

File Edit View H1 \equiv B I \Leftrightarrow A_b

1) What are the inputs ? ---> length and breadth

2) What are the outputs ? ---> area and perimeter

3) What is the area of rectangle ? ---> length * breadth

4) What is the perimeter of rectangle ? ---> 2 * (length + breadth)

...

(Home work)
Write a program to determine volume of a sphere

1) What is the input ? ---> radius I

2) What is the output ? ---> volume

3) What is the volume of sphere ? ---> $4 / 3 * \pi * r^3$

...

(Home work)
Write a program to determine simple interest and compound interest

1) What are the inputs ? ---> principle, time and rate of interest

2) What are the outputs ? ---> simple interest and compound interest

3) What is simple interest formula ? ---> $ptr / 100$

4) What is compound interest formula ? ---> $p * (1 + r / 100)^t - p$

...

(Home work)
Write a program to swap values of two objects using 3rd object

File Edit View H1 Σ B I Θ A_b

2) What are the outputs ? ---> Simple interest and compound interest

3) What is simple interest formula ? ---> $\text{ptr} / 100$

4) What is compound interest formula ? ---> $p * (1 + r / 100)^t - p$

...

(Home work)
Write a program to swap values of two objects using 3rd object

Let $x = 10$ and $y = 25$
What are the values of x and y after swap ? ---> $x = 25$ and $y = 10$

...

```
x = 10
y = 25
print("Before swap: x =", x, ",y =",y)
temp =x
x = y
y = temp
print("After swap: x =", x, ", y =",y)
```

...

(Home work)
Write a program to swap values of two objects without using 3rd object

Hint: One addition and two subtractions

```
x = 25
y = 10
print("Before swap: x =", x, ",y =",y)
x=x + y
y=x - y
x=x - y
print("After swap: x =", x, ", y =",y)
...
```

```
x = 25
y = 10
print("Before swap: x =", x, ",y =",y)
x=x + y
y=x - y
x=x - y
print("After swap: x =", x, ", y =",y)
...
...
(Home work)
Write a program to swap values of two objects without using 3rd object
Hint: One multiplication and two divisions
I
x = -200
y = 100
print("Before swap: x =", x, ",y =",y)
x=x*y
y=x//y
x=x//y
print("After swap: x =", x, ", y =",y)
...
```

30/7/25

"Write a program to print full pyramid

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

Input is in number of lines

```
lines = int(input("How many lines?: "))
```

```
for i in range(lines):
```

```
    space = lines - i - 1
```

```
    stars = 2 * i + 1
```

```
    print(' ' * space, '*' * stars)
```

Space = lines - i - 1 - adds leading spaces to center the stars.
stars = $2^*i + 1 \rightarrow$ forms the full pyramid shape with increasing
odd number of stars.

O/p: when input is 7:

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

4/8/25

① find outputs

```
for i in integers (1, 8):
```

```
    print(i)
```

```
    if i % 3 == 0:
```

```
        continue
```

```
    else:
```

```
        print('sec')
```

```
        print('Hello')
```

```
# End of loop
```

```
print('outside loop')
```

i=1 O/p 1 sec

i=2 O/p 2 sec

Hello

i=3 → i%3 == 0 is True O/p 3

i=4 → 4 sec

Hello

i=5 → 5 sec

Hello

i=6 → i%3 == 0 is True 6

i=7 → O/p = 7 sec

Hello

② Identify Error

```
if (): # Empty condition in if () → syntax error  
    print('Hud')  
continue # continue is used outside the loop  
print('sec') # it is allowed to use inside "for" "loop" "while";
```

③ # find outputs

```
for i in range (1, 8):
```

```
    print(i)
```

```
    if i%3 == 0
```

```
        continue
```

```
        break
```

```
    else:
```

```
        print('sec')
```

```
        print('Hello')
```

```
# End of the loop
```

```
print('outside loop')
```

i=1: print(i) → 1

1%3 != 0 → goes to else:

