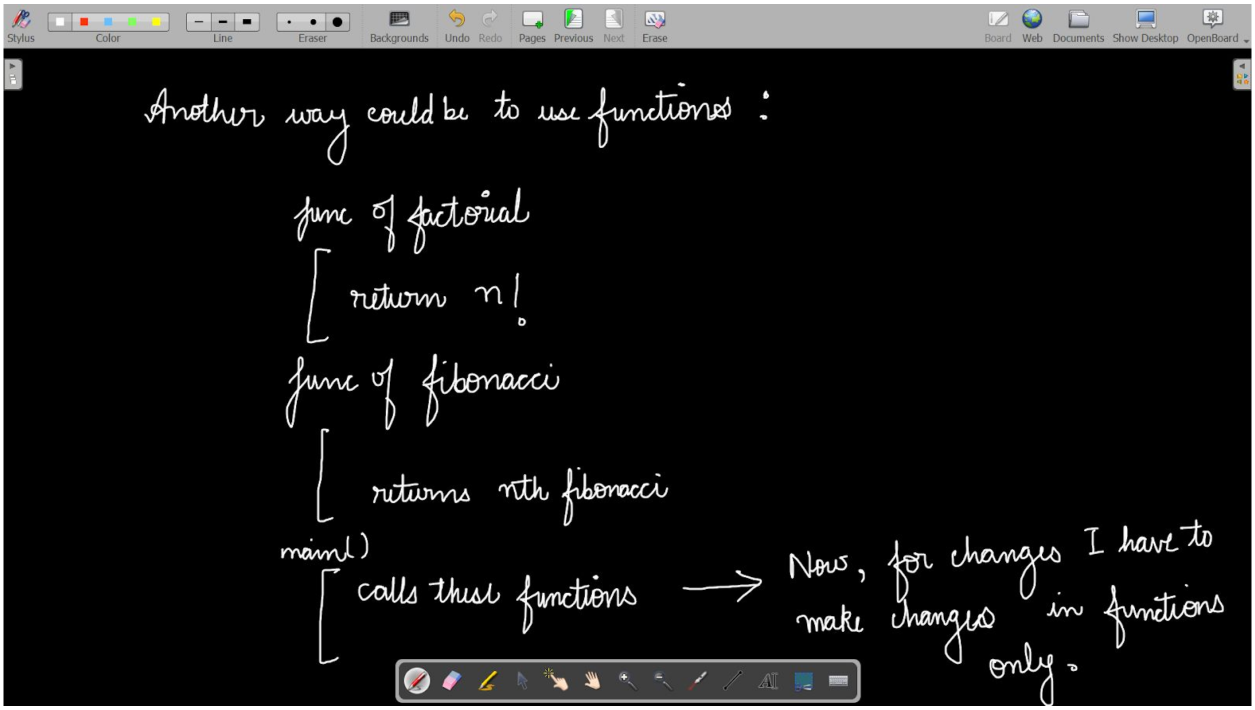
We are writing all our code in main uptil now. This causes a problem when the size of program becomes large. The code becomes difficult to maintain.

To appreciate the problem, let us take an example. Let us suppose you are told to write factorial code, then fibonacci code and then factorial again and then fibonacci again..

The code would look something like this :—

```
main() {
  factorial[ . .
  fibonacci [
  fact [
  fib [
}
```

