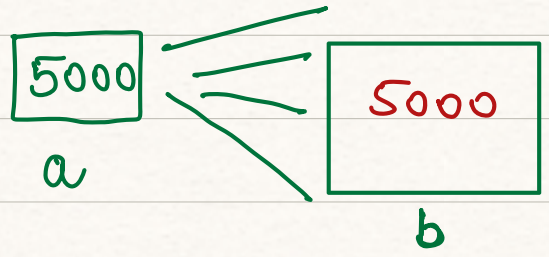
Data Types		Size (bytes)	Range
Integers	int	4	-10^9 to 10^9
	long	8	-10^{18} to 10^{18}
Decimal	float	4	
	double	8	
true/false	boolean	1	
text	String		

```
import java.util.*;
=
=
=
Scanner scn = new Scanner (System.in);

int a = scn.nextInt();
float b = scn.nextFloat();
```

Typecasting

```
int a = 5000;
long b = a;
System.out.print(b);
```

→ 5000

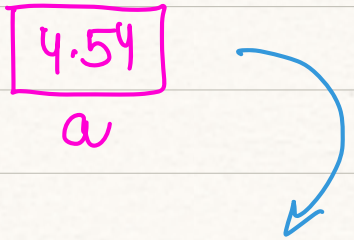


Widening typecasting → [automatically]

int → long → float → double

No loss in data

```
float a = 4.54f;
long b = a; // Error
```

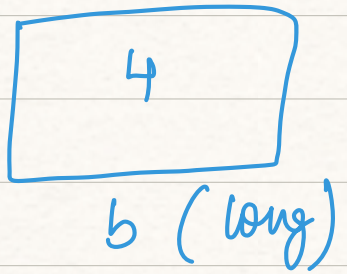


0.54

lossy conversion

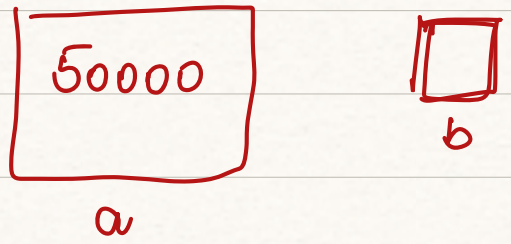
[float → long]

↳ Narrowing




```
long x = 500000l;
float y = x;
```

```
long a = 50000l;
int b = a; ⟶ Error
System.out.println(b);
```



lossy conversion from long to int

```
long a = 50000l;
int b = (int) a;
System.out.println(b);
```

⟶ 50000

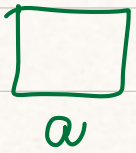
Narrowing typecasting → [forcefully]

double ⟶ float ⟶ long ⟶ int

Q1.

10^{11}

```
int a = 10000000000;
int b = 1000;
S.O.P (a*b);
```

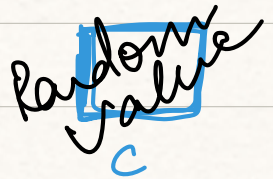


Error

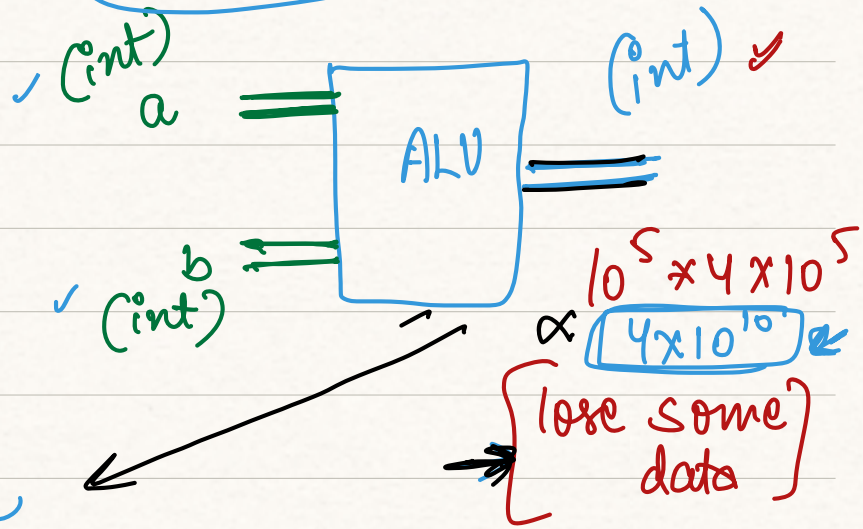
[Integer number too large]

