1. Data set: [9, 12, 6, 4, 2, 1, 15, 8, 5]
   Using linear search, what is the index of the number 15 in the data set above? (1)

2. Data set: [3, 6, 9, 12, 15, 18, 21, 24, 27]
   Using linear search, how many comparisons are needed to find the number 18 in the data set above? (1)
3. Data set: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39]
   Compare the efficiency of linear and binary search algorithms for finding the number 21. (2)

   Read through the program below which shows a linear search

```
1  arr = [11,21,3,42,55,6,7,28,19,10]
2  target = 11
3  def linear_search(Parr, Ptarget):
4      comparisons = 0
5
6      for i in range(len(Parr)):
7          comparisons = comparisons + 1
8
9          if Parr[i] == Ptarget:
10             return (i)
11     return -1
12
13 result = linear_search(arr,target)
14 if result == -1:
15     print ("Target not found")
16 else:
17     print ("Target found at position " + str(result + 1))
```

4. Name a user-defined function in the program. (1)
5. Name an in-built function in the program (1)
6. Name a parameter in the program. (1) Complete line 15 to call the function. (1)
7. How do you know that linear_search() is a function and not a procedure? (1)
8. What number is len(Parr)?
9. Is the program efficient? Ie. Will it stop when the target is found or will it keep going until the end of the list? (1)
10. What will be displayed if the target is 11? (1)

    It would be a good idea to type this program up, complete it and debug it to help you understand and answer the questions.

ANSWERS

1. Data set: [9, 12, 6, 4, 2, 1, 15, 8, 5]
   Indexing starts at 0. So number 15 is in index position 6.

2. 5 comparisons to find number 18.
   A **linear search** scans through a list of elements one by one, comparing each element with the target value until a match is found or the entire list is checked.

3. Data set: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39]
   Linear search scans through the data set sequentially, comparing each element with the target value. It is best used when the data is **not in order** or for smaller lists.
   For this data set, a linear search would take 8 steps to find number 21.

   Binary search operates more efficiently when the data is **sorted**.
   It divides the data set in half at each step, speeding up the search.
   For this sorted data set, binary search would take fewer steps as it would find the number 21 in only 4 steps.
   Binary search is more efficient for larger datasets and sorted data.

4. A user-defined function is linear_search
   Recognise a user-defined function by the command word def and the word return at the end of the code.

5. A built-in subprogram is len( ) or print( )

6. A parameter in the program is Ptarget or Parr

7. Linear_search is a function because it returns a value back to the main program….hence the word 'return' at the end. A procedure doesn't return a value.

8. len(Parr) is equal to 10. The array contains 10 values. The array is passed into the sub program via the parameter Parr.

9. The program is efficient because the loop is broken by the first return command if the target is found.

10. Target found at position 1