• print('sec') → sec

print('Hello') → Hello

i=2: 2

sec

Hello

i=3: 3

outside loop

(else) is being

quitted by i%3 != 0

(break) is being

3%3 == 0 → break

loop stops immediately

Statements & iteration both are skipped

(4) off

if(10, 20, 30): # invalid condition: this is Tuple, not a valid
print('Hud')
break # word outside the loop for & while should be
print('sec')

word along with break

(5) off find output

```
for i in range (1, 8):  
    print(i)  
    if i%3==0:  
        pass  
        print ('Hud')  
    else:  
        print ('sec')  
    print ('Hello')  
# End of the loop  
print ('outside loop')
```

i=1:

• print (i) → Hud sec

+ Hello

i=2:

sec

Hello

i=3:

Hud

Hello

i=4:

sec

Hello

i=5:

Hud

Hello

i=6:

Hud

Hello

i=7:

sec

Hello

i=8: exit() terminates here

The following set lines are not
executed (i=4 to 7):
print ('outside loop')

```
for i in range (1, 8):  
    print(i)  
    if i%3==0:  
        pass  
        print ('Hud')  
    else:  
        print ('sec')  
    print ('Hello')  
# End of the loop  
print ('outside loop')
```

⑦ # find outputs

for i in range(1, 8):

print(i)

if i % 3 == 0:

continue

else:

print('sec')

print('Hello')

else:

print('else suite')

End of the loop

print('outside loop')

$i=1$	$i=2$	$i=3$ (continue)
1 sec Hello	sec Hello	skips: print('sec') and print('Hello')

$i=4$	$i=5$	$i=6$?	$i=7$
sec Hello	sec Hello	skips Hello	sec Hello

else suite

outside the loop

⑧ find outside

for i in range(1, 8):

print(i)

if i % 3 == 0:

break

else:

print('sec')

print('Hello')

else:

print('else suite')

End of the loop

print('outside loop')

$i=1$	$i=2$
sec Hello	sec Hello

$i=3$ $3 \% 3 == 0 \rightarrow$ break \rightarrow loops

ends immediately

• else: block is skipped (because of break)

⑨ find outputs

for i in range(1, 8):

print(i)

if i == 8:

break

else:

$i=1$	$i=3$
sec Hello	sec Hello

$i=2$ sec - Hello

$i=4$ sec - Hello

$i=5$ sec - Hello

$i=6$ sec - Hello

```

print('sec')
print('Hello')
else:
    print('else suite')
End of the loop
print('outside loop')

```

i=5
sec
Hello

i=6 sec
i=7 sec
Hello Hello

else suite
outside loop

- ⑩ Write a program to search for an element in the list without using in operator and print 'found or Not found message' (Assume that there are no duplicates)

```

lst = eval(input("Enter any list : ")) [10, 20, 15, 12, 18]
x = eval(input("Enter the element to be searched : "))
found = False
for i in range(len(lst)):
    if lst[i] == x:
        print("found at index :", i)
        found = True
        break
if not found:
    print("Not found")

```

- ⑪ Write a program to search for an element in the list and print index of each element and also, number of times it is found [10, 20, 15, 12, 18, 15, 19, 14, 15, 14]

lst = [10, 20, 15, 12, 18, 15, 19, 14, 15, 14]

x = int(input("Enter the element to be searched :"))
count = 0

```

for i in range(len(list)):
    if list[i] == x:
        print(x, "is found at index", i)
        count += 1

if count > 0:
    print(x, "is found", count, "times")
else:
    print(x, "is not found")

```

(12) # Walrus operator (:=)

```

print(a := 25)      # It is valid assign expression. it assign 25 to a &
print(a=25)         # error. Not valid = equal not an expression
print(a)            # valid 25
print(a := 6 + 7)   # 13
print(a)            # 13
print((a := 6) + 7) # 13
print(a)            # 6
print((a = 6) + 7) # Error only := allowed inside the expression

```

(13) find

```

a = 0
if a := 0:
    print('Hyd')
else:
    print('sec')
if b := 0:
    print('Hyd')
else:
    print('sec', b)
if c := 0:
    print('Hyd')
else:
    print('sec')