Q2.

```
int a = 100000;
int b = 400000;
int c = a*b;
System.out.println(c);
```



$-10^9 - 10^9$

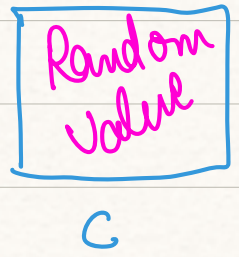


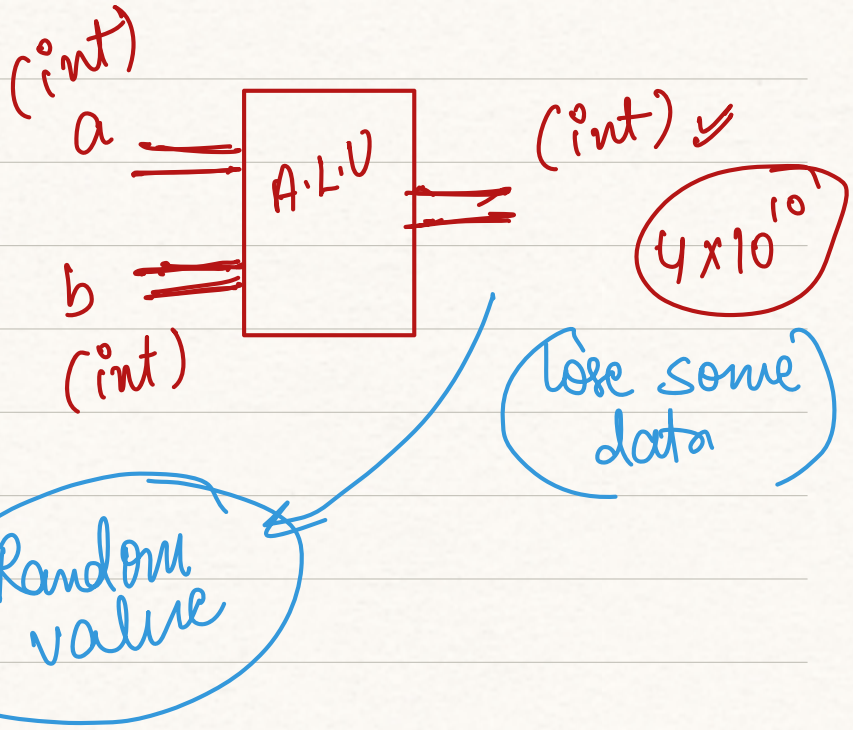
Random Value

ALU → Arithmetic and Logical Unit

Q3.

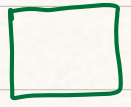
```
int a = 100000;
int b = 400000;
long c = a * b;
System.out.println(c);
// garbage value
random
```

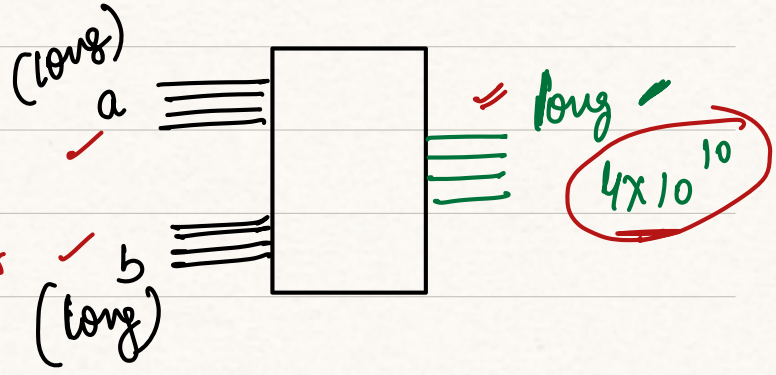




Q4.

```
long a = 100000;
long b = 400000;
→ int c = a * b; //Error
System.out.println(c);
```





