



''' Write a program to determine largest command line input

1. py prog2.py 10 20 30.8 7 40 35.6 What is the largest command line input ? ---> 40 What is argv ? ---> ['prog2.py', '10', '20', '30.8', '7', '40', '35.6'] What is list 'a' ? ---> [10, 20, 30.8, 7, 40, 35.6] How to determine largest element of list 'a' ? ---> max(a) i.e. 40 What is the result of max(argv[1:]) ? ---> '7' What is the issue with max(argv[1:]) ? ---> Largest string is obtained but not largest number
2. py prog2.py What is the output ? ---> Pls send inputs
3. py prog2.py 'Rama' 'Sita' 'Rajesh' 'Manohar' 'Vamsi' 'Amar' What is the largest command line input ? ---> 'Vamsi'
4. py prog2.py 25 'Ten' What is the output ? ---> Inputs can not be number and string
5. Hint1: Use for loop
6. Hint2: Use try and except '''

```
In [ ]: import sys
argv = sys.argv

if len(argv) == 1:
    print("Please send inputs")
    sys.exit()

inputs = argv[1:]

all_numbers = True
all_strings = True
a = []

for item in inputs:
    try:
        num = float(item)
        a.append(num)
        all_strings = False
    except ValueError:
        a.append(item)
        all_numbers = False
if not (all_numbers or all_strings):
    print("Inputs cannot be numbers and strings together")
    sys.exit()

largest = max(a)
print("argv =", argv)
print("List a =", a)
print("Largest input =", largest)
```

''' Write a program to determine command line input is even number or odd number

1. py prog3.py 26 What is the output ? ---> Even number
2. py prog3.py 45 What is the output ? ---> Odd number
3. py prog3.py What is the output ? ---> Pls send an integer input
4. py prog3.py 10.8 What is the output ? ---> Pls send an integer input
5. py prog3.py Ten What is the output ? ---> Pls send an integer input '''

```
In [ ]: if len(sys.argv) != 2:
        print("Pls send an integer input")
    else:
        try:
            num = int(sys.argv[1]) # Convert to integer
            if num % 2 == 0:
                print("Even number")
            else:
                print("Odd number")
        except ValueError:
            # This is raised when argument cannot be converted to int
            print("Pls send an integer input")
```

''' Write a program to determine average of command line inputs

1. py prog4.py 10.8 25 True 14.6 19 False 7.4 What is argv ? ---  
>['prog4.py', '10.8', '25', 'True', '14.6', '19', 'False', '7.4'] What is list 'a' ? ---> [10.8, 25, True, 14.6, 19, False, 7.4] How to determine sum of list elements ? ---> sum(a) How to determine number of list elements ? ---> len(a)
2. py prog4.py What is the output ? ---> Pls send number inputs
3. py prog4.py 25 'Ten' What is the output ? ---> Pls send number inputs '''

```
In [ ]: import sys

if len(sys.argv) == 1:
    print("Pls send number inputs")
else:
    a = []
    for arg in sys.argv[1:]:
        if arg == 'True':
            a.append(True)
        elif arg == 'False':
            a.append(False)
```

```

else:
    try:
        num = float(arg)
        a.append(num)
    except ValueError:
        print("Pls send number inputs")
        break
else:
    print(sum(a) / len(a))

```

''' Write a program to determine average of command line inputs

1. py prog4.py 10.8 25 True 14.6 19 False 7.4 What is argv ? ---> ['prog4.py', '10.8', '25', 'True', '14.6', '19', 'False', '7.4'] What is list 'a' ? ---> [10.8, 25, True, 14.6, 19, False, 7.4] How to determine sum of list elements ? ---> sum(a) How to determine number of list elements ? ---> len(a)
2. py prog4.py What is the output ? ---> Pls send number inputs
3. py prog4.py 25 'Ten' What is the output ? ---> Pls send number inputs '''

In [ ]: **import** sys

```

if len(sys.argv) == 1:
    print("Pls send number inputs")
else:
    a = []
    for arg in sys.argv[1:]:
        if arg == 'True':
            a.append(True)      # True acts as 1 in sum
        elif arg == 'False':
            a.append(False)    # False acts as 0 in sum
        else:
            try:
                a.append(float(arg))    # Accept both ints and floats
            except ValueError:
                print("Pls send number inputs")
                break
    else:
        print(sum(a) / len(a))

```

''' Write a program to sort command line inputs in ascending order and descending order

1. py prog5.py 10 20 15.8 5 12.6 What is argv ? ---> ['prog5.py', '10', '20', '15.8', '5', '12.6'] What is list 'a' ? ---> [10, 20, 15.8, 5, 12.6] How to sort list 'a' ? ---> sorted(a) How to sort list 'a' in descending order ?