```

a = 0 is true

Output : Hyd

b := 0 assign 0 to b, & 0 is false.

so the else block runs

O/p : sec:0

invalid = is not allowed in conditions its
an assignment, not a comparison

14 Write a program to determine average of inputs which are terminated with `ctrl+z` (without walrus operator)

Let input be 25, 10.8, True, `ctrl+z`,

`total = 0`

`Count = 0`

`print("Enter values (ctrl+z to stop):")`

`try:`

`while True:`

`x = input("Enter input (ctrl+z to stop) :")`

`value = eval(x)`

`if isinstance(value, (int, float, bool)):`

`total = total + value`

`count = count + 1`

`except:`

`pass`

`if count > 0:`

`avg = total / count`

`print("Average : ", avg)`

`else:`

`print("No valid inputs to compute average.")`

15 del

`a = 25`

assigns 25 to variable a

`print(a)`

25

`del a`

deletes the variable a from memory

`print(a)`

will raise an error because 'a' no longer exists

⑯ a = b = c = 25
 print(a, b, c) # o/p = 25 25 25
 del a
 print(b, c) # delete a, but not the value 25; b, c valid
 print(a) # a is deleted already
 del b
 print(c) # 25 is deleted
 print(b) # b is deleted
 del c
 print(c) # error as c is deleted
 # error because there is no c

⑰ can multiple objects be deleted with same del operator?

a, b, c = 25, 10.8, 'Hyd'
 print(a, b, c) # Assign values
 del a, b, c
 print(a) # Reference deleted by del
 print(b) # object deleted
 print(c) # Python virtual machine
 print() # both will be not executed

⑱ Lists

a = [10, 20, 15, 18] # list
 print(a) # [10, 20, 15, 18]
 del a[2] # Remove index[2] → list is [10, 20, 18]
 print(a) # o/p [10, 20, 18]
 del a # deletes the whole list object a
 print(a) # 'a' is not defined
 print(a[0]) # Not executed due to crash at previous line

19 #

```
a(10, 20, 15, 18) # assign values tuple  
print(a) # (10, 20, 15, 18)  
print(a[0]) # 10  
del a[2] # delete index 2 tuple object cannot be  
del a # will not be executed due to deleted  
print(a) # (10, 20, 15, 18)  
print(a[0]) # 10
```

5. tuple objects after deletion of index 2

values after [2] being deleted
10, 20, 15, 18
values after [0] being deleted
20, 15, 18
values after [1] being deleted
15, 18
values after [3] being deleted
18

values after [0] being deleted
20, 15, 18
values after [1] being deleted
15, 18
values after [2] being deleted
18
values after [3] being deleted
None

8/08/25 - Om Sri Ram

① Write a program to determine largest command line input

```
import sys
argu = sys.argv
if len(argu) == 1:
    print("pls send inputs")
    sys.exit()
inputs = argu[1:]
converted = []
all_number = True
all_string = True
for item in inputs:
    try:
        num = float(item)
        converted.append(num)
        all_string = False
    except ValueError:
        converted.append(item)
        all_number = False
if not all_numbers and not all_strings:
    print("Inputs cannot be number and string")
    sys.exit()
if all_numbers:
    largest = max(converted)
    print("largest command line input is:", largest)
else:
    largest = max(converted)
    print("largest command line input is:", repr(largest))
```

② Write a program to determine whether a command line input is even number or odd number;

```
import sys  
if len(sys.argv) != 2:  
    print("pls send an integer input")  
    sys.exit()  
input_val = sys.argv[1]  
try:  
    num = int(input_val)  
    if num % 2 == 0:  
        print("Even number")  
    else:  
        print('Odd number')  
except ValueError:  
    print("pls send an integer input")
```