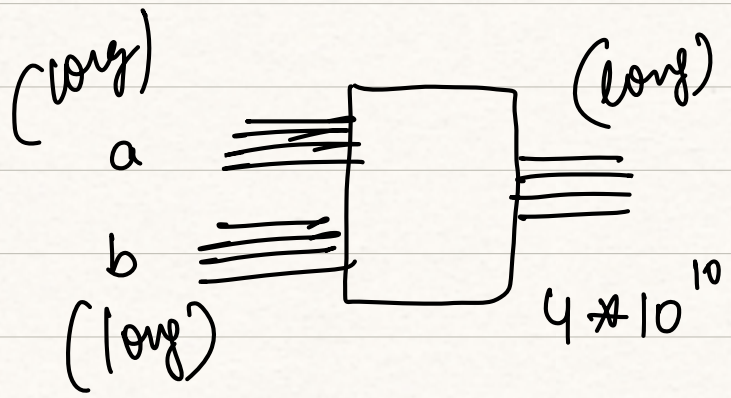
Q5.

```
long a = 100000;
long b = 400000;
long c = a * b;
System.out.println(c);
```

$4 * 10^{10}$

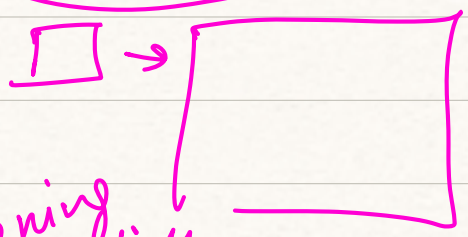
c

10^5



100000 l

int → long



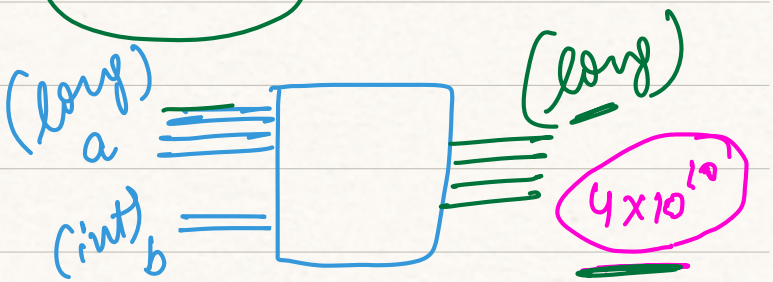
Widening typecasting

```
long a = 100000;
int  b = 400000;
long c = a * b;
System.out.println(c);
```

4×10^{10}

c

4×10^{10}



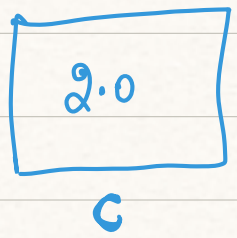
int * int ⟶ int

long * long ⟶ long

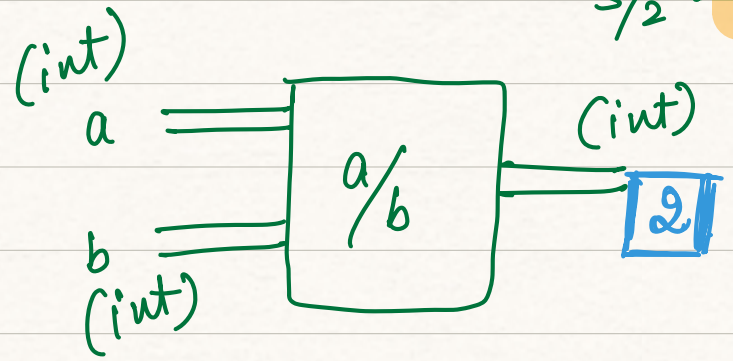
int * long ⟶ (long)

```
int    a = 5;
int    b = 2;
double c = a/b;
System.out.println(c);
```

(int)



