

## control statements :-

It is a phenomenon of controlling the flow of execution. Based on Decision handling the set of instructions. Based on Decision the program will execute top to bottom in certain point or certain instruction we have to decide which statement block is execute or Not. Based on the decisions.

The control statements are classified into 3 types

- (1) Conditional Statement (Decision Statement)
- (2) Looping statements (Iteration Statement)
- (3) Termination Statement

### Conditional Statement

It is a phenomenon of controlling the flow of execution. Based on the Decision if Decision is satisfied that particular associate block of code is execute.

→ Conditional statements are classified into five types.

- (1) If statement (simple if)
- (2) If else statement
- (3) If elif statement (If elif else statement)
- (4) If ladder (Nested If)
- (5) Match case

### (1) If statement

→ It is one of the Decisional Statement.

→ This statement will consider only one decision or condition, if "decision is satisfied" then "True" statement block of code is execute.

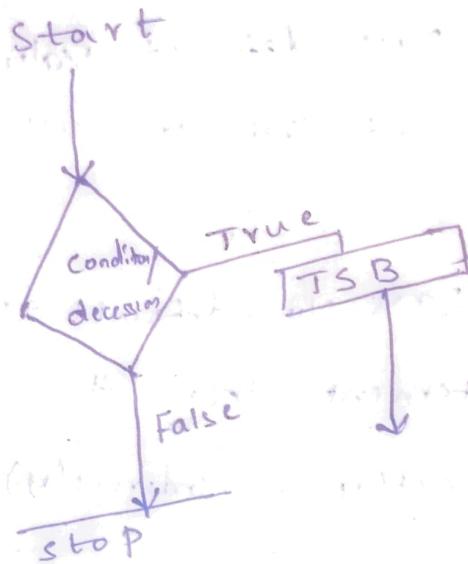
→ If we have only one condition, only one block of code is executes there.

One Block of code is executes there.  
we have to use If statement.

→ If condition is not satisfied, the control will move out of whole If statement.

if/else

## Flow chart



## Syntax

if : keyword

```
Print('start')
if condition / expression:
    ↪ tab: set of statements
    |
    |
    |
    |
    TSB
|
Print('stop')
```

Where as if is a keyword it will check condition of Decision satisfied or Not if Condition is satisfied the Control move on to True Statement Block

- if statement will consider only Boolean values either true or false.
- If we pass default value inside the if statement then it will consider false.
- If we pass Non-default value inside the if statement then it will consider True (Directly or Indirectly)

### Expression:

The combination of Literals or Datatypes with operators (unprocessed statement)

### Condition:

It is a resultant Data (processed statement) →

⋮

It indicates Intendation Block

### ltab:

It Consider 4 spaces starting of line

### TSB:

It is a set of statements when decision is True then all the statements are execute

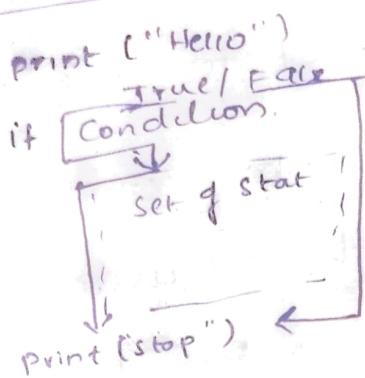
Ex:

```

① print ("Hello")
if true:
    print ("Haii")
print ("stop")

```

o/p → Hello  
Haii  
stop



② print ("Hello")
if false:
 print ("Bye")
print ("stop")

o/p → Hello  
stop

→ operators will perform the task it returns actual values or Boolean values  
→ if statement will take result from operators then if statement will decide it is True or False.

## If - Else statement

→ It is one of the Decisional Statement

→ whenever we have Only one condition, with two decision then we move on to if else statement

→ if - else

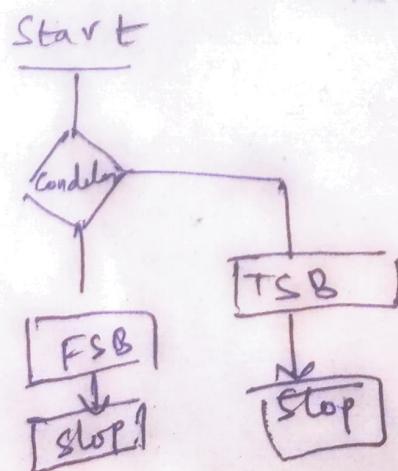
Initially the controller or if statement will check condition / decision is True (or) False

If decision is true, then true statement block Or code is execute

Or else, if condition is false,, then, false statement

Block of Code is Executed

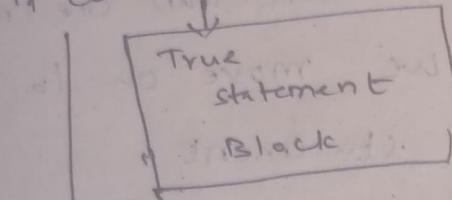
flow chart



## Syntax (on) Execution flow

→ `Print("start")`

if condition:



Print("stop")

### Else Block

It is a False Statement Block if Condition is False then the controller directly move on the else: Statement Block.

