8 primitive Data types in Java

→ long , int, short, byte → 10, 20, 98
8B 4B 2B 1B

→ double, float Precision (double) > precision (float)
8B 4B → 5.62, 9.0, 6.85

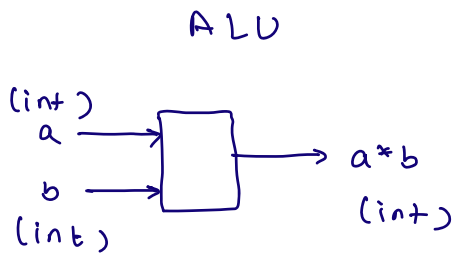
→ char : 'a' - 'z', 'A' - 'Z', *, -, @, '0' - '9'
1B

→ boolean — true
1B — false

```
void main ( ) {
    int a = 100;
    int b = 20;
    long c = a * b;
    SOP (c);
}
```

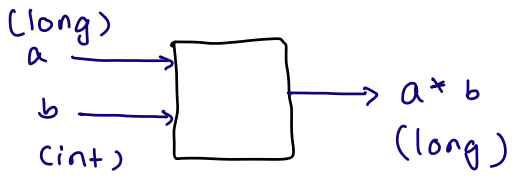
We can store smaller sized data type value in a larger size container but vice versa is not true.

```
void main () {
    int a = 100000;
    int b = 400000;
    long c = (long)(a*b);
    Sop (c);
}
```



c → some random value

```
void main () {
    int a = 100000;
    int b = 400000;
    long c = (long) a * b;
    Sop (c);
}
```



c → 4000000000

```
void main () {
    int a = 100000;
    int b = 400000;
    long c = a * (long) b;
    Sop (c);
}
```

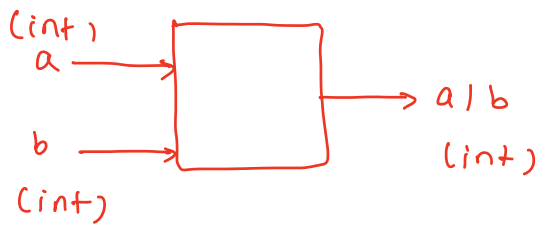
```
void main ( ) {
    float a = 5.2;
    SOP (a);
}
```

↳ double (8 bytes)

Error: possible lossy conversion;

```
void main ( ) {
    float a = 5.2f;
    SOP (a);
}
```

```
void main ( ) {
    int a = 5;
    int b = 2;
    float c = a / b;
    SOP (c);
}
```



↳ 2.0

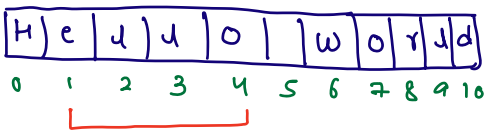
```
void main ( ) {
    char ch = 'a';
    SOP (ch);
}
```

ASCII : American standard for Information interchange

'a' → 97	'A' → 65	'0' → 48
'b' → 98	'B' → 66	'1' → 49
'c' → 99	'C' → 67	'2' → 50
⋮	⋮	⋮
'z' → 122	'Z' → 90	'9' → 57

Strings

└ sequence of chars




```
void main() {
    String str = "Hello world";
    SOP(str.length());  ⟶ 11
    char ch = str.charAt(2);  ⟶ l
    SOP(str);  ⟶ Hello world
    for (int i=0; i<str.length(); i++) {
        SOPln(str.charAt(i));
    }
    String ss = str.substring(1,5);  ⟶ ello
```

str.substring(i, j)

↳ from i to j-1

[i, j)

Q.1 Given a String, write a function that counts the no. of uppercase letters in String and return it.

	count
Str: aBcMn	2
Str: Manisha Pawar	2
Str: KLONC	3

```
int countUppercase (String str) {
    int count = 0;
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i)
        if (ch >= 65 && ch <= 90) {
            count;
        }
    }
    return count;
}
```

