1. **Python Program for Cycle Sort**

# Python program to implement cycle sort

def cycleSort(array):

  writes = 0

# Loop through the array to find cycles to rotate.

for cycleStart in range(0, len(array) - 1):

  item = array[cycleStart]

  # Find where to put the item.

pos = cycleStart

for i in range(cycleStart + 1, len(array)):

  if array[i] < item:

    pos += 1

  # If the item is already there, this is not a cycle.

if pos == cycleStart:

  continue

  # Otherwise, put the item there or right after any duplicates.

while item == array[pos]:

  pos += 1

  array[pos], item = item, array[pos]

  writes += 1

  # Rotate the rest of the cycle.

while pos != cycleStart:

  # Find where to put the item.

pos = cycleStart

for i in range(cycleStart + 1, len(array)):

  if array[i] < item:

    pos += 1

  # Put the item there or right after any duplicates.

while item == array[pos]:

  pos += 1

  array[pos], item = item, array[pos]

  writes += 1

return writes

# driver code

arr = [1, 8, 3, 9, 10, 10, 2, 4 ]

n = len(arr)

cycleSort(arr)

print("After sort : ")

for i in range(0, n) :

  print(arr[i], end = \' \')

**Output:** After sort: 1 2 3 4 8 9 10 10

1. **Python Program for Stooge Sort**

# Python program to implement stooge sort

def stoogesort(arr, l, h):

if l >= h:

return

# If first element is smaller

# than last,swap them

if arr[l]>arr[h]:

t = arr[l]

arr[l] = arr[h]

arr[h] = t

# If there are more than 2 elements in

# the array

if h-l+1 > 2:

t = (int)((h-l+1)/3)

# Recursively sort first 2/3 elements

stoogesort(arr, l, (h-t))

# Recursively sort last 2/3 elements

stoogesort(arr, l+t, (h))

# Recursively sort first 2/3 elements

# again to confirm

stoogesort(arr, l, (h-t))

# driver

arr = [2, 4, 5, 3, 1]

n = len(arr)

stoogesort(arr, 0, n-1)

for i in range(0, n):

print(arr[i], end = ' ')

**Output:** 1 2 3 4 5

1. **Python Program to print the pattern ‘G’**

# Python program to print pattern G

def Pattern(line):

  pat=""

  for i in range(0,line):

    for j in range(0,line):

      if ((j == 1 and i != 0 and i != line-1) or ((i == 0 or

        i == line-1) and j > 1 and j < line-2) or (i == ((line-1)/2)

        and j > line-5 and j < line-1) or (j == line-2 and

        i != 0 and i != line-1 and i >=((line-1)/2))):

        pat=pat+"\*"

      else:

        pat=pat+" "

    pat=pat+"\n"

  return pat

# Driver Code

line = 7

print(Pattern(line))

**output:**

\*\*\*

\*

\*

\* \*\*\*

\* \*

\* \*

\*\*\*

1. **Python Program to print an Inverted Star Pattern**

# python 3 code to print inverted star

# pattern

# n is the number of rows in which

# star is going to be printed.

n=11

# i is going to be enabled to

# range between n-i t 0 with a

# decrement of 1 with each iteration.

# and in print function, for each iteration,

# ” ” is multiplied with n-i and ‘\*’ is

# multiplied with i to create correct

# space before of the stars.

for i in range (n, 0, -1):

  print((n-i) \* ' ' + i \* '\*')

**output:**

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

1. **Python Program to print double sided stair-case pattern**

# Python3 Program to demonstrate

# staircase pattern

# function definition

def pattern(n):

  # for loop for rows

  for i in range(1,n+1):

    # conditional operator

    k =i + 1 if(i % 2 != 0) else i

    # for loop for printing spaces

    for g in range(k,n):

      if g>=k:

        print(end=" ")

    # according to value of k carry

    # out further operation

    for j in range(0,k):

      if j == k - 1:

        print(" \* ")

      else:

        print(" \* ", end = " ")

# Driver code

n = 10

pattern(n)

**Output:**

\* \*

\* \*

\* \* \* \*

\* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \* \*

1. **Python Program to print with your own font**

# Python3 code to print input in your own font

name = "GEEK"

# To take input from User

# name = input("Enter your name: \n\n")

length = len(name)

l = ""

for x in range(0, length):

  c = name[x]

  c = c.upper()

  if (c == "A"):

    print("..######..\n..#....#..\n..######..", end = " ")

    print("\n..#....#..\n..#....#..\n\n")

  elif (c == "B"):

    print("..######..\n..#....#..\n..#####...", end = " ")

    print("\n..#....#..\n..######..\n\n")

  elif (c == "C"):

    print("..######..\n..#.......\n..#.......", end = " ")

    print("\n..#.......\n..######..\n\n")

  elif (c == "D"):

    print("..#####...\n..#....#..\n..#....#..", end = " ")

    print("\n..#....#..\n..#####...\n\n")

  elif (c == "E"):

    print("..######..\n..#.......\n..#####...", end = " ")

    print("\n..#.......\n..######..\n\n")

  elif (c == "F"):

    print("..######..\n..#.......\n..#####...", end = " ")

    print("\n..#.......\n..#.......\n\n")

  elif (c == "G"):

    print("..######..\n..#.......\n..#.####..", end = " ")