5/2 = 2.5



1) 2.5

2) 2.0

✗ 3) Error

int → double

Widening

Q6.

```
int a = 100000;
int b = 400000;
long c = (long) (a*b);
S.O.P (c);
```

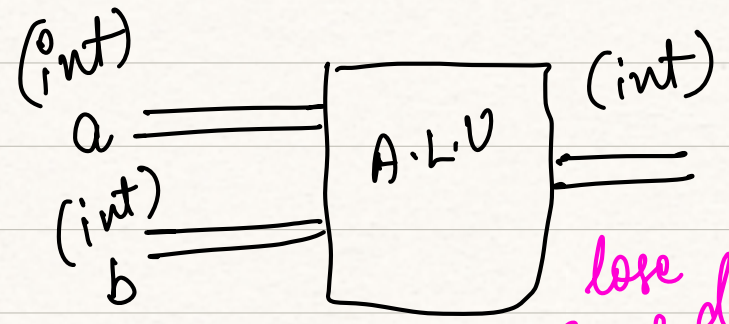
Random value

c

4×10^{10}

↳ Random values

BODMAS



lose some data

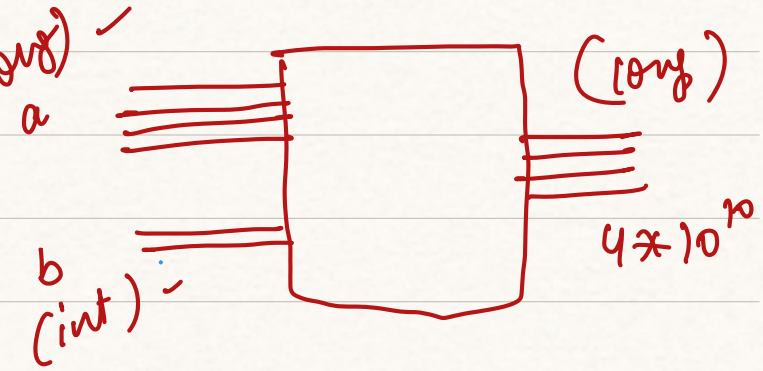
(Random value)

Q7.

```
int a = 100000;
int b = 400000;
long c = (long) a*b;
```

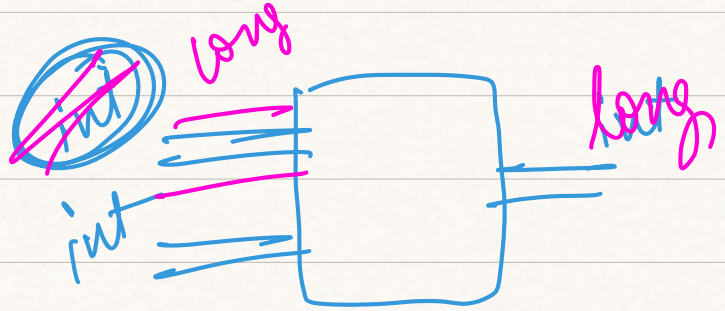
4×10^{10}

c



the data type of a is not changed

We are just telling the computer to assume it is of long type



Operators

1.) Arithmetic operators → + , − , * , / , %

% → modulus [remainder]