③ Write a program to determine average of command line input

```
import sys  
args = sys.argv  
if len(args) == 1:  
    print("pls send number inputs")  
    sys.exit()  
a = []  
for val in args[1:]:  
    if val == 'True':  
        a.append(True)  
    elif val == 'False':  
        a.append(False)  
    else:  
        try:  
            num = float(val)  
            if num.is_integer():
```

```
a.append(int(num))
```

else:

```
a.append(num)
```

except ValueError:

```
    print("pls send number inputs")
```

④ Write a program, do sort command line inputs in ascending order and descending order

```
Import sys
```

```
argu = sys.argv[1:]
```

```
print("What is argu? --> ", sys.argv)
```

```
try:
```

```
a = [float(x) for x in argu]
```

```
print("What is list, b1?-->", a)
```

```
asc = sorted(a)
```

```
print("list sorted in ascending order-->", asc)
```

```
desc = sorted(a, reverse=True)
```

```
print("List sorted in descending order-->", desc)
```

```
except ValueError:
```

```
    print("pls don't send numbers and string inputs together")
```

⑤ find outs

```
print('green' in 'Hyd' is green city') # True
```

```
print('day' in 'Sankal dayal saima') # True
```

```
print('Green' in 'Hdu' is green city') # False 'F'
```

```
print('d is' in 'Hyd' is green city') # True
```

```
print('dis' in 'Hyd' is green city') # False
```

```
print('iniu' in 'Srinivas') # False
```

```
print('iniu' not in 'Srinivas') # True
```

⑥	0	1	2	3	4	5	6	7
	R	a	m	o		R	a	o
	-8	-7	-6	-5	-4	-3	-2	-1

$a = \text{'Rama Rao'}$

`print(a[0:7:2])` # from index 0 to 6 (step 2): 'R', 'm', 'a' \rightarrow Rma

`print(a[:7])` # from beginning to index 6: 'Rama Ra'

`print(a[2:4])` # from index 2 to 3: 'm', 'a' \rightarrow ma

`print(a[2:])` # from index 2 to end: 'ma Rao' \rightarrow maRao

`print(a[:4])` # Index 0 to 3: 'Rama' \rightarrow Rama

`print(a[::2])` # Every 2nd char 'R', 'm', ' ', 'a' \rightarrow ea

`print(a[-6:-1])` # from index -6 to -1: 'm', 'a', ' ', 'R', 'a' \rightarrow eaa

`print(a[-6:])` # from -6(m) to end: 'ma Rao' \rightarrow mara

`print(a[:-4:-1])` # Reverse from end to -5: index -1 to -4 \rightarrow o, a, R

`print(a[-3:-1])` # Index -3 to -2: 'R', 'a' \rightarrow Ra

`print(a[-3:])` # from -3 to end: 'Ra' \rightarrow Ra

`print(a[::])` # full string 'Rama Rao' \rightarrow Rama Rao

`print(a[::])` # full string 'Rama Rao' \rightarrow Rama Rao

`print(a[:::-1])` # full string reversed \rightarrow oaraR mara

`print(a[:::-2])` # Every second char from end: 'a', 'a', ' ', 'm', 'R' \rightarrow amR

`print(a[-2::-2])` # a[-2:-9:-2] \rightarrow a mR

`print(a[2:8])` # Index 2 to 7: 'ma Rao' \rightarrow maRao

`print(a[2:8:-1])` # empty string \rightarrow

`print(a[:::-6])` # from -1 to -6, reverse: 'a', 'a' \rightarrow oa

`print(a[2:-3])` # Index 2 to -4 'm', 'a' \rightarrow ma

`print(a[1:6:-2])` # Index 1, 3, 5: 'a', 'a', 'R' \rightarrow aar

`print(a[::5:-5])` # from -1 to -5, step -5: only a[-1] \rightarrow o

`print(a[2:-5])` # from index 2 to 2: empty

`print(a[2:-5:2])` # from index 2 to 2: empty

`print(a[::0:-1])` from -1 to 1, reverse 'a', 'a', 'R', " ", 'a', m → 'oRAM'