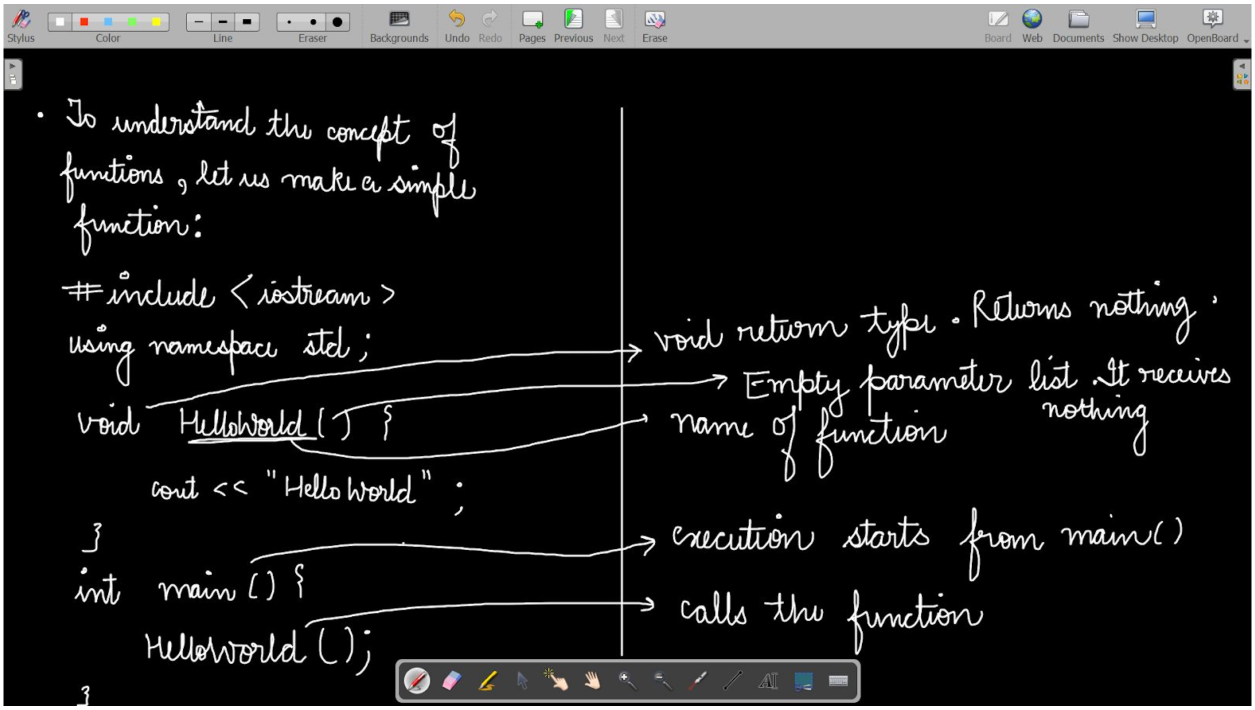
If we have to rectify a mistake in fibonacci or we have to make a improvement in factorial, we have to do it at both places.



Another way could be to use functions :

```
func of factorial
  [ return n!
func of fibonacci
  [ returns nth fibonacci
main()
  [ calls these functions
```

→ Now, for changes I have to make changes in functions only.



- To understand the concept of funtions, let us make a simple function:

```
#include <iostream>
using namespace std;
void HelloWorld ( ) {
        cout << "Hello World" ;
 }
int main () {
      HelloWorld ();
 }
```

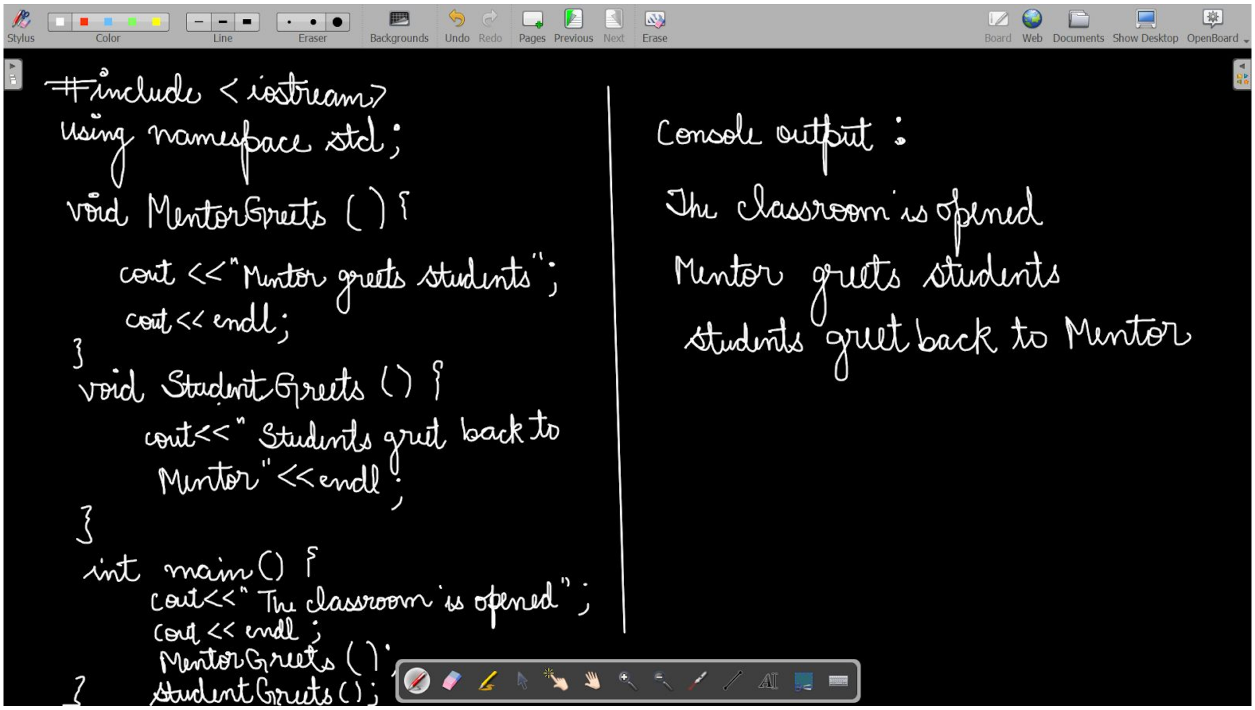
void return type. Returns nothing.