10 % 3 = 1

40 % 5 = 0

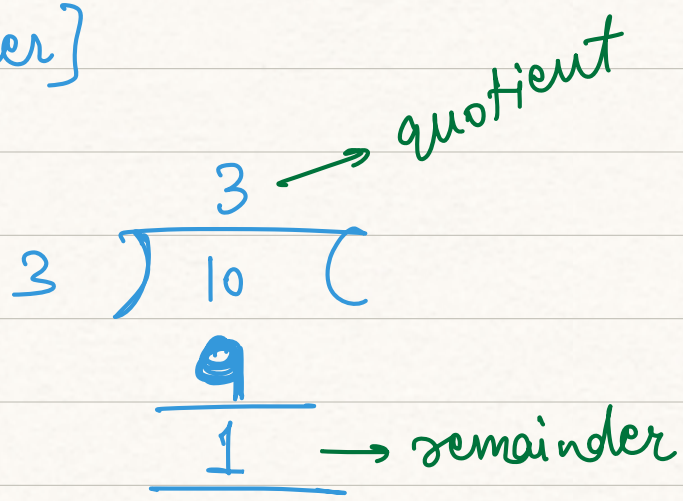
35 % 3 = 2

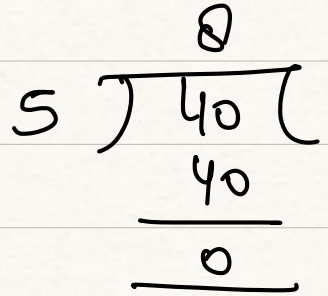
20 % 2 = 0

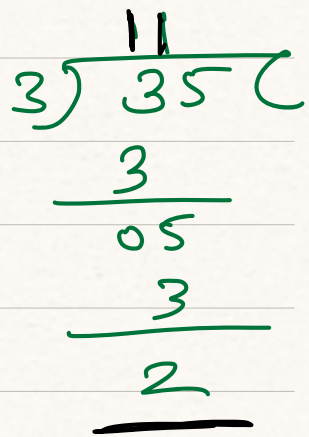
15 % 2 = 1

48 % 2 = 0

[the remainder of even number by 2 is 0]







Ques.

```
int  a = 1000 0000;
long b = 10;
```

is a greater b ⟶ true

Ques.

```
int  a = 10;
long b = 10;
```

is a equal to b ⟶ true

2.) Relation operators

		a=29, b=45	a=36, b=16
a is greater than b	a > b	false	true
a is less than b	a < b	true	false
a is greater than or equal to b	a >= b	false	true
a is less than or equals to b	a <= b	true	false
a is equals to b	a == b	false	false
a is not equals to b	a != b	true	true

a = 20 b = 20

a >= b ⟶ true

a <= b ⟶ true

a > b ✗ a == b ✓

a < b ✗ a == b ✓

placements →
(Are you eligible for placement)

cgpa > 6.0 (A)
AND
no. of active backlogs ≤ 2 (B)

Warranty →
[free services]
will you not get free service

distance > 75000 km ✗
OR
no. of years > 5 ✗

3.) Logical operators

AND → &&

OR → ||

NOT → !

A	B	A && B
T	T	True
T	F	False
F	T	False
F	F	False

(7.0) (3) (5.5) 4 6 5

(distance) (no. of years)

A	B	A || B
T	T	T
T	F	T
F	T	T
F	F	F

80000 80000 50000 40000 3 7 2

A → true

!A → false

B → false

!B → true

(5 < 3) || (8 < 5)

false, false

false

(5 < 3) || (8 > 5)

false, true

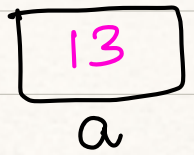
true

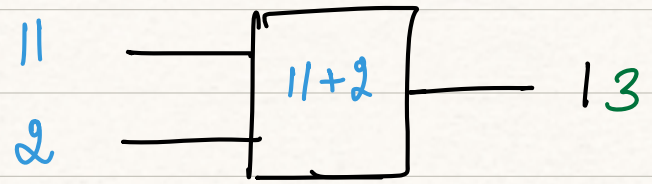
(5 > 3) && (8 > 5)

true, true

true

```
int a = 10;
a = a+1;
System.out.println(a);   ⟶ 11
a = a+2;
System.out.print(a);     ⟶ 13
```



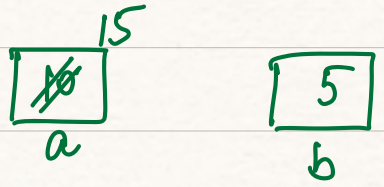


Shortcuts

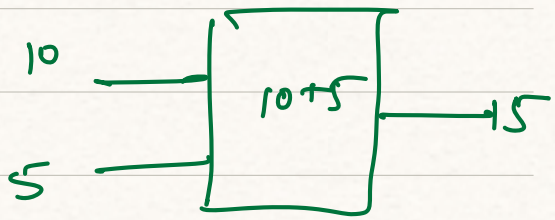
```
a = a+1;   ⇒   a++;   (inc. the value by 1)
a = a+2;   ⇒   a+=2;

a = a-1;   ⇒   a--;   (dec. the value of a by 1)
a = a-2;   ⇒   a-=2;
```

```
int a = 4;
a = a+10;  →  a += 10;
System.out.println(a);  ⟹  14
```


```
int a = 10;
int b = 5;
a = a+b;
a += b;
System.out.println(a);
```



