

Program 1:

06	11-sorted.py
1	<code>def next():</code>
2	<code> input("Next> ")</code>
3	
4	<code>str_list = ['apple', 'banana', 'cherry', 'watermelon', 'pear']</code>
5	<code>num_list = [23, 49, -15, 8, 77]</code>
6	<code>mix_list = ['apple', 'banana', 77, 66]</code>
7	
8	<code>print(f"str_list: {str_list}\nnum_list: {num_list} \n")</code>
9	<code>next()</code>
10	<code>print(f"str_list: {sorted(str_list)}\nnum_list: {sorted(num_list)}</code>
11	<code>\n")</code>
12	<code>next()</code>
13	<code>print(f"str_list: {sorted(str_list, reverse=True)}\nnum_list:</code>
14	<code>{sorted(num_list, reverse=True)} \n")</code>
15	<code>next()</code>
16	<code>print(f"str_list: {sorted(str_list, key=len)}\nnum_list:</code>
17	<code>{sorted(num_list, key=abs)} \n")</code>
18	<code>next()</code>
19	<code>newlist = sorted(num_list, key=abs, reverse=True)</code>
20	<code>print(f"num_list sorted key=abs reverse: {newlist}")</code>
21	<code>next()</code>
22	
23	<code># Using sort()</code>
24	<code>str_list.sort(key=len)</code>
25	<code>print(f"sort on length of str_list: {str_list}")</code>
26	<code>next()</code>
27	
28	<code>num_list.sort(key=abs, reverse=True)</code>
29	<code>print(f"reverse sort on absolute value of num_list: {num_list}")</code>
30	<code>next()</code>
31	
32	<code># This one will fail</code>
33	<code>print('mix_list:', sorted(mix_list))</code>

Program 2:

06	12-sorted 2d.py
1	<code>def next():</code>
2	<code> input("Next> ")</code>
3	
4	<code>def addup(elem):</code>
5	<code> return elem[0]+elem[1]</code>
6	
7	<code>def on_y(elem):</code>
8	<code> return elem[1]</code>
9	
10	<code>def diff(elem): # absolute diff</code>
11	<code> return abs(elem[0]-elem[1])</code>
12	
13	<code>mylist = [(1,1), (2,-2), (-1, 0), (1,0), (1,3), (2,4)]</code>
14	<code>print(f"\nmylist: {mylist}")</code>
15	<code>next()</code>
16	<code>print(f"mylist sorted: {sorted(mylist)}")</code>
17	<code>next()</code>
18	<code>print(f"mylist sorted by addup: {sorted(mylist, key=addup)}")</code>
19	<code>next()</code>
20	<code>print(f"mylist sorted by diff: {sorted(mylist, key=diff)}")</code>
21	<code>next()</code>
22	<code>print(f"mylist reverse sorted by on_y: {sorted(mylist, key=on_y,</code>
23	<code>reverse=True)}")</code>

Program 3:

06	13-sorted tuple.py
1	<code>tuple1 = (3, 5, 4, -1)</code>
2	<code>tuple2 = ((1,2), (4,5), (3,2), (2,4))</code>
3	<code>print(f"Tupke 1: {tuple1}\nTuple 2: {tuple2}\n")</code>
4	<code>print("Tuple 1 sorted in reverse", sorted(tuple1, reverse=True))</code>
5	<code>list1 = sorted(tuple1)</code>
6	<code>print("Tuple 1 sorted", list1)</code>
7	<code>print("Tuple 1 sorted", tuple(list1))</code>
8	<code>print("\nTuple 2 sorted", sorted(tuple2))</code>

Program 4:

```
06 14-sorted.py
1  def next():
2      input("Next> ")
3
4  list1 = [19, 3, 41, 45, -22, 111, 2, 29]
5  list2 = [9, 8, 7, 6, 5, 4, 3, 2, 1]
6  list3 = ['a', 'b', 'C', 'A', 'B', 'c', 'aa', 'about', 'this', 'that']
7
8  print(list1)
9  print(list2)
10 print(list3)
11 next()
12
13 list4 = sorted(list1, key=abs)
14 print(f"List1 sorted key=abs {list4}")
15 next()
16
17 list5 = sorted(list2)
18 print(f"List2 sorted {list5}")
19 next()
20
21 list6 = sorted(list3, key=str.lower, reverse=True)
22 print(f"List3 sorted key=str.lower() reverse {list6}")
```

Program 5:

```
06 15-sort sorted.py
1  def next():
2      input("Next> ")
3
4  str_list = ['banana', 'apple', 'watermelon', 'pear', 'peach']
5  nbr_list = [10, -3, 15, 9, 99, -77, 9]
6  print(f"str_list = {str_list}")
7  print(f"nbr_list = {nbr_list}")
8  next()
9
10 print(f"returned by sorted(): {sorted(nbr_list)}")
11 print(f"returned by sorted(): {sorted(str_list, reverse=True)}\n")
12 next()
13
14 print(f"str_list = {str_list}")
15 print(f"nbr_list = {nbr_list}")
16 # The lists were not changed
17 next()
18
19 print(str_list.sort())
20 print(nbr_list.sort(reverse=True))
21 next()
22
23 print(f"\nafter sort(): str_list = {str_list}")
24 print(f"after sort(): nbr_list = {nbr_list}")
25 # The change to the lists were done through the parameters
```