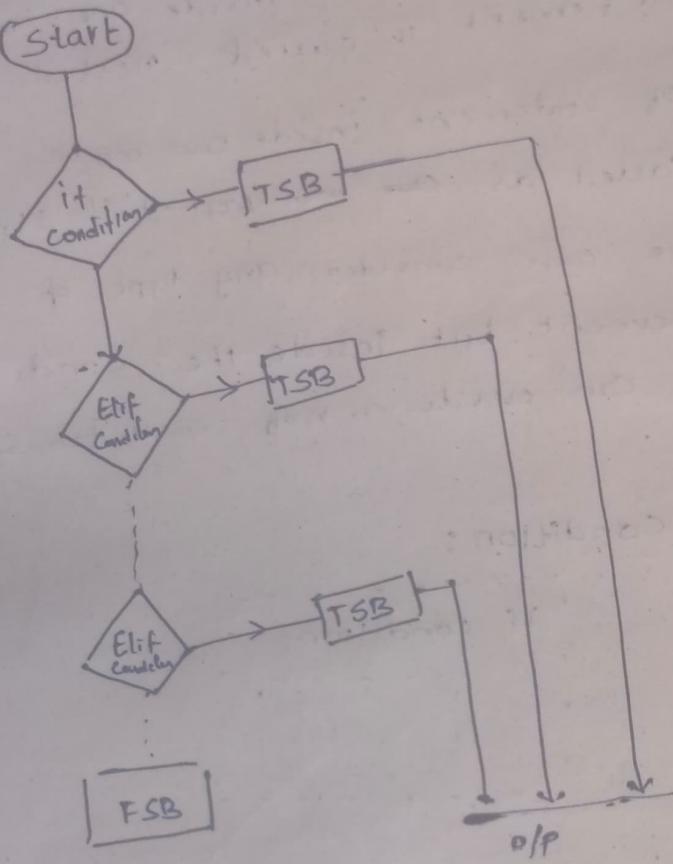
→ It is a optional Block

→ This Block we can use either

If Elif Else If Else:-

- It is the one of the Decisional Statement.
- whenever we have more than one condition or Decision then we move on to of Elif or If Elif Else statement,
- If Elif statement consists of minimum one condition, maximum n. no. of conditions and every condition have one particular Associate Block is available
  - any of the one condition is satisfied then that particular associate block of code is execute.
- Else Block is optional and If statement is mandatory Based on the no. of conditions we can create Elif Conditions

flowchart



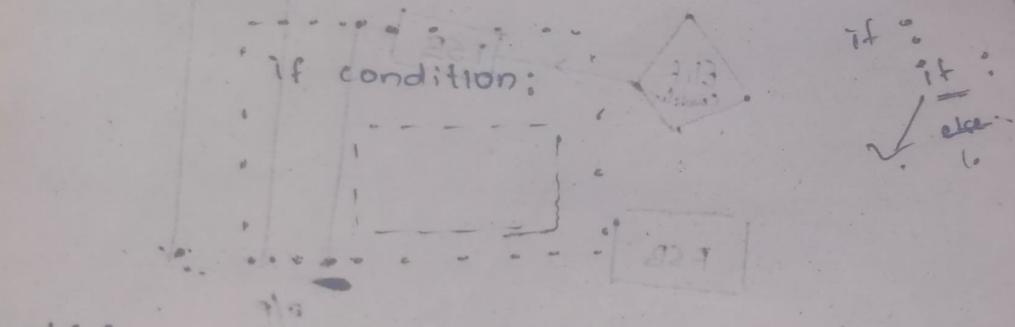
Syntax!

```
print("start")
if condition
  true: { TSB }
  elif condition
    false: { TSB }
  else:
    { FSB }
```

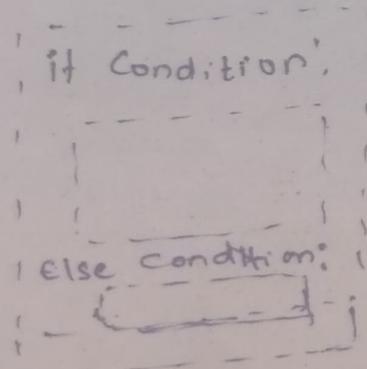
## Nested if :-

- It is one of the Decessional statement
- conditional statement inside one more Conditional Statement is called Nested Condition Statement
- If statement inside one more if statement is called as ~~one~~ nested if statement
- we can consider any type of Conditional Statement but inside the Conditional Statement we can write n. nof conditional Statement