Empty parameter list. It receives nothing

name of function

execution starts from main()

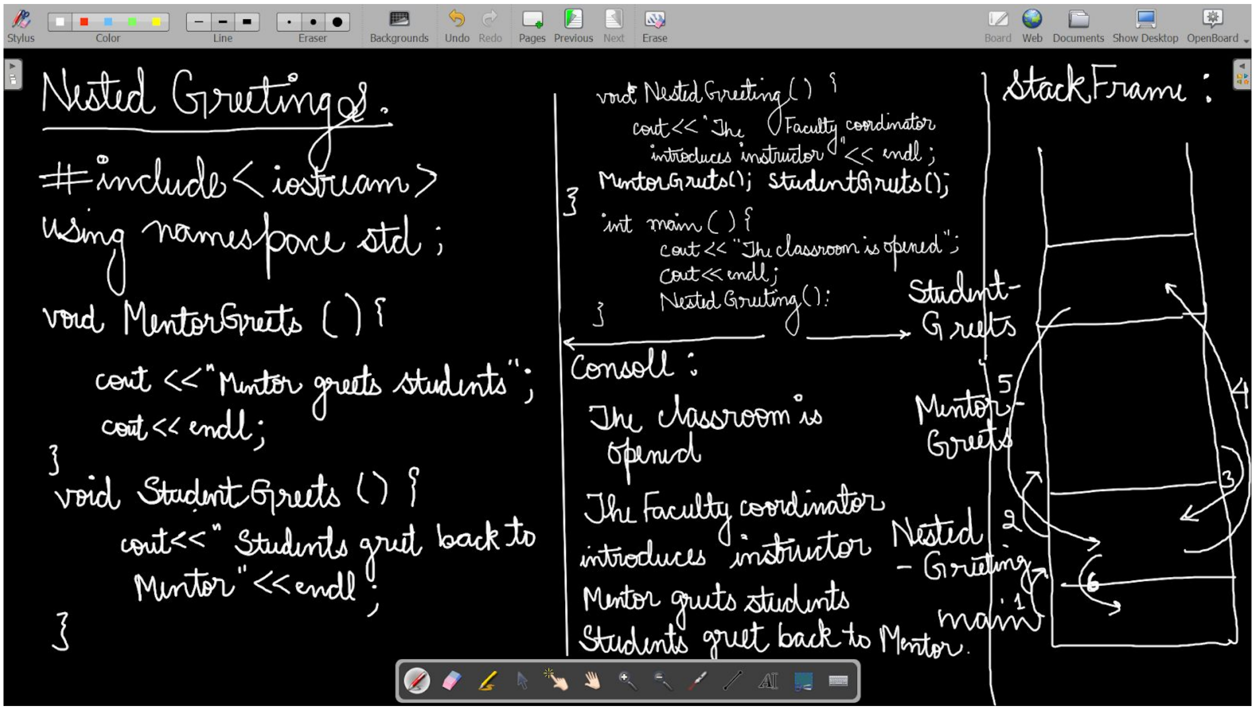
calls the function



```
#include <iostream>
using namespace std;
 void MentorGreets () {
     cout <<"Mentor greets students";
     cout << endl;
 }
 void StudentGreets () {
      cout<<" Students greet back to
       Mentor"<<endl;
 }
 int main() {
      cout<<" The classroom is opened";
      cout << endl;
      MentorGreets ();
 }    StudentGreets();
```

Console output :

```
The classroom is opened
Mentor greets students
students greet back to Mentor
```



Nested Greetings.

