

## Number in words

235 → Two Hundred thirty five.

4,567  
X XX XX

(5) (35) DIV 1000 → Q: 5  
R: 35

→ Count number of digits in Number

① Get a Number → If Number is 0 then  
Print "Zero" STOP

② Count number of digits in Number  
→ digits.

③ if digits EQUALS 5 then

④ Thousands = Quotient of  
Number DIV 1000

⑤ Number = Remainder of  
Number DIV 1000

⑥ if digits EQUALS 4 then

⑦ Thousands = Quotient of Number  
DIV 1000

⑧ Number = Remainder of Number  
DIV 1000

⑨ Print Thousands in word.

⑩ Print "Thousand"

⑪ Count number of digits in Numbers  
- digits.

⑫ If digit EQUALS 3 then

⑬ Hundred = Quotient of  
Number DIV 100

⑭ Number = Remainder of  
Number DIV 100

⑮ Print Hundred in words

⑯ Print "Hundred"

⑰ Print Number in words.

⑱ Stop

Print a two digit number into words

1	2	3	4	5	6	7	8	9
one	two							nine
10	11	12	13					14
Ten	Eleven							Nineteen

<u>20</u>	21	...	<u>29</u>
Twenty	Twenty One		Twenty Nine
<u>30</u>	31	...	<u>39</u>
Thirty	Thirty One		Thirty Nine

① if Number  $\geq 10$  AND Number  $\leq 19$

② if Number is 10 then  
Print "Ten"

③ if Number is 11 then  
Print "Eleven"

...

⑨ if Number is 19 then  
Print "Nineteen"

⑩ Stop

⑪ if Number  $> 19$  then

⑫ TensDigit = Quotient of  
Number DIV 10

⑬ Number = Remainder of  
Number DIV 10

⑭ if TensDigit EQUALS 2

⑮ Print "Twenty"

⑯ if TensDigit EQUALS 3

(17) Print "Thirty"

⋮

(20) IF Number EQUALS 1 then  
(21) Print "One"

(22) IF Number EQUALS 2 then  
(23) Print "Two"

⋮

(30) Stop

---

Number → ~~4000~~ 0

4000 DIV 1000 → Q: 4  
→ R: 0

Thousands → 4

Four Thousand

# Add 3 numbers

- ① Add first and second numbers and find sum
- ② Add sum and third number to find result.
- ③ Stop.

Variables  $\Rightarrow$  to store values  
< data type > < variable name >

```
int first;  
int second;  
int third;  
int sum;  
int result;
```

Take Input  
from User

$\Downarrow$   
No support  
from language  
for doing I/O.  
in C.

---

We use library  
functions.

```
scanf ("%d", &first);
```

$\uparrow$  Read  
a integer  
 $\downarrow$  store  
value at  
location  
of  
variable  
first  
 $\swarrow$  address

Output  $\rightarrow$  printf ( )  
Input  $\rightarrow$  scanf ( )

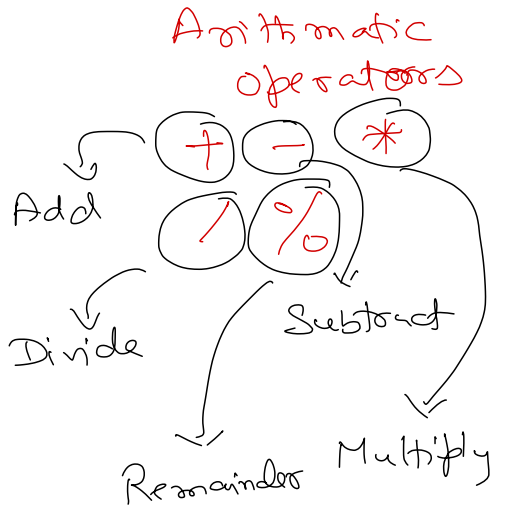
```
scanf ("%d", &second);  
scanf ("%d", &third);
```