Program 6:

```
06 21-bubble.py
1  def next():
2      input("Next> ")
3
4  def bubble(a):
5      for ub in range(len(a)-1, 0, -1):
6          for i in range(ub):
7              if a[i] > a[i+1]:
8                  a[i], a[i+1] = a[i+1], a[i]
9          print(a)
10     return(a)
11
12 list1 = [19, 3, 41, 45, 22, 111, 2, 29]
13 list2 = [5, 7, 4, 9, 3]
14 list3 = ['a', 'b', 'C', 'A', 'B', 'c', 'aa', 'about', 'this', 'that']
15
16 print(list1)
17 bubble(list1)
18 print()
19
20 print(list2)
21 bubble(list2)
22 print()
23
24 print(list3)
25 bubble(list3)
26 print()
```

Program 7:

06	22-bubble_detail.py
1	<code>def next():</code>
2	<code> input("Next> ")</code>
3	
4	<code>def bubble(a):</code>
5	<code> for ub in range(len(a)-1, 0, -1):</code>
6	<code> print('***', ub)</code>
7	<code> for i in range(ub):</code>
8	<code> print(' ' * 32, f'{a[i]:3d} <> {a[i+1]:3d}', end = ' ')</code>
9	<code> if a[i] > a[i + 1]:</code>
10	<code> print(' <- Swap')</code>
11	<code> a[i], a[i + 1] = a[i + 1], a[i]</code>
12	<code> else:</code>
13	<code> print()</code>
14	<code> print(a)</code>
15	<code> return(a) # ?</code>
16	
17	
18	<code>list1 = [19, 3, 41, 45, 22, 111, 2, 29]</code>
19	<code>list2 = [9, 8, 7, 6, 5]</code>
20	<code>list3 = [1, 3, 5, 7, 9, 11]</code>
21	
22	<code>print(list1)</code>
23	<code>list4 = bubble(list1)</code>
24	<code>print(list1)</code>
25	<code>next()</code>
26	
27	<code>print(list2)</code>
28	<code>list2 = bubble(list2)</code>
29	<code>next()</code>
30	
31	<code>print(list3)</code>
32	<code>list3 = bubble(list3)</code>

Program 8:

```
06 23-selection sort.py
1
2 def next():
3     input("Next> ")
4
5 def swap(list, i, j):
6     list[i], list[j] = list[j], list[i]
7
8 def selection_sort(a):
9     for i in range(len(a)-1):
10        min_i = i
11        for j in range(i +1, len(a)):
12            if a[min_i] > a[j]:
13                min_i = j
14        swap(a, i, min_i)
15        print(a)
16
17
18 list1 = [19, 2, 31, 45, 30, 11, 121, 27]
19 print(list1)
20 selection_sort(list1)
21 next()
22
23 list2 = [99, 88, 77, 66, 55]
24 print(list2)
25 selection_sort(list2)
26 next()
27
28 list3 = [8, 7, 6, 5, 4, 3, 2, 1]
29 print(list3)
30 selection_sort(list3)
31 next()
32
33 list3 = [6, 5, 7, 8, 4, 3, 2, 1]
34 print(list3)
35 selection_sort(list3)
```

Program 9:

06	24-selection detail.py
1	<code>def next():</code>
2	<code> input("Next> ")</code>
3	
4	<code>def swap(list, i, j):</code>
5	<code> list[i], list[j] = list[j], list[i]</code>
6	
7	<code>def selection_sort(a):</code>
8	<code> for i in range(len(a)-1):</code>
9	<code> min_i = i</code>
10	<code> for j in range(i +1, len(a)):</code>
11	<code> if a[min_i] > a[j]:</code>
12	<code> min_i = j</code>
13	<code> swap(a, i, min_i)</code>
14	<code> print(a)</code>
15	
16	
17	<code>list1 = [19, 2, 31, 45, 30, 11, 121, 27]</code>
18	<code>print(list1)</code>
19	<code>selection_sort(list1)</code>
20	<code>next()</code>
21	
22	<code>list2 = [99, 88, 77, 66, 55]</code>
23	<code>print(list2)</code>
24	<code>selection_sort(list2)</code>
25	<code>next()</code>
26	
27	<code>list3 = [8, 7, 6, 5, 4, 3, 2, 1]</code>
28	<code>print(list3)</code>
29	<code>selection_sort(list3)</code>
30	<code>next()</code>
31	
32	<code>list3 = [6, 5, 7, 8, 4, 3, 2, 1]</code>
33	<code>print(list3)</code>
34	<code>selection_sort(list3)</code>