15




```
int b = 8;
b = b-1;
b--;
System.out.println(b);  → 7
b++;
System.out.println(b);  → 8
```

8 → 7 → 8

b = b+1;

Unary operator

a = a+1

(Post increment) a++ ;

++a ; (Pre increment)

```
int a=10;
a++; ⟶ a = a+1;
System.out.println(a);
        ↳ 11
```

```
int a=10;
++a; ⟶ a = a+1;
System.out.println(a);
        ↳ 11
```

[Post increment]

```
int a=10;
int b = a++;   // first the value will be used after that it will increment
System.out.println(b); → 10
System.out.println(a); → 11
```

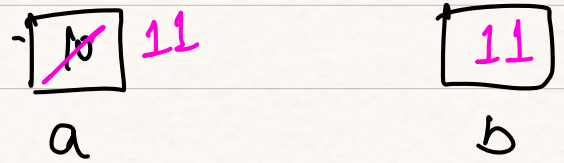
a	b
~~10~~ 11	10

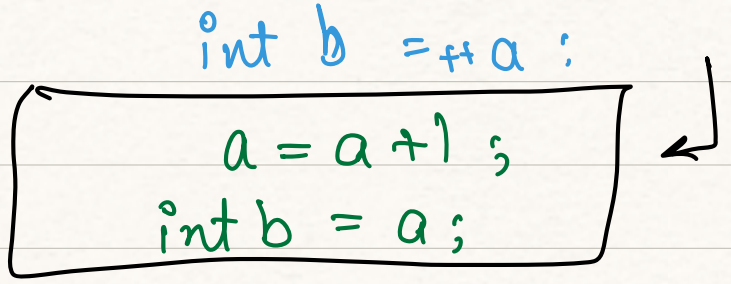
```
int b = a++;
```

```
int b = a;
a = a+1;
```

[Pre increment]

```
int a = 10;
int b = ++a;        // first increment and then use the value
System.out.println(b);   ⟶ 11
System.out.println(a);   ⟶ 11
```





Doubts

Sum 3

```
Scanner scn = new Scanner (System.in);
int n1 = scn.nextInt();
int n2 = scn.nextInt();
int n3 = scn.nextInt();

int sum = n1+n2+n3;

System.out.print(sum);
```

MCQ

```
int a = 10;
int b = 20;
```

$\frac{10}{20} = 0.5$

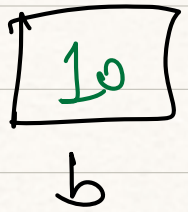
```
int c = a/b; // c = 0

int d = b/c;
```

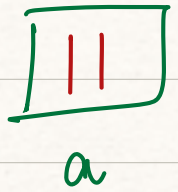
$\frac{20}{0}$

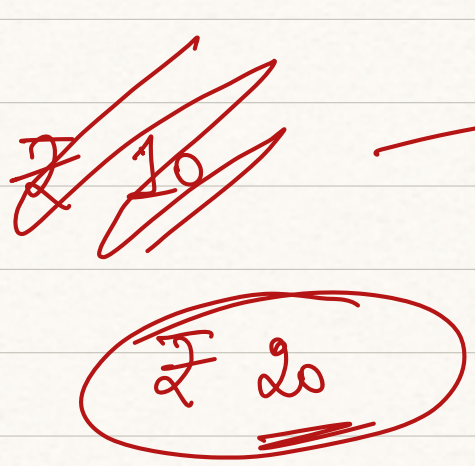
```
a = 10;

b = a++;
```

↳ `int b = a;`
`a = a + 1;`

```
b = (a = a+1)

b = ++a;
```







notes = 10

old money = budget

20

$\frac{\text{budget}}{\text{new amount}}$

$\frac{126.3}{5} = 25.26$

25

40 min

T = 20;

