Palindrome

if Number EQUALS Reverse then

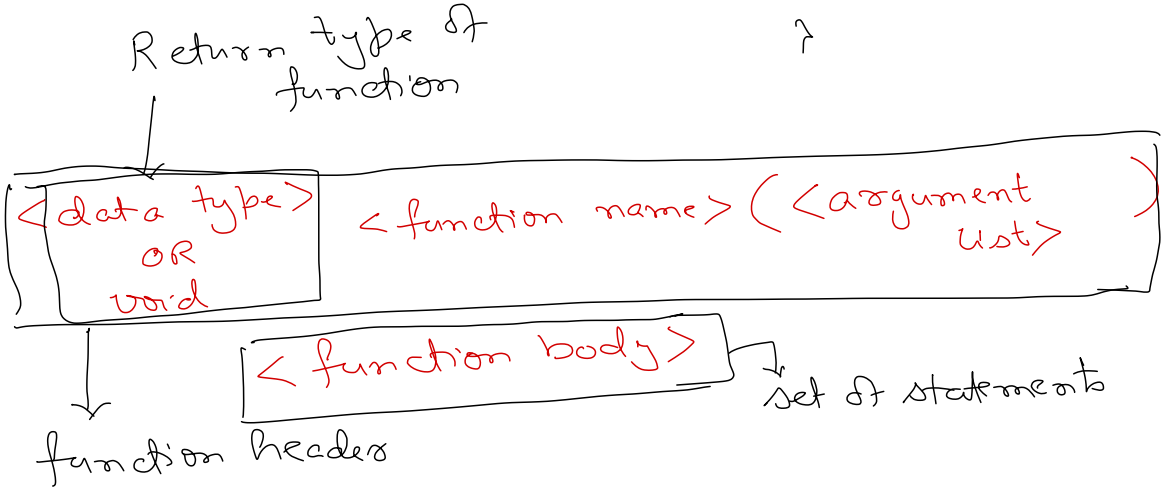
Its a Palindrome

else

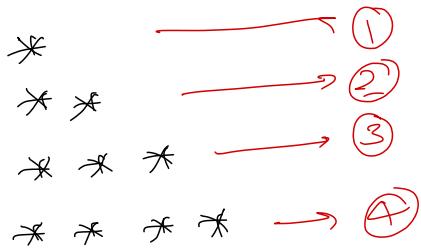
Its NOT Palindrome.

Function

```
int main()
{
   Read number
   Reverse number
   check
}
```



Printing Patterns



lineCount → 3
lineNo → 0 1 2 3
starsPrinted → 0 1 0 1 2 3 2 1 0

```
*
* *
* * *
```

① Get lineCount — how many lines to print. → scanf("%d", &lineCount);

② Set lineNo to 0. → lineNo = 0;

③ Repeat following steps while lineNo NOT EQUALS lineCount

```
while (lineNo != lineCount) {
```

④ Increment lineNo by 1.

```
lineNo = lineNo+1;
```

⑤ Set starsPrinted to 0.

```
starsPrinted = 0;
```

⑥ Repeat following steps while starsPrinted NOT EQUALS lineNo

```
while (starsPrinted != lineNo) {
```

⑦ Print "*"

```
printf("*");
```

⑧ Increment starsPrinted by 1.

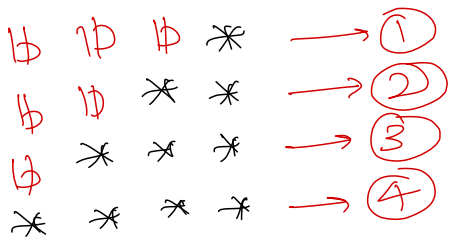
```
starsPrinted = starsPrinted+1;
}
```

⑨ Print newline. → printf("\n"); }

⑩ Stop.

---

b ⇒ blank space



3 spaces	1 star
2 spaces	2 stars
1 space	3 stars
0 space	4 stars

lineCount → 3

lineNo → 0 1 2 3

spacePrinted → 0 1 2 0 1 0

starsPrinted → 0 1 0 1 2 0 1 2 3

␣ ␣ *

␣ * *

* * *

(1) Get lineCount – How many lines to print.

(2) Set lineNo to 0.

(3) Repeat following steps while lineNo NOT EQUALS lineCount

(4) Increment lineNo by 1.

(4.1) Set spacePrinted to 0.

(4.2) Repeat following steps while spacePrinted NOT EQUALS (lineCount - lineNo)

(4.3) Print " "

(4.4) Increment spacePrinted by 1

(5) Set starsPrinted to 0.

(6) Repeat following steps while starsPrinted NOT EQUALS lineNo

(7) Print "*"

(8) Increment starsPrinted by 1.

spacePrinted = 0;

while (spacePrinted != (lineCount - lineNo))

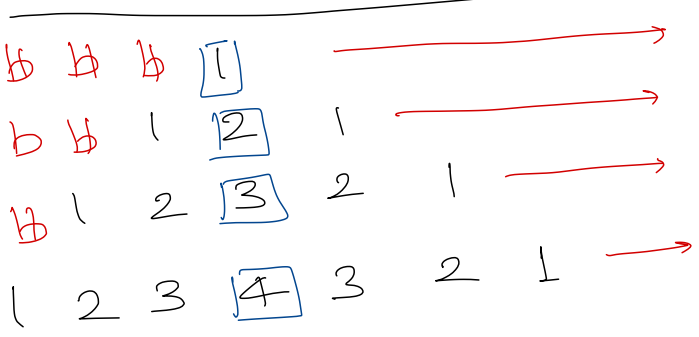
{ printf(" ");