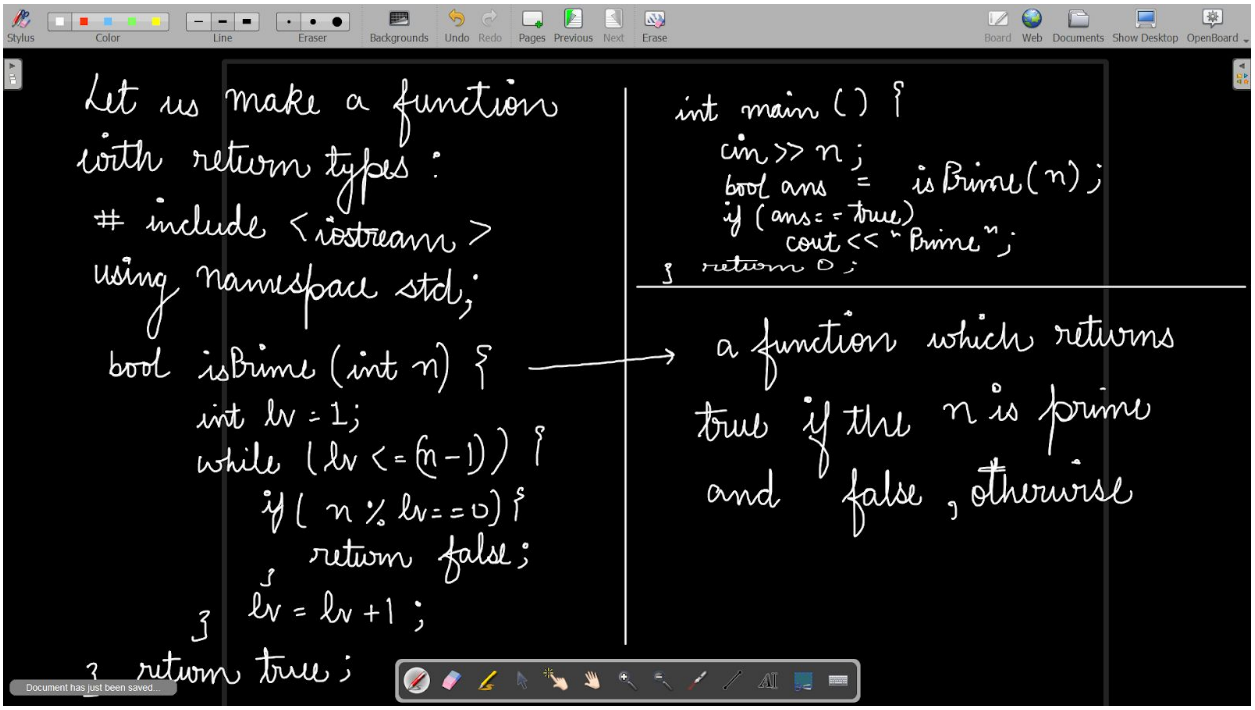
```
#include<iostream>
using namespace std;

void MentorGreets () {
    cout <<"Mentor greets students";
    cout<< endl;
}
void StudentGreets () {
    cout<<" Students greet back to
    Mentor"<<endl;
}
```

```
void NestedGreeting() {
    cout<<"The Faculty coordinator
     introduces instructor"<< endl;
MentorGreets(); StudentGreets();
}
int main () {
      cout<<"The classroom is opened";
      cout<<endl;
      NestedGreeting():
}
```

Consoll :

```
The classroom is
opened
The Faculty coordinator
introduces instructor
Mentor greets students
Students greet back to Mentor.
```

StackFrame :