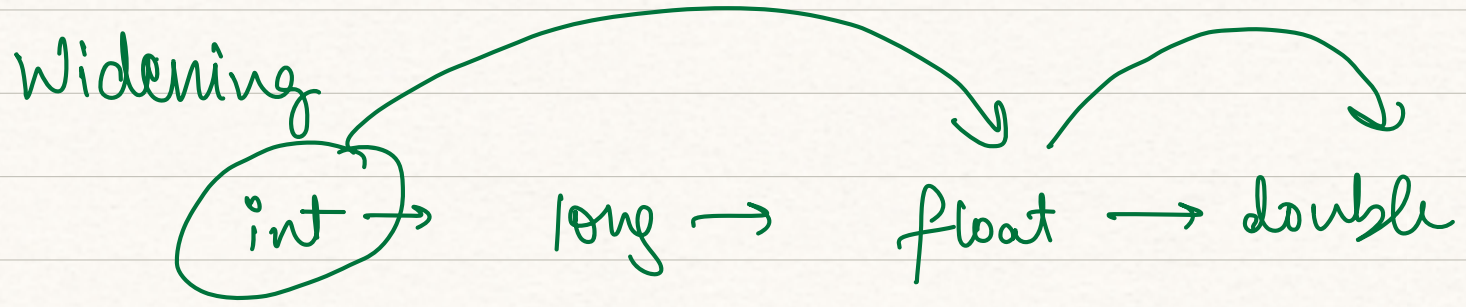
20 min

20

T = 15 min;

40 - T

ans = 25




```
int n = Scn.nextInt()
```

Total amount = 126

amount Spent = 14

amount left = 112

T.A → 116

Spent → 12

left → 104

$a = 10$, $b = 5$, $c = 3$

BODMAS

int d = a + c / 2 * b ;

$\frac{3}{2} = 1$

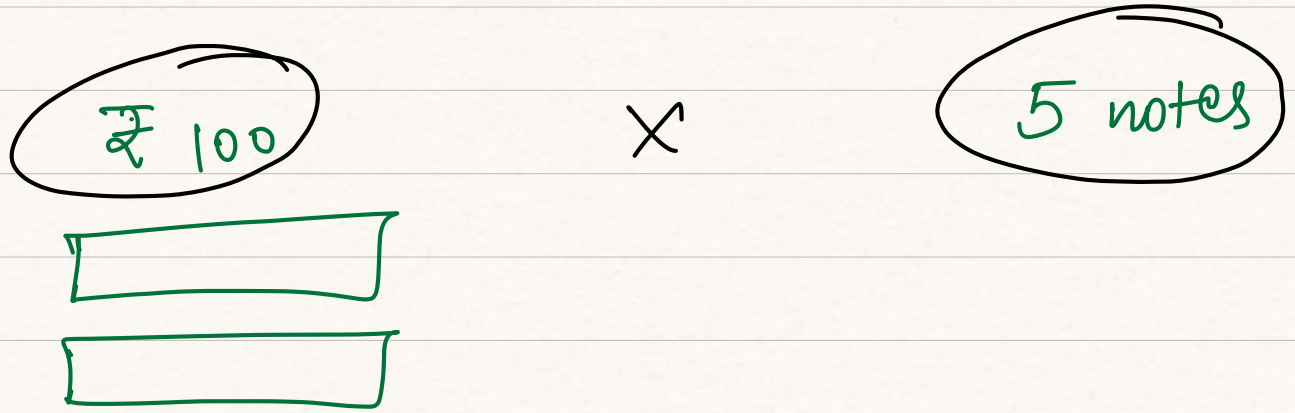
$10 + (3/2) * 5$

$10 + (1 * 5)$

$10 + 5$

$d = 15$

```
Scanner scn = new Scanner(System.in);
int budget = scn.nextInt(); // 124
int spent = scn.nextInt(); // 14
int left = budget - spent; // 124-14 = 110
System.out.println(left);
```



total amount = 500