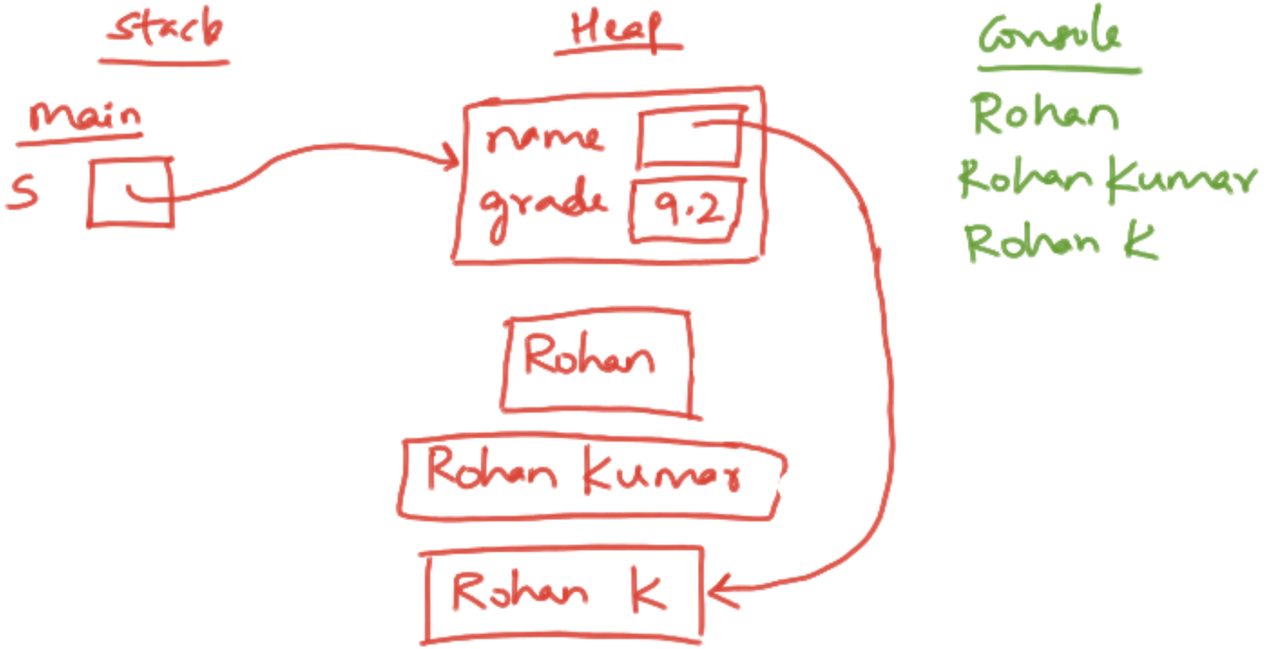
```
class Student {
    String name;
    double grade;

    Student (String n, double g){
        this.name = n;
        this.grade = g;
    }
    void setName(String n){
        this.name = n;
    }
}
class Driver {
    main() {
        Student s = new Student("Rohan", 9.2);
        print(s.name);
        s.name = "Rohan Kumar";
        print(s.name);

        s.setName("Rohan K");
        print(s.name);
    }
}
```