Program 10:

```
06 31-linear.py
1  def next():
2      input("Next> ")
3
4  def search(x, a):
5      foundIndex= None
6      count = 0
7      for i in range(len(a)):
8          if a[i]==x:
9              count += 1
10             foundIndex=i
11         return foundIndex, count
12
13 mylist = [2, 3, 25, 7, 7, 4, 9, 8, 7, 16, 25, 2]
14 test = [10, 8, 7]
15 for j in test:
16     print('\nList:', mylist)
17     idx, count = search(j, mylist)
18     if idx != None:
19         print(f"Searching for {j}: Found, index = {idx}, value =
20 {mylist[idx]}, count = {count}")
21     else:
22         print(f"Searching for {j}: Not found, count = {count}")
23     next()
```

Program 11:

```
06 32-linear-first.py
1  def next():
2      input("Next> ")
3
4  def search(x,a):
5      count = 0
6      for i in range(len(a)):
7          if a[i]==x:
8              count += 1
9              return i, count
10         return None, count
11
12 mylist = [2, 3, 25, 7, 7, 4, 9, 8, 7, 16, 25]
13 test = [10, 8, 7]
14
15 for j in test:
16     print('\nList:', mylist)
17     idx, count = search(j, mylist)
18     if idx!=None:
19         print(f"Searching for {j}: Found, index = {idx}, value =
20 {mylist[idx]}, count = {count}")
21     else:
22         print(f"Searching {j}: Not found. count = {count}")
23     next()
```


Program 12:

```
06 33-recursive.py
1  def search(x,list,lb,ub):
2      print(f"Searching {x} between {lb} and {ub}")
3      if lb >= ub:
4          return None
5      elif list[lb]==x:
6          return lb
7      else:
8          return(search(x,list,lb+1, ub))
9
10 mylist = [2, 3, 25, 7, 7, 4, 9, 8, 7, 16, 25]
11 test = [10, 8, 7]
12 for j in test:
13     print('\nList:', mylist)
14     idx = search(j, mylist, 0, len(mylist))
15     if idx != None:
16         print(f"Searching for {j}: Found, index = {idx}, value =
17 {mylist[idx]}".)
18     else:
19         print(f"Searching for {j}: Not found.")
```

Program 13:

```
06 41-binary.py
1  def search(x,a):
2      low, high = 0, len(a)
3      while low<high:
4          mid=(low+high)//2
5          print('\n', a[low:high])
6          print(f"Search Range [{low:2d},{high:2d}], mid = {mid:2d},"", end='
7 ')
8          if x==a[mid]:
9              print('=' , f'{a[mid]:2d}', end=' ')
10             return mid
11             elif x<a[mid]:
12                 print('<', f'{a[mid]:2d}', end=' ')
13                 high=mid
14             else:
15                 print('>', f'{a[mid]:2d}', end=' ')
16                 low =mid+1
17                 print(f"New Range [{low:2d},{high:2d}]")
18             return None
19
20 a = [3, 5, 7, 9, 10, 11, 12, 13, 14, 15, 17, 23, 32, 35, 37, 43, 55, 66, 77]
21 print(a)
22 x = int(input('Enter an integer to search: '))
23 idx = search(x, a)
24 if idx!=None:
25     print(f"\nFound {x} at index = {idx}")
26 else:
27     print(f"\n{x} not found")
```

Program 14:

```
06 42-binary.py
1  def search(x,a):
2      low, high = 0, len(a)
3      while low<high:
4          mid=(low+high)//2
5          print('\n', a[low:high])
6          print(f"Search Range [{low:2d},{high:2d}], mid = {mid:2d}",
7  end=' ')
8          if x==a[mid]:
9              print('=', f'{a[mid]:2d}', end=' ')
10             return mid
11             elif x<a[mid]:
12                 print('<', f'{a[mid]:2d}', end=' ')
13                 high=mid
14             else:
15                 print('>', f'{a[mid]:2d}', end=' ')
16                 low =mid+1
17                 print(f"New Range [{low:2d},{high:2d}]")
18             return None
19
20 a = [3, 5, 7, 9, 10, 11, 12, 13, 14, 15, 17, 23, 32, 35, 37, 43, 55,
21 66, 77]
22 print(a)
23 x = int(input('Enter an integer to search: '))
24 idx = search(x, a)
25 if idx!=None:
26     print(f"\nFound {x} at index = {idx}")
27 else:
28     print(f"\n{x} not found")
```

Program 15:

```
06 43-recursive binary.py
1  def b_search(x,list, low, high):
2      while low<=high:
3          mid=(low+high)//2
4          print("b_search(",x,",list,",low, ",",high,")", sep='')
5          if x==list[mid]: return mid
6          elif x<list[mid]: return(b_search(x,list,low,mid-1))
7          else: return(b_search(x,list,mid+1,high))
8      return None
9
10 a = [3, 5, 7, 9, 10, 12, 15, 17, 23, 32, 35, 37, 43]
11
12 print (a)
13 idx = b_search(10, a, 0, len(a))
14 if idx!=None:
15     print("Found index =", idx, "value = ", a[idx])
16 else:
17     print("Not found")
```