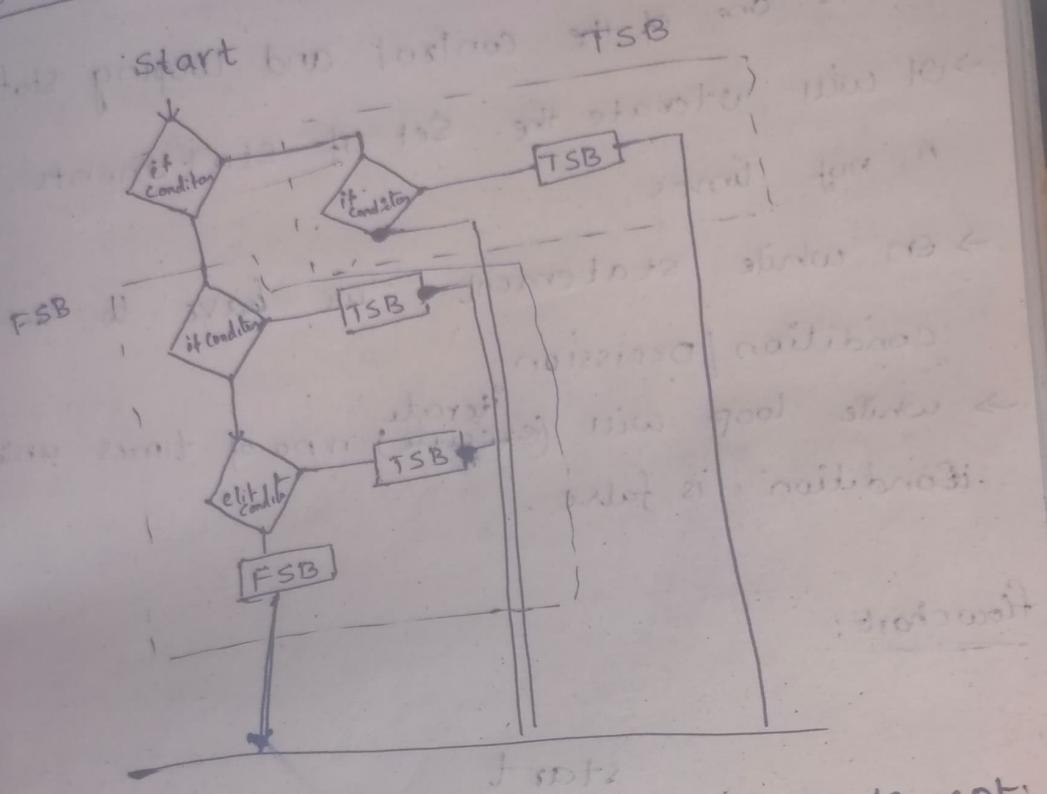
If Condition :



else :



flow chart :-



### Looping Statement (or) Iteration Statement:

- It is one of the control statement
- It is a phenomenon of repeating set of statements until the decision is false
- Looping statement is classified into two types:
  - (1) While Loop
  - (2) For Loop

Q. W.A.P. to check whether the given number is greater than 1 and less than 5, or not. If condition is satisfied, to display HelloWorld.

A1

$$a = 4$$

if ( $a > 1$  and  $a < 5$ ):

Print ("HelloWorld")

$1 < a < 5$

$a = 4$   
 if ( $a > 1$  and  $a < 5$ ):  
 Print ("HelloWorld")  
 $a = 4$   
~~if ( $a > 1$  and~~

$$a = 6$$

if ( $1 < a < 5$ ):

print ("HelloWorld")

Q. W.A.P. to check the given number is divisible by 7 or not, if divisible by 7 to perform multiplication with 2 and display the value.

A2

$$a = 21$$

if  $a \% 7 == 0$ :

Print ( $a * 2$ )

O/p - 42

$$a = 8$$

if  $a \% 7 == 0$

Print ( $a * 2$ )

O/p - (nothing empty)

$a = \text{eval}(\text{input}("Enter a number:"))$

if  $a \% 7 == 0$ :  $\rightarrow$  O/p → enter a number: 7

Print ( $a * 2$ )

$\rightarrow$  14

the gun number  
and 5, or not  
display HelloWorld

$a = 4$   
if  $(a > 1)$  and  
 $(a < 5)$ :  
Print (*i*, HelloWorld)  
 $a = 4$   
if  $(a > 1)$  and

~~(a) and~~

is desirable  
to perform,  
the value

$$= \hat{=} 0$$

number? 7

OP → cuts the value : 140  
£80

WAP to check whether a number is divisible by 2 and  
2 and number is divisible by 2 and satisfied to perform  
divisible by 6. If condition is satisfied skip place ip 2  
between eight shift width 8. 9-1.672

if  $a > 2$  and  $a^1 \cdot 2 = 0$  and  ~~$a^1 \cdot 6 = 0$~~

if  $a > 2$  and  $a \cdot 2 = 0$  and

~~at~~ at  $a \gg 2$

Penut (a)

O/p. → enter any value : 18

40

Q. WAP to check whether given number is even or not if even to perform square otherwise cube

$a = 8$ if $a \% 2 == 0$ : $a^* = a$ print(a)	$a = 8$ if $a \% 2 == 0$ : $a = a^{**} 2$ print(a)
--	---