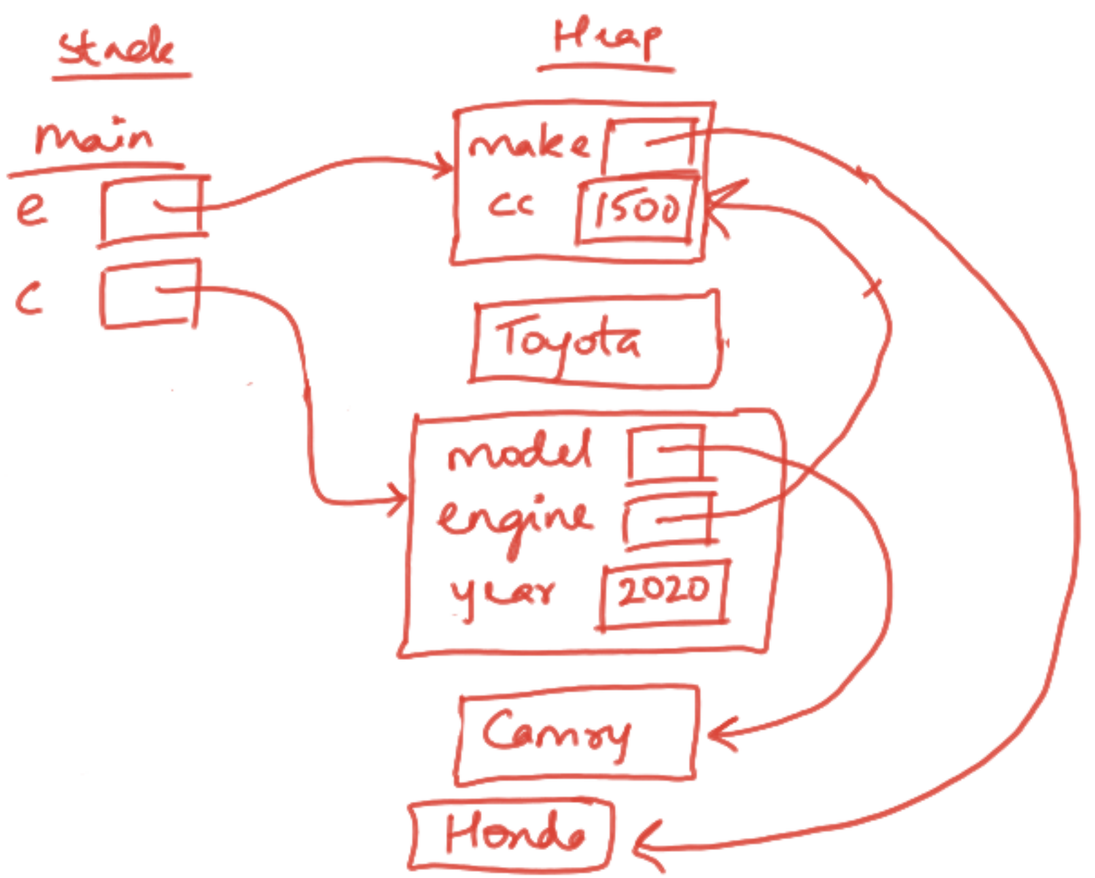


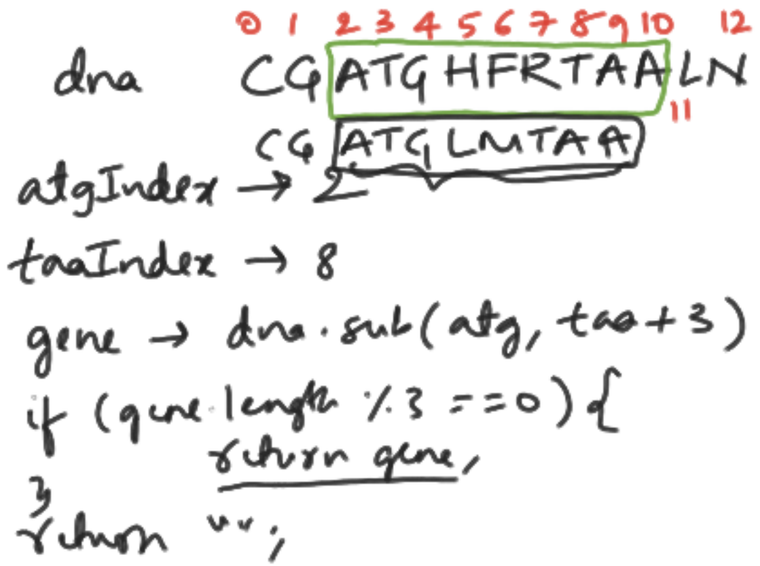
```
class Car {
    String model;
    Engine engine;
    int year;

    Car(String m, Engine eng, int y){
        this.model = m;
        this.engine = eng;
        this.year = y;
    }
}

class Engine {
    string make;
    int cc;

    Engine(Sting m, int c) {
        this.make = m;
        this.cc = c;
    }
    void setMake(string m){
        this.make = m
    }
}
class Driver {
    main(){
        Engine e = new Engine("Toyota", 1500);
        Car c = new Car( "Camry", e, 2020);
        c.engine.setMake("Honda");
    }
}
```





word.substring(a+1, b) without "..."