↳ ch >= 'A' && ch <= 'Z'

Q.2 Given a String, write a function that returns true if the String is Palindromic.

eg: madam, nitin, noon, mom, dad, wow

```
Str = reverse(str);
```

Str = m a d a m

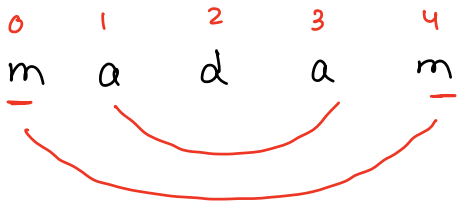
rev = m a d a m

s1.equals(s2) → true / false

Str = a b c a

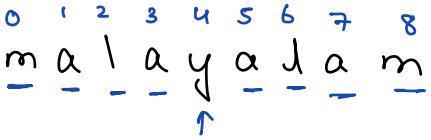
rev = a c b a

```
if (str.equals(rev) == true) {
        return true;
}
else {
        return false;
}
```



str.charAt(0) = str.charAt(4)

str.charAt(1) = str.charAt(3)



i

str.charAt(0) = str.charAt(8) j

str.charAt(1) = str.charAt(7)

str.charAt(2) = str.charAt(6)

str.charAt(3) = str.charAt(5) i < j

redundant

str.charAt(4) = str.charAt(4)

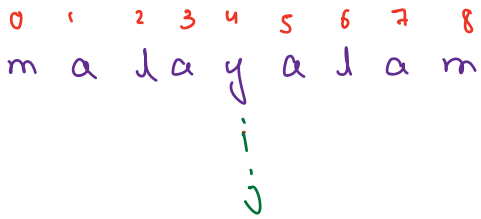
str.charAt(5) = str.charAt(3)

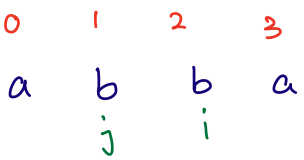
str.charAt(6) = str.charAt(2)

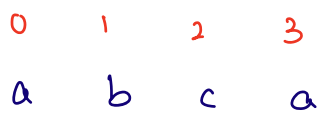
str.charAt(7) = str.charAt(1)

str.charAt(8) = str.charAt(0)

```
int i = 0 , j = str.length() - 1;
while (i < j ) {
    if (str.charAt(i) != str.charAt(j)){
        return false;
    }
    i++;
    j--;
}
return true;
```







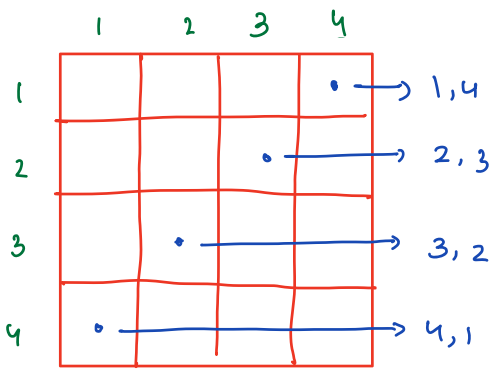
Doubts

```
while (i != j) {
    if (str.charAt(i) != str.charAt(j)) {
        return false;
    }
    i++;
    j--;
}
```

0	1	2	3
a	b	b	a

-1 → j

4 → i



i + j = M + 1

1, 4

↓

2, 3

↓

3, 2

↓

4, 1

Wave print : col wise

0	1	2	3
1	2	6	10
4	8	3	5
7	9	5	16

output : 1 4 7 | 9 8 2 | 6 3 5 | 16 5 10

0th col ↓ | 1th col ↑ | 2nd col ↓ | 3rd col ↑

```
for (int j = 0; int j < M; j++) {
    if ( j % 2 == 0) {
        for (int i = 0; i < N; i++) {—3
    3
    else {
        for (int i = N-1; i >= 0; i--) {—3

    3
}
```