---> sorted(a , reverse = True)

2. py prog5.py 25 'Ten' What is the output ? ---> Pls don't send number and string inputs together ''

```
In [ ]: import sys

args = sys.argv[1:]

a = []
try:
    for arg in args:
        a.append(float(arg))
except ValueError:
    print("Pls don't send number and string inputs together")
else:
    print("Ascending order :", sorted(a))
    print("Descending order :", sorted(a, reverse=True))
```

## Find outputs (Home work)

print('green' in 'Hyd is green city') print('day' in 'Sankar dayal sarma') print('Green' in 'Hyd is green city') print('d is' in 'Hyd is green city') print('dis' in 'Hyd is green city') print('iniv' in 'Srinivas') print('iniv' not in 'Srinivas')

```
In [ ]: # Find outputs (Home work)
print('green' in 'Hyd is green city') # True
print('day' in 'Sankar dayal sarma') # True
print('Green' in 'Hyd is green city') # False
print('d is' in 'Hyd is green city') # True
print('dis' in 'Hyd is green city') # False
print('iniv' in 'Srinivas') # True
print('iniv' not in 'Srinivas') # False
```

''' (Home work) Slice demo program 0 1 2 3 4 5 6 7 R a m a R a o -8 -7 -6 -5 -4 -3 -2 -1 ''' a = 'Rama Rao' print(a [ : 7 : 2]) print(a [ : 7]) print(a [2 : 4]) print(a [2 : ]) print(a [ : 4 ]) print(a [ : : 2]) print(a [-6 : -1]) print(a [-6 : ]) print(a [ : -4 : -1]) print(a [-3 : -1]) # a[-3 : -1 : 1] ---> string from indexes -3 to -2 in steps of 1 i.e. Ra print(a [-3 : ]) print(a [ : : ]) print(a [ : ]) print(a [ : : -1]) print(a [ : : -2]) print(a [-2 : : -2]) # a[-2 : -9 : -2] ---> string from indexes -2 to -8 in steps of -2 i.e. amR print(a [2 : 8]) print(a [2 : 8 : -1]) print(a [ : -6 : -1]) print(a [2 : -3]) print(a [1 : 6 : 2]) print(a [ : -5 : -5]) print(a [2 : -5]) print(a [2 : -5 : 2]) print(a [ : 0 : -1]) print(a [-5 : 0 : -2])

```
In [ ]: ''' (Home work)
Slice demo program
```

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

```

a = 'Rama Rao'
print(a [ : 7 : 2]) # a[ 0 : 7 : 2] ---> string from indexes 0 to 6 in s
print(a [ : 7]) # a[ 0 : 7 : 1] ---> string from indexes 0 to 6 in s
print(a [2 : 4]) # a[ 2 : 4 : 1] ---> string from indexes 2 to 4 in s
print(a [2 : ]) # a[ 2 : 8 : 1] ---> string from indexes 2 to 7 in s
print(a [ : 4 ]) # a[ 0 : 4 : 1] ---> string from indexes 0 to 3 in s
print(a [ : : 2]) # a[ 0 : 8 : 2] ---> string from indexes 0 to 7 in s
print(a [-6 : -1]) # a[-6 : -1 : 1] ---> string from indexes -6 to -2
print(a [-6 : ]) # a[-6 : 8 : 1] ---> string from indexes -6 to 7 in s
print(a [ : -4 : -1]) # a[ -1 : -4 : -1] ---> string from indexes -1 to -4
print(a [-3 : -1]) # a[-3 : -1 : 1] ---> string from indexes -3 to -1
print(a [-3 : ]) # a[-3 : 8 : 1] ---> string from indexes -3 to 7 in s
print(a [ : : 1]) # a[ 0 : 8 : 1] ---> string from indexes 0 to 7 in s
print(a [ : ]) # a[ 0 : 8 : 1] ---> string from indexes 0 to 7 in s
print(a [ : : -1]) # a[-1 : -9 : -1] ---> string from indexes -1 to -8
print(a [ : : -2]) # a[-1 : -9 : -2] ---> string from indexes -1 to -8
print(a [-2 : : -2]) # a[-2 : -9 : -2] ---> string from indexes -2 to -8
print(a [2 : 8]) # a[ 2 : 8 : 1] ---> string from indexes 2 to 7 in s
print(a [2 : 8 : -1]) # a[ 2 : 8 : -1] ---> string from indexes 2 to 7 in s
print(a [ : -6 : -1]) # a[-1 : -6 : -1] ---> string from indexes -1 to -6
print(a [2 : -3]) # a[ 2 : -3 : 1] ---> string from indexes 2 to -4 in s
print(a [1 : 6 : 2]) # a[ 1 : 6 : 2] ---> string from indexes 1 to 5 in s
print(a [ : -5 : -5]) # a[-1 : -5 : -5] ---> string from indexes -1 to -5
print(a [2 : -5]) # a[ 2 : -5 : 1] ---> string from indexes 2 to -6 in s
print(a [2 : -5 : 2]) # a[2 : -5 : 2] ---> string from indexes 2 to -6 in s
print(a [ : 0 : -1]) # a[-1 : 0 : -1] ---> string from indexes -1 to 0
print(a [-5 : 0 : -2]) # a[-5 : 0 : -2] ---> string from indexes -5 to 0