$a = \text{eval}(\text{input}("enter the number"))$   
 if  $a \% 2 == 0$ :  
 $a^{**} = 2$   
 print(a)  
 O/P → enter the number: 6  
 36

Q. WAP to check whether the given number is odd or not if number is odd to swap & store it outside the swap

$a = 7$ if $a \% 2 != 0$ : $a = \text{swap}(a)$ print(a)	$a = \text{eval}(\text{input}("enter a value"))$ if <del>number</del> $a \% 2 != 0$ : $a = \text{swap}(a)$ print(a) O/P → enter a value: 7
---	--

number as  
input given

```
args ( )
a = eval(input("Enter a value:"))
st = ""
if num%2 == 0:
    st += str(a)
print(st, type(st))
```

O/P → enter a value : 7

WAP to check given value is -ve and  
that value should be more than -5 and  
less than -20. and numbers should be even. if  
all conditions are satisfied to extract  
only digits.

num = -2  
if num < 0 and num > -5  
and num < -20 and num%2 == 0  
num = -18  
if num < 0 and num > -5 and num < -20 and num%2 == 0  
num%2 == 0

num%2 == 18

print(num)

→ The statement does not satisfy us

### Imp

(1)  $a = [1, 2]$

$b = a$

$b += [3]$

Print(a)

[1, 2, 3]

(2)  $a = [1, 2]$

$b = [\cancel{1}, \cancel{2}] a$

$b = b + [300]$

Print(a)

[1, 2]

Q. WAP to check whether given value is -ve and it should not be more than -5 and less than -20, and number should be even if condition is satisfied to extract only digits

def num = -40

if num % 10 == 0 and num > -5 and num < -20 and num % 2 == 0:

    num \* = -1

    Print(num)

% → 40

Q) WAP to check given value is integer or not of given value is converted to perform. Given number is converted to string and extract +. Starts character and display it.

```
#include <iostream>
using namespace std;
int main()
{
    if (num == 0)
        cout << "Enter a number : ";
    else
        cout << "Enter a number : " << num;
}
```

Q) WAP to check whether given value is float or complex if condition satisfied to perform multiplication with 6 and display the value

```
#include <iostream>
using namespace std;
int main()
{
    float num;
    cout << "Enter a number : ";
    cin >> num;
    if (num == 0)
        cout << "Enter a number : ";
    else
        cout << "Enter a number : " << num * 6;
}
```

O/p → enter a number : 2

15.6

O/p → enter a number : 10  
100

Q. WAP to check whether given value is integer and value should be more than 15 and value should be odd. If condition is satisfied to convert to string and store it inside the list [ ]

Q1  
n = eval(input("enter a num:"))  
a = []  
if type(n) == int and n > 15 and n % 2 != 0:

n = str(n)

~~int~~[ ]

a += [n]

print(a)

Output → enter a num: 6

Output → enter a number: 19

Output → [19]

Q. n = eval(input("enter a number:"))

a = []

if type(n) == int and n > 15 and n % 2 != 0:

n = str(n)

a += n

print(a)

Output → enter a number: 10  
[1, 0]

Given value is  
be more than 15.  
if condition is  
met and store

and  $n \cdot 2 = 0$ .

Q) WAP to check if the given value is  
single value data type or not of single val  
to perform multiplication with 4.

a = eval(input("enter a number"))  
if type(a) == float or type(a) == complex  
or type(a) == int or type(a) == bool or  
type(a) == bytes or type(a) == None:  
 $a^* = 4$  : (14.0)

print(a)  
O/P → entered a number: 15

O/P → enter a number: 15  
15 \* 4 = 60

Q) Write a program to find out the sum of first 10 natural numbers.

a = eval(input("enter a number"))  
if type(a) in [int, str, float, complex, bool, bytes,  
None]:

$n = 4$  :  
if type(n) == float or type(n) == complex:  
 $n = int(n)$

print(n)

O/P → enter a number: 4

16 [16] is False

it is wrong a value < 4

16

Q. W.A.P. to check whether given value  
is Dictionary if it is Dictionary up to  
Extract all the values from the Dictionary

~~a = int(input("enter a Dictionary:"))~~

```
a = eval(input("enter a Dictionary:"))
if type(a) == dict:
    print(a.values())
else:
    print("not a dictionary")
```

O/P → enter a Dictionary : { 'a': 10, 'b': 9 }

\* (10, 9)

Q. W.A.P. to check whether Given key is  
Present in dictionary or not. If key is  
Present and extract the value.

~~a = eval(input("enter a Dictionary:"))~~

```
a = eval(input("enter a key:"))
b = { 'a': 10, 'b': 9, 'c': 10 }
if a in b.keys():
    print(b[a])
```

O/P → enter a key : b

9

Given value  
array to  
Dictionary  
say: {}

b, b: a, z

u  
u  
u vs. 2.

