WAP to convert 'a'-'z' to 'A'-'z'.

I/P : {a, b, c, ... }

O/P : { A, B, C, D .... }

```
int main()
{
    char name;
    cin >> name;
    cout << convert(name);
}
```

```
char convert(char ch)
{
    char ans;
    ans = ch - 'a' + 'A';
    return ans;
}
```

(1)

```
ans = name - 'a' + 'A';
```

A → 65

a → 97

let

name = c

'c' - 'a' + 'A';

2 + 'A';

ans = C;

Armstrong no.

eg (1)

23

2 digit

no. of digit

$2^2 + 3^2 = 4 + 9 = 13$

23 == 13

23 is not Armstrong no.

if yes → armstrong no.

No - Not Armstrong no.

(11) 153

3 digit

$1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

153 == 153

↓

yes ∴ 153 is Armstrong No.

```
int countDigit(int num)
{
  int count = 0;
  while(num)
  {
    count++;
    n /= 10;
  }
  return count;
}

int main()
{
  int n;
  cin>>n;
  int digit=countDigit(n);
  cout<<Armstrong(n,digit);
}

bool Armstrong(int num, int digit)
{
    int n = num, ans = 0, rem;
  while(n)
  { rem = n%10;
    n /= 10;
  ans+=pow(rem, digit) // (rem^digit).
  }
  if(ans == num)
      return 1;
  else
      return 0;
}
```

Find Trailing zero in a fact (ufg)

			O/P
e.g	6! = 720	→	1
	8! = 40320	→	1

(i)

$6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6$

$4 = 2^2$, $6 = 2 \times 3$

$2 \times 3 \times 2^2 + 5 \times 2 \times 3$

$2^4 \times 3^2 \times 5$

$5 \times 2 = 10$

(ii) $2^4 \times 3^2 \times 5^2$

$2^2 \times 5^2 = 100$

$10 = 5 \times 2$

no. of 2's and 5's

(iii) $2^4 \times 5^3 = 10^3$

↳ 000

No. of 5 if n!.

$2^n \times 3^y \times 5^z$

$n > y > z$ →

$1 \times 2 \times 3 \times 4 \times 5 \times \ldots \times 10 \times \ldots \times 15 \times \ldots \times 20$

$5/5 = 1$

$15/5 = 3$

$\times \ldots 25 \ldots$ 50

$25 = 5 \times 5$ (2)

$50 = 5 \times 5 \times 2$

$\frac{25}{5} = 5/5 = 1$

$5 + 1 = 6$.

$\frac{50}{5} = \frac{10}{5} = 2$

$10 + 2 = 12$.

$100! \rightarrow \frac{100}{5} = \frac{20}{5} : 4$

$20 + 4 = (24)$

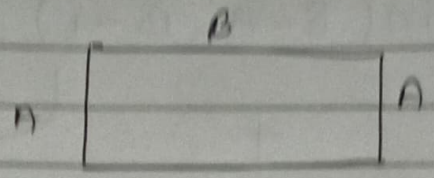
$\underline{148!} \Rightarrow \frac{148}{5} = \frac{29}{5} + \frac{5}{5} = 1$

$29 + 5 + 1 = (35)$.

```
int trailingzero(int n)
{
    int count=0;
    while(n>=5)
    {
        count += n/5;
        n /=5;
    }
    return count;
}
```

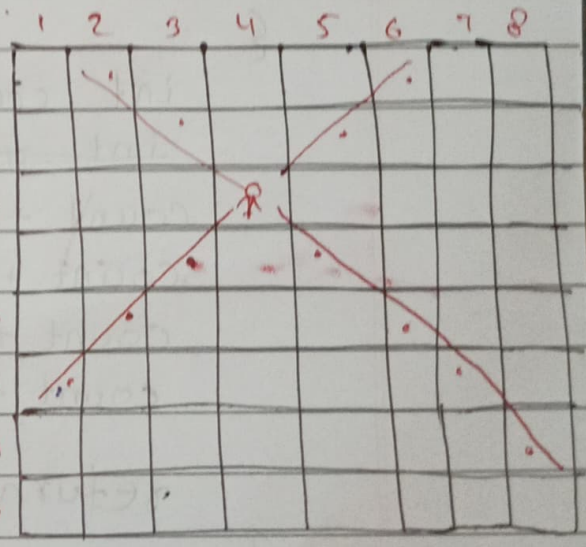
Rectangle (InterviewBit)

a, b, c, d




```
bool Reactangle (int a, int b, int c, int d)
{
    if ((a == b) && (c == d) ||
        (a == c) && (b == d) ||
        (a == d) && (b == c) )

        return 1;

    else

        return 0;

}
```

Bishop (InterviewBit)



A B

3 4 (11)

Total : 8 4

(R, D) 8 8

Right, Down 3 4

(5, 4)

min (4)

8, 1

3 4

(5, 3) → (3) (Left, Down)

min(8-A, 8-B)
↳ Right, Down.
+
min(8-A, B-1)
↳ (Left, Down)
+
min(A-1, B-1)
↳ (left, UP)
+
min(A-1, 8-B)
↳ (Right, UP)

Total = min(8-A, 8-B) + min(8-A, B-1)
+ min(A-1, B-1) +
min(A-1, 8-B).

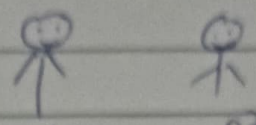
```
int Bishop( int A, int B)
{
    int count = 0
    int ans = max(2,3
    count += min(8-A, 8-B);
    count += min(8-A, B-1);
    count += min(A-1, B-1);
    count += min(A-1, 8-B);

    return count;
}
```

Nim Game (leetcode)

We can take 1 to 3 step.

1 2 3

n



1st 2nd

1 2 3

4 5 6 3, 4

1st make always factor of 4, I will always win, but if no. is already in 4 factor then I will loose.

n = 4, 8, 12, 16, 20

↳ loose.

```
if (n % 4 != 0)
    return 1; // win
else
    return 0; // loose.
```