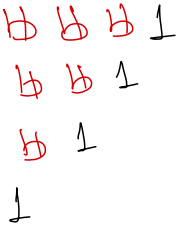
spacePrinted = spacePrinted + 1;

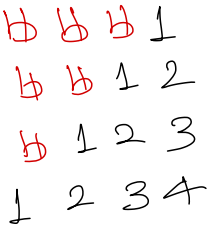
}

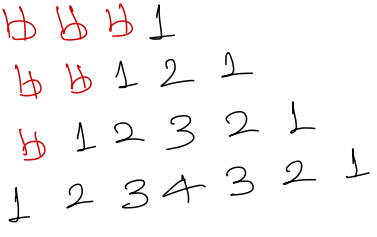
④ Print newline.

⑤ Step









Line No → 1 2 3 4

Corr No → 1 0 2 1 0 3 2 1 0

(1) Get lineCount – how many lines to print.

(2) Set lineNO to 0.

(3) Repeat following steps while lineNo NOT EQUALS lineCount by 1.

(4) Increment lineNO

(4.1) Set spacePrinted to 0.

(4.2) Repeat following steps while spacePrinted NOT EQUALS (lineCount – lineNo)

(4.3) Print " "

(4.4) Increment spacePrinted by 1

(5) Print "1"

(6) Set currNo to 1.

(7) Repeat following steps while (currNo NOT EQUALS lineNo)

(8) Increment currNo by 1

(9) Print currNo.

(10) Set currNo to lineNo - 1.

(11) Repeat following steps while (currNo > 0)

(12) Print currNo.

(13) Decrement currNo by 1.

(14) Print new line.

(15) Stop

Armstrong Number

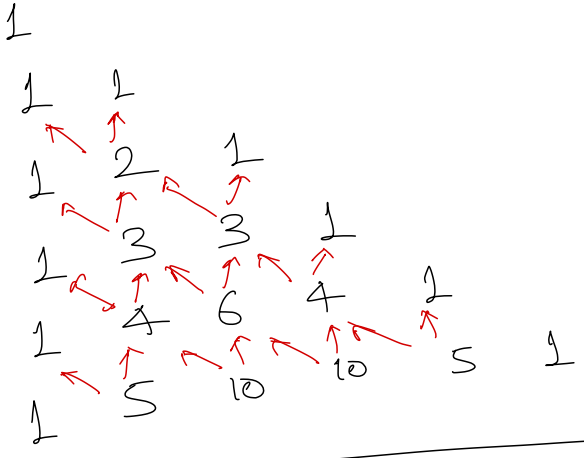
Number → 1234

digit Count → 4

$$1^4 + 2^4 + 3^4 + 4^4 = \text{Number}$$

x^y = Multiplying x y times.

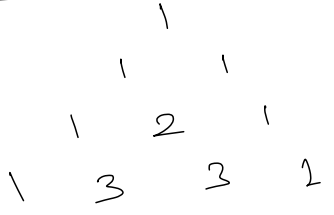
Pascal Triangle



n^{th} row

r^{th} column

${}^nC_r$



Print number digits in word

Number → 243

243 DIV 10 → Q: 24, R: 3

24 DIV 10 → Q: 2, R: 4

2 DIV 10 → Q: 0, R: 2

Three Four Two

Reverse → 342

(1) Get No.
(1.5) Find digit count of No.
(2) Reverse the no.
(3) Repeat following steps while (no > 0)
  (4) digit = remainder no DIV 10
  (5) if digit EQUALS 1 then
      Print "one"
  else if ...
      :
  (6) no = quotient no DIV 10.
  (7) Decrement digit count by 1.
(8) Print "zero" digit count times.
(9)

No → 100
Rev → 1

Array → Is a variable that can store multible values of same type.

(1) Get a number.
(2) Repeat following steps while (number > 0)
  (3) digit = remainder number DIV 10
  (4) Append digit to digits Array.
  (5) number = quotient number DIV 10
(6) Set curr value to last element of array.
(7) Repeat following while all array elements are not processed.
  (8) if curr value is 1 then ...
  (9) Set curr value to prev value of array
(10) Stop.

Number → ~~243~~ ~~24~~ ~~2~~ 0

Two Four Three

digits Array → |3|4|2|

digit → 3 4 2

243 DIV 10 → Q 24, R 3

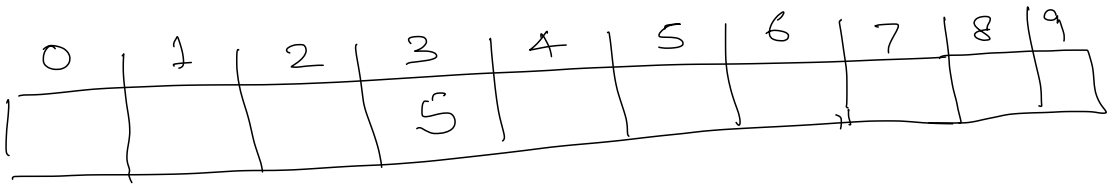
24 DIV 10 → Q:2, R:4

2 DIV 10 → Q:0, R:2

curr Value → ~~2~~ ~~4~~ 3

<data type> <arrayName> [<size>];

int digitsArray [10];



Operator [] ← subscript operator
to access array elements.

digitsArray [3] = 5;

C++

```
std::string s = "hello";

>  ==  s.length()
<  !=  s.append(1,'')

s[0] = 'c';
```