word.substring(a, b+1)

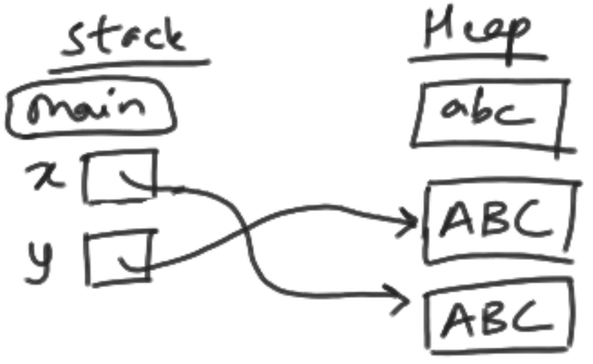
href = "http://YOUTUBE.com/abc"
href = "https://youTUBE.com/def"
href = "http://youtube.com/efg"

yOuTubE
YOUTUBE

```
String x = "abc";
x.toUpperCase();
println(x);
String y = x.toUpperCase();
println(y);
x = x.toUpperCase();
println(x);
```

Console

abc
ABC
ABC



foreach word:

1. check if youtube.com exists (in all cases)
2. convert word to lower case

- Youtube.com
- YOutube.com
- YouTube.com

} youtube.com

abc how ✓
is ABc Or
is aBcen
whynow
weoABC

w → isABcen

y → ABC

wll → ISABCEN

ALC, ABC, aBC, abC

```
beg = item.indexOf("\"");
end = item.lastIndexOf("\"");
print(item.ss(beg+1, end));
```


```
   0 1 2 3 4 5
s = "abcdef"
```

s.ss(0, 6) → abcdef

s.ss(3, 5) → de

s.ss(4, 6) → ef

s.ss(3) → def

s.ss(3, s.length()) → def

```
s="banana"
  0 1 2 3 4 5
```

s.indexOf("an") → 1

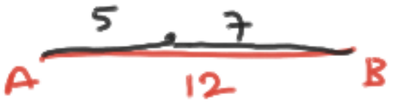
s.lastIndex("an") → 3

s.lastIndex("an", 2) → 1

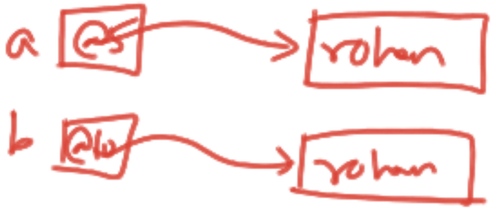
s.indexOf("an", 1) → 1

s.lastIndex("an", 3) → 3

5, 7, 12




```
String a = "rohan";
String b = "rohan";
String c = a;
```




```
a == b      F
b == c      F
a == c      T
a.equals(b);   T
b.equals(a);   T
b.equals(c);   T
c.equals(a);   T
```

1234

① b = 0
d = 4
n = 1234

② b = 4
n = 123
d = 3

③ b = 43
n = 12
d = 2

"abc" + 10 = "abc10"

Stack Heap

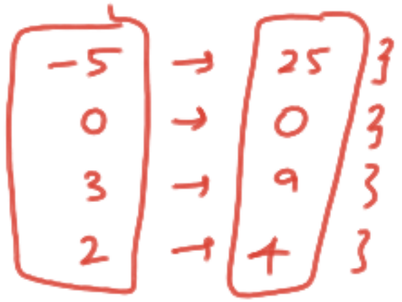
main

testRN2
number 456978
revRes

revNum2
number 456978
back
lastDig 8

Console
8

```
testRN() {
    int a = 321;
    int rev = rN(a);
    print("Reverse of "+a+":"+rev);
    int expected = 123;
    if (rev != expected) {
        print("failed for 321");
    }
}
```



inputs = {-5, 0, 3, 2, -11} for i=0 to 3

expected Values = {25, 0, 9, 4, 121}

```
input = inputs[i];
exp = expVals[i];
ans = sq(input)
if (ans != exp) {print failed}
```

```
int[] intArray = {5, -3, 11, 2};
int[] intArray = new int[] {5, -3, 11, 2};
```
(0 1 2 3)

```
int[] intArray2
      = new int[4];
intArray2[0] = 5;
intArray2[1] = -3;
intArray2[2] = 11;
intArray2[3] = 2;
```

```
// "hello", "world", "abc"
String[] arr1 = new String[3];
arr1[0] = "hello";
arr1[1] = "world";
arr1[2] = "abc";

String[] arr2
   = new String[]{"hello", "world", "abc"};
println(arr2[2]);

int len = arr2.length;
for(int i=0; i<len; i++){
    println(arr2[i]);
}


int x = 5;
int[] arr = new int[x];
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

[1, 2, 3, 4, 5]

