|  |  |  |
| --- | --- | --- |
| **PROGRAM 1:**  a, b, c = map(int, input("Enter 3 integers which are sides of a triangle: ").split())  print(f"a={a}, b={b}, c={c}")  if a < b + c and b < a + c and c < a + b:  is\_triangle = 'y'  else:  is\_triangle = 'n'  if is\_triangle == 'y':  if a == b and b == c:  print("Equilateral triangle")  elifa != b and a != c and b != c:  print("Scalene triangle")  else:  print("Isosceles triangle")  else:  print("Not a triangle")  **PROGRAM 2 and 3:**  istriangle = 'n'  while istriangle != 'y':  a, b, c = map(int, input("\nEnter 3 integers which are sides of a triangle: ").split())  print(f"a={a}\tb={b}\tc={c}")  c1 = 1 <= a <= 10  c2 = 1 <= b <= 10  c3 = 1 <= c <= 10  if not c1:  print(f"The value of a={a} is not in the range of permitted values.")  if not c2:  print(f"The value of b={b} is not in the range of permitted values.")  if not c3:  print(f"The value of c={c} is not in the range of permitted values.")  if c1 and c2 and c3:  if a < b + c and b < a + c and c < a + b:  istriangle = 'y'  else:  print("Not a triangle")  if istriangle == 'y':  if a == b == c:  print("Equilateral triangle")  elifa != b != c != a:  print("Scalene triangle")  else:  print("Isosceles triangle") | | **PROGRAM 4 and 5:**  # Initializing lock, stock and barrel prices  lock\_price = 45  stock\_price = 30  barrel\_price = 25  # Initializing Production limits  lock\_limit = 70  stock\_limit = 80  barrel\_limit = 90  # Initializing Commission levels  commission\_level\_1 = 0.10  commission\_level\_2 = 0.15  commission\_level\_3 = 0.20  # Collecting the salesperson's inputs  locks\_sold = int(input("Enter the number of locks sold: "))  stocks\_sold = int(input("Enter the number of stocks sold: "))  barrels\_sold = int(input("Enter the number of barrels sold: "))  # Calculating total sales  total\_sales = (locks\_sold \* lock\_price) + (stocks\_sold \* stock\_price) + (barrels\_sold \* barrel\_price)  # Printing total sales  print("The total sales for the salesperson is: $" + str(total\_sales))  # Calculating commission  if total\_sales<= 1000:  commission = total\_sales \* commission\_level\_1  elif total\_sales<= 1800:  commission = (1000 \* commission\_level\_1) + ((total\_sales - 1000) \* commission\_level\_2)  else:  commission = (1000 \* commission\_level\_1) + (800 \* commission\_level\_2) + ((total\_sales - 1800) \* commission\_level\_3)  # Printing the result  print("The total commission for the salesperson is: $" + str(commission)) |
| **PROGRAM 6:**  def binary\_search(arr, target):  low = 0  high = len(arr) - 1  while low <= high:  mid = (low + high) // 2  mid\_element = arr[mid]  if mid\_element == target:  return mid # Target found, return the index  elifmid\_element< target:  low = mid + 1 # Search in the right half  else:  high = mid - 1 # Search in the left half  return -1 # Target not found  # Example usage:  arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  target = 7  result = binary\_search(arr, target)  if result != -1:  print(f"Element {target} found at index {result}")  else:  print(f"Element {target} not found")  **PROGRAM 7:**  def quicksort(array):  if len(array)<2:  return array  else:  pivot=array[0]  less= [i for i in array[1:] if i<=pivot]  greater=[i for i in array[1:]if i>pivot]  return quicksort(less) + [pivot] + quicksort(greater)9.  print(quicksort([10,5,2,3]))  **PROGRAM: 08**  per = float(input())  if per < 0 or per > 100:  print("Invalid percentage, enter again")  else:  if per >= 90:  grade = 'A'  elif per >= 80:  grade = 'B'  elif per >= 70:  grade = 'C'  elif per >= 60:  grade = 'D'  else: | | grade = 'E'  if grade == 'A':  print("EXCELLENT")  elif grade == 'B':  print("Very Good")  elif grade == 'C':  print("Good")  elif grade == 'D':  print("Above Average")  else:  print("Satisfactory")  print(f"The percentage = {per} and grade is {grade}")  **PROGRAM: 09**  def NextDate(month, date, year):  # Checking for valid inputs  if month not in range(1,12):  print ("Value of month not in the range 1...12")  return  elif date not in range(1,31):  print ("Value of date not in the range 1...31")  return  elif year not in range(1812,2012):  print ("Value of year not in the range 1812...2012") return  # Calculating next date if given date is not invalid  if date == 31 and month in [1, 3, 5, 7, 8, 10, 12]:  date = 1  if month == 12:  month = 1  year += 1  else:  month += 1  elif date == 30 and month in [4, 6, 9, 11]:  date = 1 month += 1  elif date == 28 and month == 2 and year%4 != 0:  date = 1  month += 1  elif date == 29 and month == 2 and year%4 != 0:  date = 1  month += 1  else:  date += 1  print("The next date is:",month,date,year)  # Driver Code  month = 2  date = 28  year = 2000  print("Date enterred is: ", month, date, year)  print()  NextDate(month, date, year) | | |