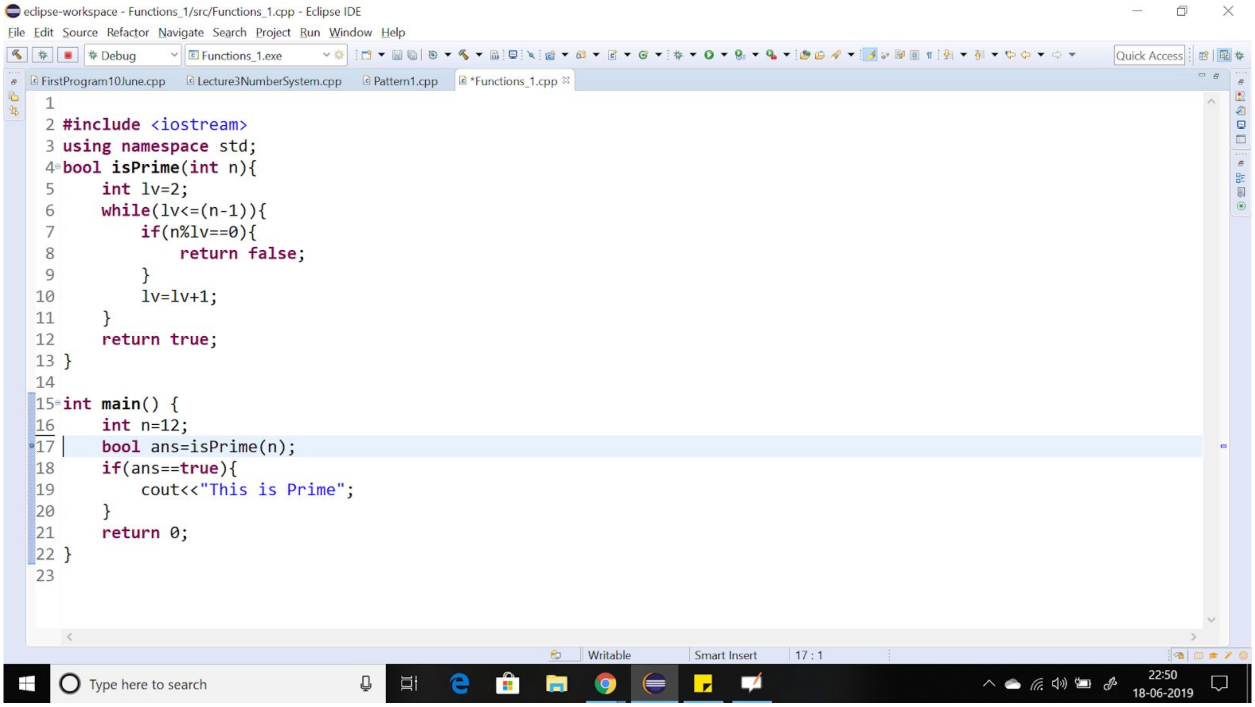
Let us make a function with return types :

```
# include <iostream>
using namespace std;

 bool isPrime (int n) {
        int lv = 1;
        while (lv <= (n-1)) {
            if ( n % lv == 0) {
              return false;
            }
            lv = lv + 1 ;
        }
  } return true;
```

```
int main () {
   cin>> n;
   bool ans  =  isPrime(n);
   if ( ans == true)
       cout << "Prime";
} return 0;
```

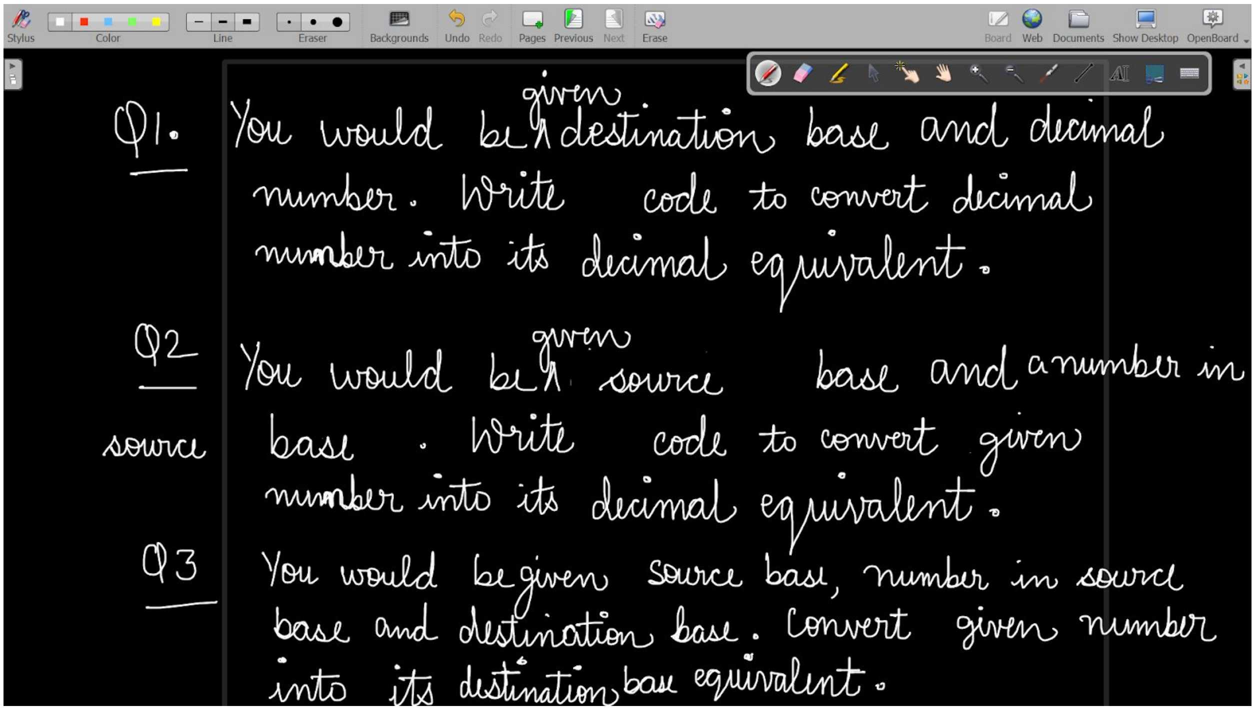
a function which returns true if the n is prime and false , otherwise