"")

Q) WAP to check given value is present  
in the dictionary or not & value is  
present extracts starting middle index value

```
a = eval(input("Enter a value:"))
d = {'a': 10, 'b': 20, 'c': 30, 'd': 40, 'e': 50, 'f': 60}
if 'a' in d.values():
    print([int(d.values())])
    e = [int(d.values())]
    print(e[0], e[-1])
```

O/p → enter a value: 30
[0 30 50]

Q) WAP to check length of the collection is  
even and length of collection should not zero

and value should be string & list & tuple  
If condition is satisfied to extract starts value to middle val

```
a = eval("Enter a collection:")
if (type(a) in [str, list, tuple]) and
    len(a) != 0 and len(a) % 2 == 0:
    print(a[0: len(a)//2 : 1])
```

O/p → enter a collection: "string"

stri

O/p → enter a collection: [10, 20, 30, 40]
[10 30 30]

Q. WAP to check given value is set, and, length  
the value is more than 5. If condition  
is satisfied, to perform swap the values  
first and last and display the collected  
list.

A: a = eval(input("enter a value"))

If type(a) == set and len(a) > 5 :

a = list(a)

a[0], a[-1] = a[-1], a[0]

print(a)

O/P → enter a value : {10, 20, 30, 40, 50, 60}

[30, 20, 40, 10, 60, 50]

Q. Write a program to append a new value to a list.

Ans: In this program, we will take a list and add a new value at the end of the list. We can do this by using the append() method.

Code (Please see the video):

```
l = [10, 20, 30, 40, 50]
l.append(60)
print(l)
```

Output: [10, 20, 30, 40, 50, 60]

Explanation: The append() method adds the new value at the end of the list.

Q. What is list comprehension?

Ans: List

Set, and length  
If condition  
map the values  
in the collection  
"))

o/p > 5

0]

0, 40, 50, 60 }

07

50

50

5

5

5

5

5

5

5

5

5

Q. WAP to check either given value is  
mutable or Not and if it is  
mutable, adding the value in middle of  
collection

```
a = eval(input("enter a value"))
if type(a) in [list, dict, set]:
    a = list(a)
    print(a[0 : len(a)/2] + value + a[len(a)/2 :])
else:
    print(a[0 : len(a)/2 : len(a)/2] - value)
    print(a[0 : len(a)/2 : len(a)/2] - value)
```

Q. WAP to check whether Given character is  
Upper Case or Not and display  
ASCII value of Given character

```
ch = eval(input("enter a character:"))
if ord(ch) "65" ≤ ord(ch) ≤ "90":
    print(ord(ch))
```

Print ord(ch)

O/p — enter a character: S

(or) 83

(or)

```
ch = eval(input("enter a character:"))
if type(ch) == str and len(ch) == 1 and 'A' <= ch <= 'Z':
    print(ord(ch))
```

Q. WAP to check whether given character is lower case or not. If it is lower case to display previous character of that character.

S.  $ch = b$

If  $97 \leq ch \leq 122$

~~print~~ ord(ch)

$a = \text{ord}(ch) - 1$

~~print~~ print(chr(a))

chr.eval(input("enter a character:"))  
if type(ch) == str and len(ch) == 1 and  $97 \leq ch \leq 122$

~~resp~~ (chr(ord(ch)-1))

~~print~~ print(res)

Q. WAP to check whether given character is

Alphabet or Not. If it is Alphabet, store the character inside the dictionary as ~~key~~ value and by as ASCII value.

S.  $ch = "a"$

If  $97 \leq ch \leq 122$  or  $65 \leq ch \leq 90$ :

$d = \text{ord}(ch)$

~~print~~ {ch: d}

whether given character  
is lower case  
character or not

Q. WAP to check whether given character  
is ASCII Number or Not, if ASCII Number  
to perform replication with 3

Ex: ch = '2'  
if '0' <= ch <= '9':

Print(ch \* 3)

ch = '2'

if type(ch) == str and len(ch) == 1 and '0' <= ch <= '9':  
print(ch \* 3)

Q. WAP to check whether given character is Special character or not?  
If it is special character, extract and display

Previous, given and Next character

Ex: ch = \*  
if ch not in

ch = \*

if not ('a' <= ch <= 'z' or 'A' <= ch <= 'Z' or  
'0' <= ch <= '9'):

Print(ch) -> Print(chr(ord(ch) + 1), ch, chr(ord(ch) + 1))  
Print(chr(ord(ch) - 1), ch, chr(ord(ch) + 1))

Q. WAP to check whether first and last characters are same or not in given string if both characters are same then both the characters are replaced to next characters

copy

Ch = "B A A A"

if ch[0] == ch[-1]:

chlist(ch)

ch[0] = chr(ord(ch[0]) + 1)

ch[-1] = chr(ord(ch[-1]) + 1)

print(chr(89 \* (ch[0]) + 1) + ch[1 : len(ch) - 2 : 1] + chr(89 \* (ch[-1]) + 1))

O/P → BAB

(or)

Ch = AMMA

if ch[0] == ch[-1]:

chlist(ch)

ch[0] = chr(ord(ch[0]) + 1)

ch[-1] = chr(ord(ch[-1]) + 1)

res = ''.join(s)