```

''' Write a program to concatenate two strings separated by space but swap first two characters of the two strings. Assume that each string has a minimum of two characters

Let inputs be Java and Python What are the outputs ? ---> PyvaJathon

Hint: Use slice '''

```

In [ ]: string1 = input("Enter first string: ") # Java
        string2 = input("Enter second string: ") # Python

        new_str1 = string2[:2] + string1[2:]
        new_str2 = string1[:2] + string2[2:]

        output = new_str1 + " " + new_str2
        print(output)

```

''' 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

1. Let input be PYTHON What is the output ? ---> PYON
2. Let input be Hyd What is the output ? ---> Nothing ''

```
In [ ]: s = input("Enter a string: ")

if len(s) < 4:
    print("")
else:
    result = s[:2] + s[-2:]
    print(result)
```

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

				0	1
2	3	4			

Let input be V A M S I -5 -4 -3 -2 -1

What are the outputs ? ---> Character at index 0 : V Character at index 1 : A  
Character at index 2 : M Character at index 3 : S Character at index 4 : I

Character at index -1 : I  
Character at index -2 : S  
Character at index -3 : M  
Character at index -4 : A  
Character at index -5 : V

Hint: Use two for loops '''

```
In [ ]: s = input("Enter a string: ")

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

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

''' Write a program to print characters at even and odd indexes without slice

							0	1
2	3	4	5	6	7			

Let input be R a m a R a o

odd = '' + 'a' + 'a' + 'R' + 'o' = 'aaRo' even = '' + 'R' + 'm' + '' + 'a' = 'Rm a'

1. What action to be made when index is even ? ---> Concatenate the character to even object
2. What action to be made when index is odd ? ---> Concatenate the character to odd object
3. Hint: Use single for loop ''

```
In [ ]: s = input("Enter a string: ")
even = ''
odd = ''
for i in range(len(s)):
    if i % 2 == 0:
        even += s[i]
    else:
        odd += s[i]
print("Even index characters:", even)
print("Odd index characters :", odd)
```

''' Let input be A 4 B 3 C 2 \$ 5 0 1 2 3 4 5 6 7

What is the output ? ---> AAAABBBCC\$\$\$\$\$

1. What is the result of 'A' \* 4 ? ---> 'AAAA'
2. i a[i] a[i + 1] out

''

0 'A' '4' '' + 'A' \* 4 = 'AAAA'

2 'B' '3' 'AAAA' + 'B' \* 3 = 'AAAA' + 'BBB' = 'AAAABBB'

4 'C' '2' 'AAAABBB' + 'C' \* 2 = 'AAAABBB' + 'CC' = 'AAAABBBCC'

6 '\$' '5' 'AAAABBBCC' + '\$' \* 5 =  
 'AAAABBBCC' + '\$\$\$\$\$' =  
 'AAAABBBCC\$\$\$\$\$'

What is the difference between `a[i]` and `a[i + 1]` ? ---> `a[i]` is ith char of string and `a[i + 1]` is (i + 1)th char of string ''

```
In [ ]: a = input("Enter sequence: ").split()

out = ""

