```
↳
usr_str=input("   ")
user_conf=input(" yes|no")

if user_conf=='yes':
        pt(usr_str.lower())
```

Cond -1

```
↳ import os
  t_w=os.get_terminal_size().columns
  given_str= input("   ")
  user_conf=input(" yes|no")
  if user_conf =="yes":
        pt(given_str.center(t_w).title()
                  .ljust(t_w).
                  .rjust(t_w).

↳ my_even_no=[2,4,6,8,10,12]
  user_num=eval(input("  "))
  if user_num in my_even_no:
        pt("The no is even in list")
  else:
        pt("The no is not even in list")

↳ a=eval(input("  "))
  b=eval(input("  "))
  if a>b:
        pt(f'{a} > {b}')
  elif a<b:
        pt(f'{b} is > {a}')
  else:
        pt(f'{a} is equal to {b}')
```

```
↳ usr_str= input("          ")
  usel_conf= input("yes/no")

  if usel_conf == "yes":
          pt(usr_str.title())

  else:
      pt(usr_str)
```

↳ notowards.py

```
num =eval(input("    "))
if num not in [          ]:
      pt("    ")

else:
      if num==1
```

```
↳ num_word = {1:"one", 9:"nine", 10:"ten"}
      if num in num_word:
            pt(num_word[num])

      else:   if num ==1:
                  pt
```

ABdUl

Cond-2

Losical

a=5
b=7
c=3

↳ a>b and a>c
False

↳ a>b or a>c
True

a=8
a>0 and a<10
True

a=8
not a → False

↳ Examples

```
a. num=eval(input("  "))
   num%2==0:
   even

b. age>=18:
      pt("  ")
   else: pt("  ")

c. age >=18 and age<=60:
   elisible work
```

```
try:
    age=int(input("  "
except ValueErroe:
    pt("Invalid")
else:
    if age>=18 and
       age <=
        pt
    else:
        pt
finally:
    pt(
```

Marks

```
marks = int(input("   ")
if marks >=0 and marks <=100:
            pt("valid")
else:
      pt("Invalid")
```

Gender

```
gender= input('M|F')
if gender=='m' or gender=='M':
          pt('Male')
elif: gender=='f' or gender=='F':
          pt('Female')
else:
      pt("  ")
```

Vowel

```
ch= input("")
if ch== 'a' or                ....:
        pt("vowel")
else:
      pt("Consonant")
```

Marks:-

```
maths= int(input(" "))
science=
english
if maths >=45 and science >=45 and english>=
                  pt("pass")
else: pt("fail")
```

Cond-3

```
a=10
if a>=0:
    pt("a non -ve")
else: pt('a is -ve')
```

```
↳ if temp ==25:
      pt('normal')
  ~~else~~:
  elif temp >25:
      pt(" high")
  else:
      pt (temp high")
```

```
if temp ==25:
    pt(" normal")
else:
    if temp >25:
        pt ("high")
    else:
        pt ("low")
```

↳ discount Bill

```
if amt <1000:
    total =amt-(amont *0.1)
elif amt >=1000 and amt <5000:
    total=amt- (amt *0.15)
elif amt >=5000 and amt <10000:
    total= amt - (amt * 0.2)
else:
    total= amt - (amt* 0.25)
pt(f'Total: {total}')
```

Cond-3

```
a=10
if a>=0:
    pt("a non -ve")
else: pt('a is -ve')
```

```
↳ if temp ==25:
      pt('normal')
  ~~else~~:
  elif temp >25:
      pt(" high")
  else:
      pt(temp high")
```

```
if temp ==25:
    pt(" normal")
else:
    if temp >25:
        pt("high")
    else:
        pt("low")
```

↳ discount ~~x~~ Bill

```
if amt <1000:
    total =amt-(amont *0.1)
elif amt >=1000 and amt<5000:
    total=amt-(amt *0.15)
elif amt >=5000 and amt <10000:
    total= amt-(amt * 0.2)
else:
    total= amt-(amt*0.25)
pt(f'Total: {total}')
```

```
day= eval(input("  "))
if day ==1:
    pt("Monday)
    !
    !
else:
    pt("Invalid")
```

Calendar Month

```
Month = eval(input(" "))
if Month ==1:
        pt("january")

  !
  !
else:   pt("Invalid month")
```

Disit

```
if digit ==0:
    pt("zero")

  !
else:
    pt("out of 10")
```

Cond-4

Leap:-

```
year = eval(input(" "))
if year % 4 == 0 and (y % 100 != 0 or
                      y % 400 == 0):
    pt("Leap")
else:
    pt("not a leap")
```

String comp:-

```
s1 = "software" }   s -> 115
s2 = "Hardware" }   H -> 72

    s1 > s2 } True

-> s1 = "python"  }  s1 < s2   False
   s2 = "pycharm" }

-> s1 = "inteser" }  s1 == s2
   s2 = "inteser" }  True

-> s1 = "printer" }  s1 > s2
   s2 = "print"   }  True
```

↳ Short_ckt
X

5 and 10

0101
1010

T and T
10 ✓

5 and 0
T and F
0 ✓

0 and 10
↳ 0

5 or 10
5 ✓

0 or 10
10 ✓

↳ a=10
b=5
c=3

a>b and b>c
True

a<b and b>c
False

a>b or b>c
True

a<b or b<c
False

Bitwise

↳ a=25
bin(a)
0b11001

a=25
a.bit_length()
5

1011 -11
1100 -12
1101

↳ 10 & 13

```
 1010
 1101
 ----
 1000   } 8
```

```
 1010   or
 1101
 ----
 1111  → 15
```

10^13

```
 1010
 1101
 ----
 0111   } 7
```