O/P → BMMA

and last  
in given  
same line,  
placed to

Q. W.A.P. to check second data item is more  
than last data item in given list of integers  
if condition is satisfied swap their values  
and sum of two values and append the  
result ~~start of the list~~ ~~at end of list~~

Ans  $L = [1, 4, 3, 2]$   
if  $L[0] > L[-1]$ :  
 $L[0], L[-1] = L[-1], L[0]$ .  
 $a = L[-1] + L[0]$   
 $a = \text{int}(a)$   
print ( $L + a$ )

Q. W.A.P. first and second characters are  
Sequence or not of sequence ~~the both~~  
~~characters are counted who and check if~~  
~~Saluted never the string.~~

Ans ~~or~~

$a = "ABCDE"$   
if ~~or~~  $\text{ord}(a[0]) == \text{ord}(a[1])$ :  
print ( $st[0:-1]$ )

Q. WAP to check whether given value is +ve, -ve or neutral. when it is +ve display (or) -ve (or) Neutral. when it is neutral to sum value as +ve (or) if it is neutral (or) display given value is neutral (or) if it is -ve display given value is

Negative

```
A
n = eval(input("enter a number:"))
if n <= 0 n > 0:
    print("given value is positive")
elif n == 0:
    print("given value is neutral")
else n < 0:
    print("given value is negative")
```

O/p → enter a number: 5

given value is positive

Enter a number:-5

given value is negative

Enter a number: 0

given value is neutral

value a +ve  
+ve display  
neutral to  
or  
cs

Q. WAP to check whether Given value is  
divisible by 4 to display Given value  
is divisible by 4 either Given value  
is divisible by 7 then to display Given  
value is divisible by 7 either Given value  
is divisible by 9 to display given value is  
divisible by 9

```
u
n = eval(input("enter a number:"))
if n%4 == 0: print("Given value is divisible by 4")
elif n%7 == 0: print("Given value is divisible by 7")
elif n%9 == 0: print("Given value is divisible by 9")
```

Q. o/p → enter a number: 36

Given value is divisible by 4

enter a number: 35

Given value is divisible by 7

enter a number: 81

Given value is divisible by 9

(a)

du

$$n = 252$$

if  $n \% 4 == 0$  and  $n \% 7 == 0$  and  $n \% 9 == 0$ :  
Print ("Given value is divisible by 4, 7 and 9")

else

elif  $n \% 4 == 0$  and  $n \% 9 == 0$ :  
Print ("Given value is divisible by 4 and 9")

elif  $n \% 4 == 0$  and  $n \% 7 == 0$ :

Print ("Given value is divisible by 4 and 7")

elif  $n \% 7 == 0$  and  $n \% 9 == 0$ :

Print ("Given value is divisible by 7 and 9")

elif  $n \% 4 == 0$ :

Print ("Given value is divisible by 4")

elif  $n \% 7 == 0$ :

Print ("Given value is divisible by 7")

elif  $n \% 9 == 0$ :

Print ("Given value is divisible by 9")

Q. W.A.P. to check Given value is even or odd or  
Neutral.

du

$n = eval(input("enter a number"))$

if  $n \% 2 == 0$ :

Print ("Given value is even")

elif  $n \% 2 != 0$ :

Print ("Given value is odd")

if  $n == 0$ :  
 print ("Give value in neutral")  
 o/p → enter a number: 9  
 Given number is odd.  
 WAP to check given no. is positive, even  
 or negative even or odd or negative odd.  
 Q: n = eval (input ("enter a number:"))  
 if  $n > 0$  and  $n \% 2 == 0$ :  
 print ("positive even")  
 if  $n < 0$  and  $n \% 2 == 0$ :  
 print ("negative even")  
 if  $n > 0$  and  $n \% 2 != 0$ :  
 print ("positive odd")  
 if  $n < 0$  and  $n \% 2 != 0$ :  
 print ("negative odd")

o/p → enter a number: 5

positive odd

enter a Number: -4

negative even

BP

Q. WAP to check whether given value is  
more than 65 and less than 125  
and if it is even so to print the  
ASCII character value of Given value. Given  
value is less than 122 and more than  
97 to display the value stored in  
the Dictionary as a value and key as  
a character. Given value is in the range of  
48 to 57 to perform square of Given value

Q. WAP to  
check or  
swap the  
two tuple  
values  
extracted  
from a

Ques  

```
n = eval(input("Enter a number:"))
m = { }
if 65 <= n < 125 and n % 2 == 0:
    print(chr(n))
elif 97 <= n < 122:
    print(chr(n))
else:
    m[chr(n)] = n
if 48 <= n <= 57:
    print(n**2)
O/P -> Enter a Number: 98
{ 122 : 98 } { C : 99 }
```

value is  
to print the  
given  
than  
it inside  
key as  
range of  
value

Q. WAP to check given value is in the  
list or tuple or string. If it is list to  
swap the second and last elements - or  
if tuple append the new 2 values. If it starts  
one value in start and another one in end is  
values are 100, [10, 20]. If it is string to  
extract the negative even position characters  
on given strings

n = eval(input("enter the value"))

if type(n) == list:

n[1], n[-1] = n[-1], n[1]

print(n)

elif type(n) == tuple:

n = (100,) + n + ([10, 20],)

print(n)

elif type(n) == str:

n = n[-2::-2]

print(n)