# process in steps of 2: a[i] → char, a[i+1] → number
for i in range(0, len(a), 2):
    out += a[i] * int(a[i + 1])

print(out)
```

''' Write a program to merge two strings to form a new string

1. Let inputs be HYD and VAMSI What is the output ? ---> HVYADMSI

	0	1	2		
a --->	H	Y	D		
	0	1	2	3	4
b --->	V	A	M	S	I

i a[i] b[i] c

..

0 'H' 'V' '' + 'H' + 'V' = 'HV'

1 'Y' 'A' 'HV' + 'Y' + 'A' = 'HVYA'

2 'D' 'M' 'HVYA' + 'D' + 'M' = 'HVYADM'

Concatenate remaining characters of the other string to object 'c' What is the final result ? ---> 'HVYADMSI'

Hint: Use single while loop and slice ''

```
In [ ]: a = input("Enter first string: ")
b = input("Enter second string: ")
```



```

c = '' # merged result
i = 0 # index

# Merge character-by-character using a single while loop
while i < len(a) and i < len(b):
    c += a[i] + b[i]
    i += 1

# Append remaining characters from the longer string
c += a[i:] + b[i:]

print(c)

```

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

1. Let input be RAMA RAO What is the output ? ---> RAMO
2. out = '' + 'R' = 'R' + 'A' = 'RA' + 'M' = 'RAM' + ' ' = 'RAM ' + 'O' = 'RAM O'
3. What action to be made if the character is not in out object ? ---> Concatenate the character to out object
4. What action to be made if the character is already in out object ? ---> Ignore the character
5. Hint: Use not in operator '''

```

In [ ]: s = input("Enter a string: ")

out = ""

for ch in s:
    if ch not in out:
        out += ch

print(out)

```

## len() function demo program (Home work)

```

print(len('Hyd')) # 3 print(len('Rama Rao')) # 8 print(len('9247')) # 4 print(len('+ $-')) # 3
print(len('')) # 0 print(len(' ')) # 1 print(len('A2#')) # 3 print(len(str(3456))) # 4
print(len('Sec')) # 3 ''' What does len(str) do ? ---> Returns number of characters in the string '''

```

```
print(chr(65)) # Converts unicode 65 → 'A' print(chr(90)) # 90 → 'Z' print(chr(97))
# 97 → 'a' print(chr(122)) # 122 → 'z' print(chr(48)) # 48 → '0' print(chr(57)) # 57
→ '9' print(chr(36)) # 36 → '$' print(chr(32)) # 32 → ' ' (space)
```

## ord() function demo program

```
print(ord('A')) # Converts 'A' to unicode value 65 print(ord('Z')) print(ord('a'))
print(ord('z')) print(ord('0')) print(ord('9')) print(ord('$')) print(ord(' '))
```

### ''' ord() function

1. What does ord() function do ? ---> Converts character to unicode value
2. How many unicode values exist ? ---> 512
3. What is the range of unicode values ? ---> 0 to 511
4. What are the unicode values of 'A' - 'Z' ? ---> 65 to 90 What are the unicode values of 'a' - 'z' ? ---> 97 to 122 What are the unicode values of '0' - '9' ? ---> 48 to 57
5. What is another name of unicode ? ---> Extended Ascii

Note: chr() and ord() are quite opposite functions '''

```
In [ ]: print(ord('A')) # Converts 'A' to unicode value 65
print(ord('Z')) # Converts 'Z' to unicode value 90
print(ord('a')) # Converts 'a' to unicode value 97
print(ord('z')) # Converts 'z' to unicode value 122
print(ord('0')) # Converts '0' to unicode value 48
print(ord('9')) # Converts '9' to unicode value 57
print(ord('$')) # Converts '$' to unicode value 36
print(ord(' ')) # Converts ' ' to unicode value 32
```

''' Let input be A4M3Z5D2

What is the output ? ---> AEMPZ\_DF

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

i a[i] a[i + 1] out

..

0 'A' '4' '' + 'A' + chr(65 + 4) = '' + 'A' + 'E' = 'AE'

2 'M' '3' 'AE' + 'M' + chr(77 + 3) = 'AE' + 'M' + 'P' = 'AEMP'

4 'Z' '5' 'AEMP' + 'Z' + chr(90 + 5) = 'AEMP' + 'Z' + '' = 'AEMPZ'

6 'D' '2' 'AEMPZ\_' + 'D' + chr(68 + 2) =  
'AEMPZ\_' + 'D' + 'F' = 'AEMPZ\_DF'

Hint: Use chr() and ord() functions '''

```
In [ ]: a = input("Enter input string: ")

out = ""
for i in range(0, len(a), 2):
    letter = a[i]
    digit = int(a[i + 1])
    shifted_char = chr(ord(letter) + digit)
    out += letter + shifted_char

print(out)
```