7) Write a program to concatenate two strings separated by space but swap first two char : str1, str2 to str2, str1

```
str1 = input ("Enter first string:")
```

```
str2 = input ("Enter second string:")
```

```
if len(str1)<2 or len(str2)<2:
```

```
    print("Input should be a min of 2-char string.")
```

```
else:
```

```
    new_str1 = str2[:2] + str1[2:]
```

```
    new_str2 = str1[:2] + str2[2:]
```

```
    result = new_str1 + " " + new_str2
```

```
    print("Result:", result)
```

8) Write a program to print first two and the last two characters of the string

print an empty string if string has less than four characters.

```
s = input ("Enter a string:")
```

```
if len(s)<4:
```

```
    print("")
```

```
else:
```

```
    result = s[:2] + s[-2:]
```

```
    print(result)
```

9) Write a program to print characters of the string in forward and reverse directions without slice

0 1 2 3 4

U A M S I

-5 -4 -3 -2 -1

O A15s . ©Akhi reddy

```
s = input("Enter a string: ")  
for i in range(len(s)):  
    print(f"Character at index {i}: {s[i]}")
```

reverse

```
for i in range(-1, -len(s)-1, -1):  
    print(f"Character at index {i}: {s[i]}")
```

- ⑩ Write a program to print characters at even and odd indexes without slice.

0 1 2 3 4 5 6 7
R a m a R a O
-8 -7 -6 -5 -4 -3 -2 -1

```
s = input("Enter any string: ")
```

even = ""

odd = ""

```
for i in range(len(s)):
```

if i%2 == 0:

even += s[i]

else:

odd += s[i]

```
print("String at even indexes : ", even)
```

```
print("String at odd indexes : ", odd)
```

- ⑪ Write a program to print characters at even

let input be A 4 B 3 C 2 \$ 5

Example: if input is A 4 B 3 C 2 \$ 5 then output should be A B C \$

```
s = input("Enter any string with alternate characters and digit: ")
```

if len(s)%2 != 0:

```
    print("String should have alternate character and digit")
```

else:

result = ""

for i in range (0, len(s), 2):

char = s[i]

digit = s[i+1]

if not digit.isdigit():

print("string should have alternate character and digit")

break

result += char * int(digit)

else:

print("Result : ", result)

⑫ Write a program to merge two strings to form a new string.

a = input ("Enter first string : ")

b = input ("Enter second string : ")

c = "

i = 0

while i < len(a) and i < len(b):

c += a[i] + b[i]

i += 1

c += a[i:] + b[i:]

print("Result : ", c)

⑬ Write a program to remove duplicate characters of the string without using set.

s = input ("Enter any string: ")

out = "

for ch in s:

if ch not in out:

out += ch

print ("Result : ", out)

(4) len() function

```

print(len('Hud'))      # 3
print(len('Rama Rao')) # 8
print(len('9247'))     # 4
print(len('t-$'))      # 4
print(len(""))          # 0
print(len('A2#'))      # 3
print(len(''))           # 1
print(len(3456))        # Error converts digits, convert to str
print(len('sec'.len())) # Syntax error, cannot be used here

```

(5) chr() function

```

print(chr(65))          # A (upper case A)
print(chr(90))          # Z (upper case Z)
print(chr(122))          # a (lower case a)
print(chr(48))          # 0 (digit zero)
print(chr(97))          # 9 (digit nine)
print(chr(57))          # $ (dollar sign)
print(chr(36))          # , (space)

```

(6) ord() function

```

print(ord('A'))          # 65
print(ord('Z'))          # 90
print(ord('0'))          # 48
print(ord('2'))          # 48
print(ord('0'))          # 48
print(ord('9'))          # 57
print(ord('$'))          # 36
print(ord(','))          # 32

```

(16) Let input be A4M3Z5D2

0	1	2	3	4	5	6	7
A	4	M	3	Z	5	D	2

```
s = input("Enter any string with alternate character and digit:")
if len(s)%2 != 0:
    print("pls enter string with alternate char and digit")
else:
    out = ""
    for i in range(0, len(s), 2):
        ch = s[i]
        num = s[i+1]
        if not num.isdigit():
            print("pls enter string with alternate char and digit")
            break
        out += ch + chr(ord(ch) + int(num))
    print("Result : ", out)
```