O/P → enter value: (10, 30)

(10, 30, 100, [10, 20])

Q. W.A.P. to check given character is what type of character

```
ch = char(put("enter a character: "))

if ch >= 'A' & ch <= 'Z':
    print ("Upper case")
elif ch >= 'a' & ch <= 'z':
    print ("Lower case")
elif '0' >= ch & ch <= '9':
    print ("ASCII number")
else:
    print ("Special character")
```

O/P → enter a character : \*  
Special character

O/P → enter a character : A  
Upper Case

Upper Case

(02,01) control wings & a/b  
(03,01,00,00,01)

is what type  
r: ))

Program to check given character is upper case or lower case  
(@) speed steel number of given character  
in upper case to convert upper to lower case of given character  
given character is lower case to convert lower to upper.  
if given character is upper case then value is ASCII number to multiply with '2' then value is INT. of given character is special character to extract  
Before 5th power off the given character  
next 5th power of given character, given character

If  
n = chr (input ("Enter a character: "))

if  $A \leq n \leq Z :$

$$n = \text{chr}(\text{ord}(n) + 32)$$

print (n)

elif  $a \leq n \leq z :$

$$n = \text{chr}(\text{ord}(n) - 32)$$

print (n)

elif '0'  $\leq n \leq 9 :$

$$n = \text{int}(n)$$

print ( $n * 3$ )

else:

$$p = \text{chr}(\text{ord}(n) - 5)$$

$$q = \text{chr}(\text{ord}(n) + 5)$$

Print (p, q, n)

Q. WAP to check whether given number is Positive or it is negative. Check the given value is even or it is even to display given value is even or the odd. If the given value is Negative, to check given value is divisible by 3 or Not by using nested if statement.

A/Q

n = eval(input("Enter a number:"))

if n > 0:

    if n % 2 == 0:

        Print(n, "is even")

    else:

        Print(n, "is odd")

else n < 0:

    if n % 3 == 0:

        Print("n is divisible by 3")

else:

    Print("Given value is not divisible by 3")

O/P → enter a number: 4

4 is even

enter a number: -18

Given value is divisible by 3

Given number is  
the given value  
is displayed given  
in value by  
nested if statements

In WAP to check whether given two values  
are more than 45 or condition is satisfied to  
find out greater of two values, if two values are  
not more than 45 to check and the relation ship b/w  
two values

a,b = 46, 56  
if a > b:

print ("greatest value a, a")

else:  
print ("b is a greater value in a, b")

elst:  
a < b and b < 45:

it a < b:  
print ("b is greater value")

elst a > b:  
print ("b is a greater value")

elst a == b:  
print ("a and b are equal")

Q. WAP to find out the grade (on care of student  
score is 90. first class, of student score is 60  
Score is zero then student is failed)

~~if m == 90:~~  
~~print ("student got first class")~~  
if m <

```
m = eval(input("enter marks:"))
```

```
if m >= 0 and m <= 100:
```

```
    if m >= 35% and m > 50%:
```

```
        print ("Third class")
```

```
if 85
```

```
if 85 <= n < 100:
```

```
    print ("Grade A")
```

```
elif 75 <= n < 85:
```

```
    print ("Grade B")
```

```
elif 55 <= n < 75:
```

```
    print ("Grade C")
```

```
elif 0 <= n < 55
```

```
    print ("Grade fail")
```

```
else:
```

```
    print ("Malpractise!")
```

Q. WAP to check whether a given character is alphabet or not, if alphabet to check, the given alphabet character is uppercase or not, if uppercase, to check vowel are not, if vowel to display the next character of given character, if lowercase, to convert to uppercase and display the character, for else given character is not a alphabet, than to check given character is ascii number or special character, and display the result.

```

    ch > 'A' or ch < 'Z'
    if ac = ch <= 27 or ac = ch >= 37
        if ch in {A, E, I, O, U}
            print chr(ord(ch) + 1)
        else:
            print chr(ord(ch) - 32)
    else:
        if '0' <= ch <= '9':
            print ("given character is ASCII number")
        else:
            print ("given character is special character")

```

O/P → B(i-1, 10) is 40

Q. WAP to check given value is primitive or not,  
 if value is non primitive, to check it is mutable or not  
 It mutable, to check if a given value a set or not,  
 if set, replace the middle element into new element, or  
 direct extract all values from the set, or set to  
 extract the first and last elements of set.  
 set immutable, to converting it into mutable  
 and swap first and last positions

```

# l = [10, 20, 30, 40]           # l = (100, 0.5, 789)
# l[0] = 10, l[2] = 20            # l[0] = 100
# l = {10, 20, 30, 40}           # 0 == true
# l = "Hello"                   # l == b'Hello'
# l = (10, 50, 60, 70, 80, 90)   # l == 1500
if type(l) in [str, int, tuple, dict, set]:
    if type(l) in [set, list, dict]:
        if type(l) == list:
            print([len(l)] / 2)
        else type(l) == dict:
            print(list(l.values()))
        else type(l) == set:
            l = tuple(l)
            print([l[0], l[-1]])