sum = first + second;  
↪ assignment operator

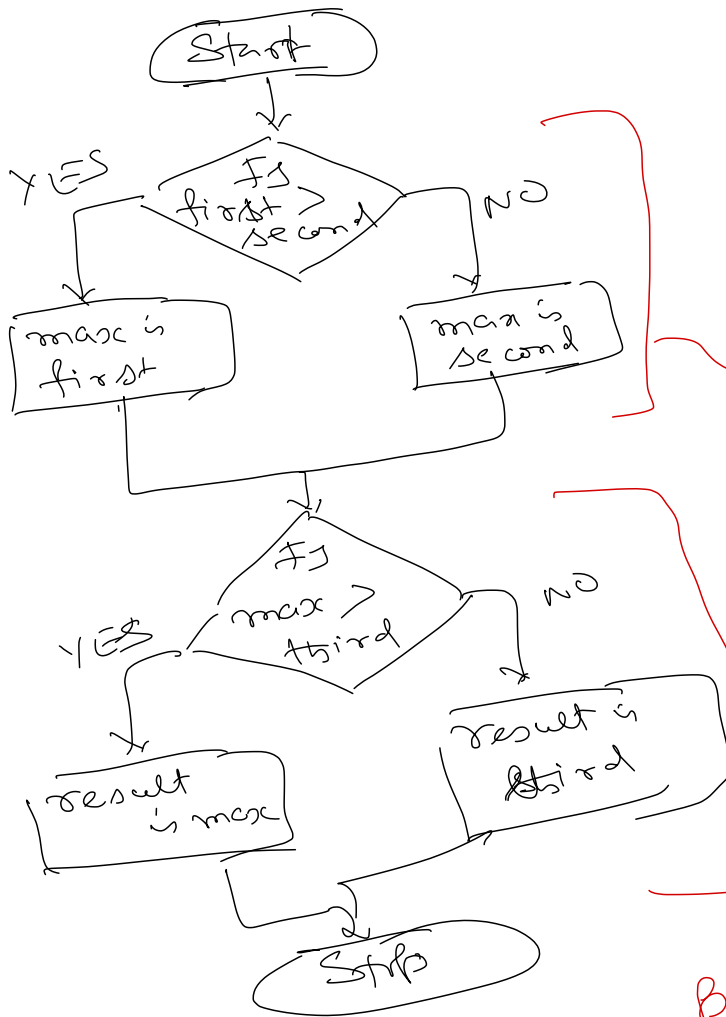
var = value

result = sum + third;

printf ("%d", result);  
Print a integer      integer value to be printed.



Find max of 3 numbers



```
int first;  
int second;  
int third;  
scanf("%d", ...)
```

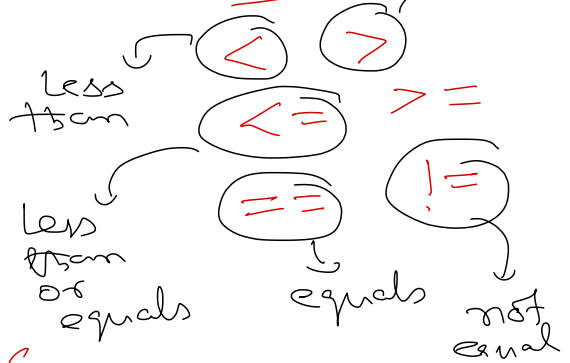
```
if (first > second)  
    max = first;  
else  
    max = second;
```

```
if (max > third)  
    result = max;  
else  
    result = third;
```

```
printf("%d", result);
```

greater than

## Relational Operators



if ← keyword  
else  
using which  
we can write  
conditional  
statements in C

if ( $<\text{condition}>$ ) TRUE [else, ...] FALSE

variable  
initialization

Find sum of first N numbers.

① Get value of N.

② Set sum to 0

③ Set no to 1.

④ Repeat following while  $\text{no} \leq N$ .

⑤ Add no to sum.

⑥ Increment no by 1

⑦ Stop.

int no;  
int n;

int sum = 0;

scanf("%d", &n);

no = 1;

while ( $\text{no} \leq N$ ) {

sum = sum + no;

no = no + 1;

}



## Loop

while  $\leftarrow$  we write a loop in C

while (<condition>)  
<statement>

while the condition is true execute statement



block of statements

var = var + val;

var {  
+ =  
- =  
\* =  
/ =  
}

$\rightarrow$  var = var \* 10;  
 $\rightarrow$  var \*= 10;

x = y + 10; ~~\*~~ write using short form operator.