    print("\n..#....#..\n..#####...\n\n")

  elif (c == "H"):

    print("..#....#..\n..#....#..\n..######..", end = " ")

    print("\n..#....#..\n..#....#..\n\n")

  elif (c == "I"):

    print("..######..\n....##....\n....##....", end = " ")

    print("\n....##....\n..######..\n\n")

  elif (c == "J"):

    print("..######..\n....##....\n....##....", end = " ")

    print("\n..#.##....\n..####....\n\n")

  elif (c == "K"):

    print("..#...#...\n..#..#....\n..##......", end = " ")

    print("\n..#..#....\n..#...#...\n\n")

  elif (c == "L"):

    print("..#.......\n..#.......\n..#.......", end = " ")

    print("\n..#.......\n..######..\n\n")

  elif (c == "M"):

    print("..#....#..\n..##..##..\n..#.##.#..", end = " ")

    print("\n..#....#..\n..#....#..\n\n")

  elif (c == "N"):

    print("..#....#..\n..##...#..\n..#.#..#..", end = " ")

    print("\n..#..#.#..\n..#...##..\n\n")

  elif (c == "O"):

    print("..######..\n..#....#..\n..#....#..", end = " ")

    print("\n..#....#..\n..######..\n\n")

  elif (c == "P"):

    print("..######..\n..#....#..\n..######..", end = " ")

    print("\n..#.......\n..#.......\n\n")

  elif (c == "Q"):

    print("..######..\n..#....#..\n..#.#..#..", end = " ")

    print("\n..#..#.#..\n..######..\n\n")

  elif (c == "R"):

    print("..######..\n..#....#..\n..#.##...", end = " ")

    print("\n..#...#...\n..#....#..\n\n")

  elif (c == "S"):

    print("..######..\n..#.......\n..######..", end = " ")

    print("\n.......#..\n..######..\n\n")

  elif (c == "T"):

    print("..######..\n....##....\n....##....", end = " ")

    print("\n....##....\n....##....\n\n")

  elif (c == "U"):

    print("..#....#..\n..#....#..\n..#....#..", end = " ")

    print("\n..#....#..\n..######..\n\n")

  elif (c == "V"):

    print("..#....#..\n..#....#..\n..#....#..", end = " ")

    print("\n...#..#...\n....##....\n\n")

  elif (c == "W"):

    print("..#....#..\n..#....#..\n..#.##.#..", end = " ")

    print("\n..##..##..\n..#....#..\n\n")

  elif (c == "X"):

    print("..#....#..\n...#..#...\n....##....", end = " ")

    print("\n...#..#...\n..#....#..\n\n")

  elif (c == "Y"):

    print("..#....#..\n...#..#...\n....##....", end = " ")

    print("\n....##....\n....##....\n\n")

  elif (c == "Z"):

    print("..######..\n......#...\n.....#....", end = " ")

    print("\n....#.....\n..######..\n\n")

  elif (c == " "):

    print("..........\n..........\n..........", end = " ")

    print("\n..........\n\n")

  elif (c == "."):

    print("----..----\n\n")

**Output:**

..######..

..#.......

..#.####..

..#....#..

..#####...

..######..

..#.......

..#####...

..#.......

..######..

..######..

..#.......

..#####...

..#.......

..######..

..#...#...

..#..#....

..##......

..#..#....

..#...#...

1. **Get Current Date and Time using Python**

# Getting current date and time using now().

# importing datetime module for now()

import datetime

# using now() to get current time

current\_time = datetime.datetime.now()

# Printing value of now.

print("Time now at greenwich meridian is:", current\_time)

**Output:**

Time now at greenwich meridian is: 2023-01-11 01:59:59.381247

1. **Python | Find yesterday’s, today’s and tomorrow’s date**

# Python program to find yesterday,

# today and tomorrow

# Import datetime and timedelta

# class from datetime module

from datetime import datetime, timedelta

# Get today's date

presentday = datetime.now() # or presentday = datetime.today()

# Get Yesterday

yesterday = presentday - timedelta(1)

# Get Tomorrow

tomorrow = presentday + timedelta(1)

# strftime() is to format date according to

# the need by converting them to string

print("Yesterday = ", yesterday.strftime('%d-%m-%Y'))

print("Today = ", presentday.strftime('%d-%m-%Y'))

print("Tomorrow = ", tomorrow.strftime('%d-%m-%Y'))

**Output:**

Yesterday = 10-01-2023

Today = 11-01-2023

Tomorrow = 12-01-2023

1. **Python program to convert time from 12 hour to 24 hour format**

# Python program to convert time

# from 12 hour to 24 hour format

# Function to convert the date format

def convert24(str1):

  # Checking if last two elements of time

  # is AM and first two elements are 12

  if str1[-2:] == "AM" and str1[:2] == "12":

    return "00" + str1[2:-2]

  # remove the AM

  elif str1[-2:] == "AM":

    return str1[:-2]

  # Checking if last two elements of time

  # is PM and first two elements are 12

  elif str1[-2:] == "PM" and str1[:2] == "12":

    return str1[:-2]

  else:

    # add 12 to hours and remove PM

    return str(int(str1[:2]) + 12) + str1[2:8]

# Driver Code

print(convert24("08:05:45 PM"))

**Output:** 20:05:45