```
[
    [1, 2, 3],
    [4, 5, 6]
]
```

```
int[][] array = new int[2][3];
```

```
[
    [1, 2, 3]
    [4, 5, 6]
]
```

array[0][1] → 2

```
int[] x = new int[] {7, 9, 13};
int y = x[0];
```

```
int[][] array = new int[][] { {1, 2, 3}, {4, 5, 6} }
```

array[0] → {1, 2, 3}

array[1][2] → 6

array[2][1] → Error

```
int[] x = array[1];
```

x[0] → 4

x[2] → 6

x[3] → Error

```
int y = x[1];
```

y → 5

```
String[] a = array[0];
```
→ Error

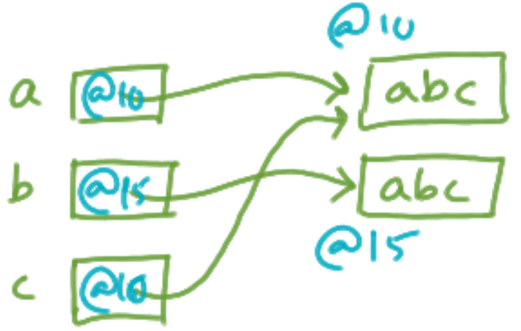
```
int[][] abc = new int[][]{
                    { 1, 2, 3},
                    { 4, 5, 6}
              };
```

```
int x = 5
int y = 7
if (x == y) {
    print ("x is equal to y");
}
```

x 5

y 7

```
String a = "abc";
String b = "abc";
String c = a;
```



a == b → false

b == c → false

a == c → true

a.equals(c) → true

c.equals(b) → true

c == c → true

b.equals(b) → true

{1, 2, 3}

{

{1, 2, 3},

{4, 5, 6}

}

{

{

{1, 2, 3},

{4, 5, 6}

}

2D

{

{1, 2},

{3}

}

2D

}

array[1][0][1]

{1, 2, 3}

{

{1, 2, 3}

}

{

{3},

{1, 2}

}

arr[0][0]

arr[0] → {1, 2, 3}

1

{

1

}

Holiday
0 1 2 3 4 5 6

Madam
0 1 2 3 4

f	b
0	6 = len-1 = len-f-1
1	5 = len-2 = len-f-1
2	4 = len-3 = len-f-1

$0 \le f < floor(l/2)$

AAAAAA
0 1 2 3 4 5

f → 0, 2

$5! = 5 \times 4 \times 3 \times 2 \times 1$
$= 5 \times 4!$

$fac(n) = n \times (n-1) \times (n-2) \times \cdots \times 1$
$= n \times fac(n-1)$

Recursion

```
int factorial (int n) {
    int prod = 1;
    for ( i=1 ; i <= n ; i++){
        prod = prod * i;
    }
    return prod;
}
```

```
int factorial(int n) {
    if (n==1) {
        return 1;
    }
    return n * factorial(n-1);
}
```

0 1 2 3 4
Madam

f b

f = 0
b = len - 1 = 4

f = 1
b = 3

f = 2
b = 2

ArrA

f = 0
b = 3
↓
f = 1
b = 2
↓
f = 2
b = 1