Unary increment / decrement

++ --

$\rightarrow$  Increment / Decrement value by 1.

$var = var + 1;$

$var += 1;$

$++ var;$  Pre increment

$var = var - 1;$

$var -= 1;$

$-- var;$

Postfix Pre decrement

$var = var + 5;$

$var++;$

Post Increment

$var--;$

Post Decrement

Postfix

Count the number of digits in a positive integer

- ① Get No.  $\rightarrow$   $\text{int no;}$   
 $\text{int count;}$   
 $\text{scanf("%d", \&no);}$
- ② Set count to 1.  $\rightarrow \text{count} = 1;$
- ③ while No  $\geq 10$  do  $\rightarrow \text{while (no} \geq 10)$   
 $\{$
- ④ Set No to quotient of No / 10.  $\rightarrow \text{no} = \text{no} / 10;$   
 $\rightarrow \text{count} = \text{count} + 1$
- ⑤ Increment Count by 1.  $\}$
- ⑥ Stop.

÷      %  
Division      Remainder

↓  
 When dividing two integers, it returns quotient

## Reverse a Number

- ① Get a Number. → `int reverse;`
- ② Set Reverse to 0. → `int number;`
- ③ Repeat following steps while Number is not zero. → `scanf("%d", &number);`
- ④ Digit = remainder of Number DIV 10. → `reverse = 0;`  
`while (number != 0)`  
`{`  
`digit = number % 10;`
- ⑤ Multiply reverse by 10. → `reverse = reverse * 10;`
- ⑥ Add Digit to reverse. → `reverse = reverse + digit;`
- ⑦ Set Number to quotient of Number DIV 10. → `number = number / 10;`
- ⑧ Stop.

# Fibo Series

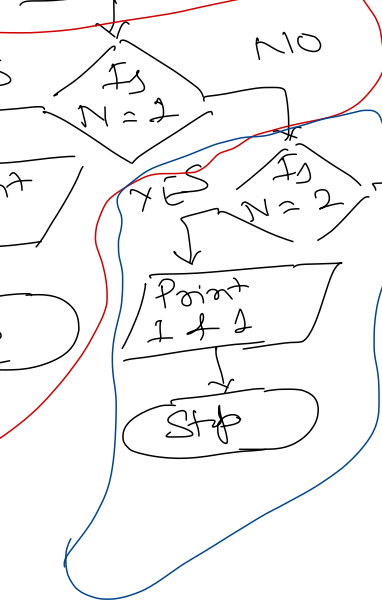
```

int term2;
int term1;
int n;
int count;

```

Start

Get N



```

scanf("%d", &n);
if (n == 1) {
    printf("1\n");
    return 0;
}

```

```

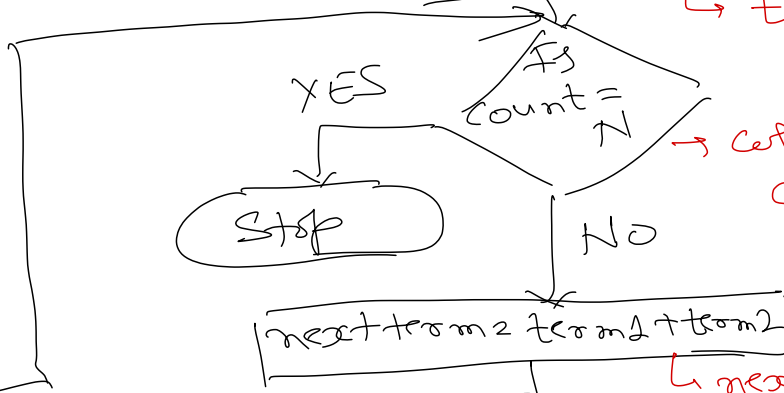
if (n == 2) {
    printf("1 1\n");
    return 0;
}

```

```

printf("1 1");
Count = 2;
term1 = 1;
term2 = 1;

```



```

while (Count < N)

```

```

int nextTerm;
nextTerm = term1 + term2;

```