```

Q. What to check  
should be more  
check given l  
and l is play  
first index  
and test  
display the

(100, 05, 7893)  
100  
true  
0101

Q) WAP to check whether given length of the string should be more than 40. If condition is odd is satisfied, then check given length of string is odd or else extract and display middle character. or else extract first index position to length of collection - 2 and then length + collection in divisible by 3 then display the substring with power of length of substring.

```
s = "abcde.fghijk"
if len(s) > 10 :
    if len(s) % 2 == 0 :
        print(s[len(s)//2])
    else :
        print(s[0 : len(s)-2])
        if len(s) % 3 == 0 :
            print(s[0 : len(s)* len(s)])
```

O/P → abcde.fghijk

Q) WAP to check whether a given input is dict & not of dict, to extract and find the first and last values are same or Not. If same, to perform the addition or concat based on values. If it is not same, to convert list and concat new values into the list.

Q) Given value is list, to extract and delete the middle value and append the extracted value into Substring of list

If given value is tuple, to delete the first, last, middle value, display the output.

If given value is single value, to check value is greater than 25 or not, if true contract the last digit and display it

```
def n = {'a': 'vinay', 'b': 'Rohit', 'c': 'vijay', 'd': 'vinay'}
if type(n) == dict:
    l = list(n.values())
    if :
        m = l[0]
        n = l[-1]
        if m == n or type(m) == type(n):
            print(res = m + l[0] + l[-1])
        else:
            print(res)
    else:
        res = list(d.items())
        res [len(res)//2 : len(res)//2] = ['d', 'so']
        print(res)
```

Print(res)

exit type(n) == List :

a = n[len(n)//2]

del n[len(n)//2]

d = [a] + n

print(d)

elif type(n) == tuple:

n = list(n)

del n[0]

del n[-1]

del n[len(n)//2]

print(tuple(n))

if type(in) == (int, float)?  
if n >= 5:  
    print(f'{n/10} to  
    n, type(int))

(1) - apple = ['iphone', 'ipod', 'watch', 'mac']  
- mobiles = ['ipplus', 'visor', 'oppo', 'iphone']  
- person = ['iphone', 'ipod', 'smartwatch']  
ip = 'iphone'  
op = ['apple', 'mobile', 'person']

ip = 'mac'  
op = ['apple']  
ip = 'ipod'  
op = ['apple', 'person']  
ip = 'iphone'

if ip in apple:  
    L = []  
    if ip in apple:

L += ['apple']

If ip in mobile:

L += ['mobile']

If ip in person:

L += ['person']

Q) WAP to Given user id and password is Available or not if user id and password both are available to display valid user id. of any of them. If not a valid user then display invalid user id and password.

Initially store user id and password in collection

Ans

USERID = { 'Robot': 12345, 'Swami': Robert@123,

Minay : Swami@123 }

or credentials [ ]

d = { 'sai': { 'sai', 'sai@123' } }

user = input ("enter username")

password = input ("enter password")

if user in d:

    res = d[user]

    if user == res[0] and password == res[1]:

        print ("login success")

    else:

        print ("Invalid Password")

else:

    print ("Invalid User id")

password is Available or  
are available or  
Not a variable  
word  
in collection  
constant  
variable { 123 }

Q. WAP to extract uppercase characters in  
Given string.

n = "APPLE"

for i in n:  
if 'A' <= i <= Z:  
print(i)

APPLE

if 'A' <= n[0] <= 'Z':  
print(n[0])

else:

while  
Q. WAP to print HelloWorld in 5 times

n = 5  
while n > 0:  
print("Hello World")  
n = n - 1

### Tracing point

[5]

while  $a$ : True  
    Print("Hello")

$$n = a - 1 \\ 5 - 1$$

$n = 4$

[3]

while  $n > 0$ : True  
    Print("Hello")

$$a = n - 1 \\ = 4 - 1 \\ = 3$$

$n = 3$

while  $n > 0$ : True  
    Print("Hello")

$$n = n - 1$$

$$= 3 - 1 \\ = 2$$

$n = 2$

while  $n > 0$ : True  
    Print("Hello")

$$n = n - 1$$

$$= 2 - 1 \\ = 1$$

$n = 1$   
while  $n > 0$ : True  
    Print("Hello")

$$n = n - 1$$

$$= 0$$

$n = 0$

while  $n > 0$ : False

Ex:  
 $a = 5$   
while  $a$ : True  
    Print("Hello")  
 $a = 1$   
else → Hello  
Hello  
Hello  
Hello  
Hello

Ex:  
 $a = 5$   
while

else →

Ex: a