```
isPalindrome(String word){
    int f = 0
    int b = word.length() - 1
    while (f<b) {
        char fChar = w.charAt(f)
        char bChar = w.charAt(b)
        if(fChar != bChar) {
            return false;
        }
    }
    return true;
}
```

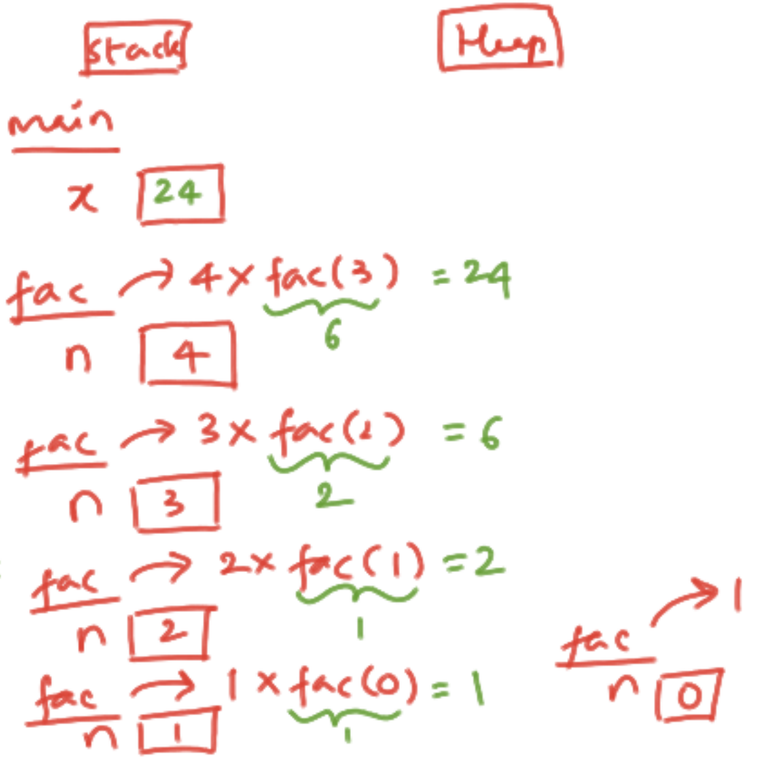
factorial(n) = n × (n-1) × ... × 2 × 1

factorial(n) = n × factorial(n-1)

$$fac(n) = \begin{cases} 1 & \text{if } n = 0 \\ n \times fac(n-1) & \text{otherwise} \end{cases}$$

fac(3) = 3 × fac(2)
= 3 × (2 × fac(1))
= 3 × 2 × 1 × fac(0)
= 3 × 2 × 1 × 1
= 6

```
fac(int n){
   if(n==0){
      return 1;
   }
   return  n * fac(n-1);
}
main {
      int x = fac(4)
}
```



$f(n) = 1^2 + 2^2 + 3^2 + \cdots + n^2$

```
f(int n) {
    int sum = 0
    for i = 1 to n {
        sum += (i*i)
    }
    return sum;
}
```

$$f(n) = \underbrace{1^2 + 2^2 + 3^2 + \cdots + (n-1)^2}_{f(n-1)} + n^2$$

$$f(n) = n^2 + f(n-1)$$

$$f(4) = \underbrace{1^2 + 2^2 + 3^2}_{f(3)} + 4^2$$

$$= f(3) + 4^2$$

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ n^2 + f(n-1) & \text{otherwise} \end{cases}$$

```
f(int n) {
    if (n == 0) {
        return 0;
    }
    return (n*n) + f(n-1);
}
```

m[ada]m

boolean isP(String w){

isP(word) = (first == last)
&& (isP(middlePart))

isP(madam) = (m==m) && isP(ada)

isP(ada) = (a==a) && isP(d)

isP(a) = (a==a) && isP("")

isP("") = true

```
boolean isP(String w){
    int len = w.length();
    if(len==1 || len==0){
        return true;
    }
    char firstChar = w.charAt(0)
    char lastChar = w.charAt(len-1)
    String mid = w.substring(1, len-1);
    ...
```

```
...
if((firstChar==lastChar) &&
    isP(mid)){
    return true;
}
return false;
}
```

$f(n) = 1 + 2 + 3 + \cdots + (n-1) + n$

```
int f (int n) {
    int sum = 0;
    for i (1 to n) {
        sum += i
    }
    return sum;
}
```

$f(5) = 1 + 2 + 3 + 4 + 5$

$= 5 + f(4)$

$f(n) = \boxed{1 + 2 + 3 + \cdots + (n-1)} + n$

$= n + f(n-1)$

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ n + f(n-1) & \text{otherwise} \end{cases}$$

```
int f(int n) {
    if (n == 0) {
        return 0;
    }
    return n + f(n-1);
}
```