Code:



```

#include <iostream>
using namespace std;
bool isPrime(int n){
    int lv=2;
    while(lv<=(n-1)){
        if(n%lv==0){
            return false;
        }
        lv=lv+1;
    }
    return true;
}

int main() {
    int n=12;
    bool ans=isPrime(n);
    if(ans==true){
        cout<<"This is Prime";
    }
    return 0;
}

```

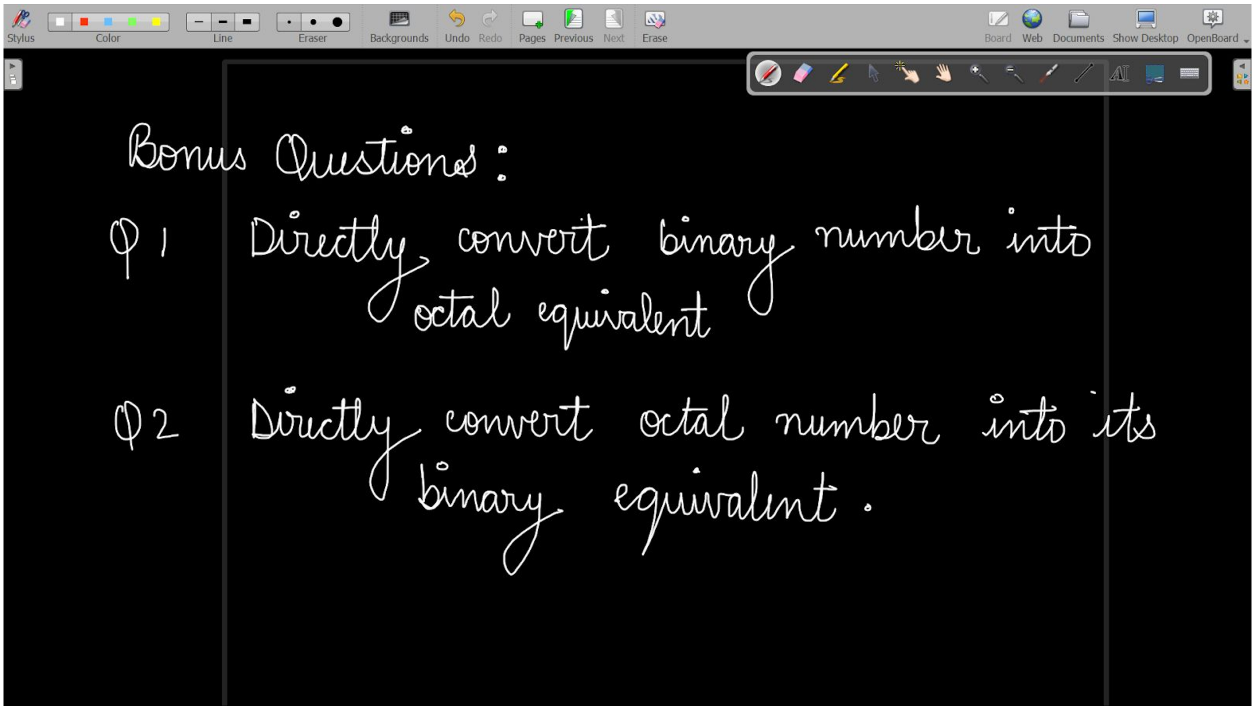
Questions to Practice:



Q1. You would be given destination base and decimal number. Write code to convert decimal number into its decimal equivalent.

Q2 You would be given source base and a number in source base. Write code to convert given number into its decimal equivalent.

Q3 You would be given source base, number in source base and destination base. Convert given number into its destination base equivalent.



Bonus Questions:

Q 1 Directly convert binary number into octal equivalent

Q 2 Directly convert octal number into its binary equivalent.