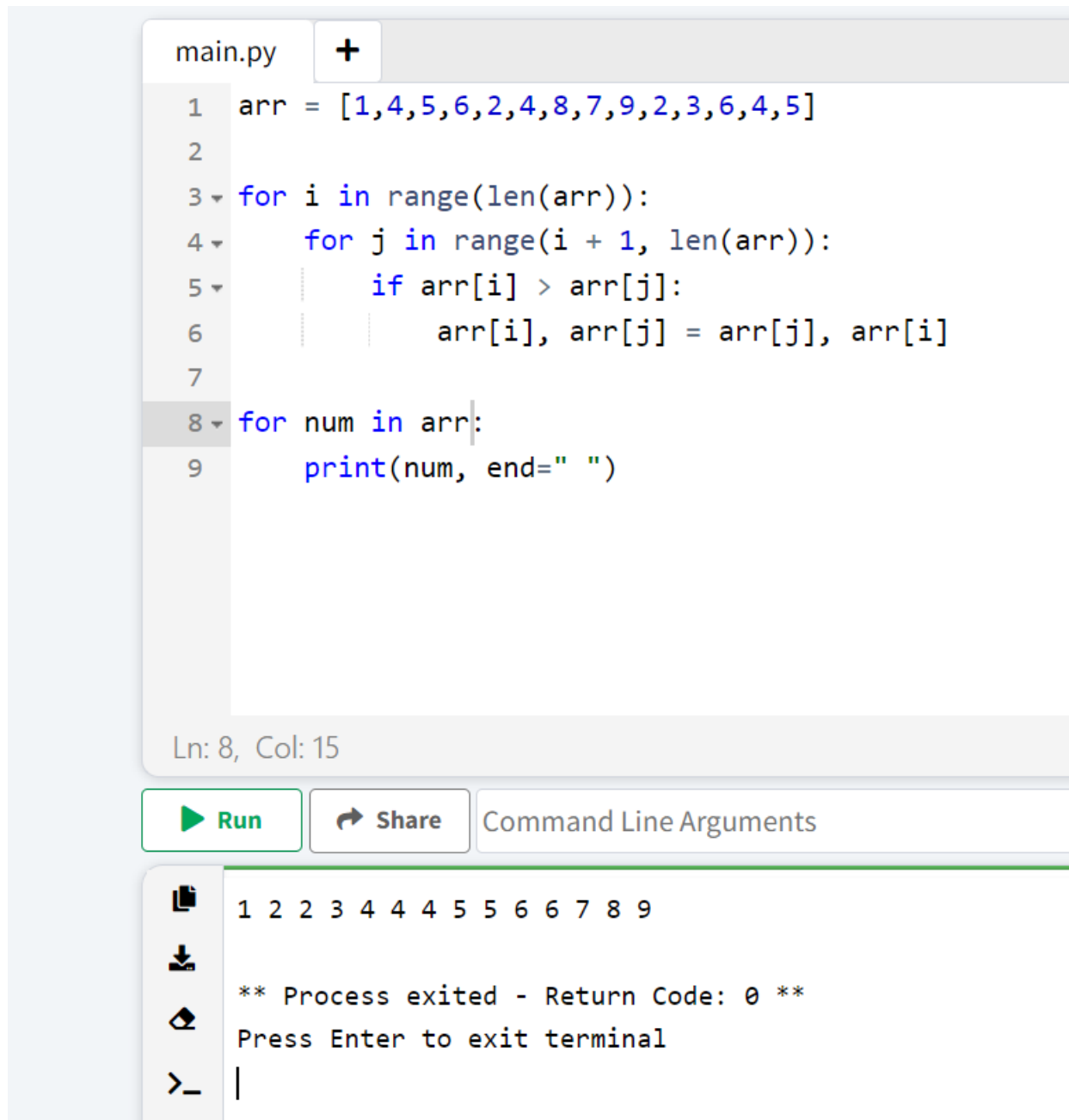


**Problem Statement :** Write a Program in Python to apply Bubble Sort Algorithm

**Objective :** To Write a Program in Python for Bubble Sort Algorithm

**Source Code & Input-Output :**



The screenshot shows a Python IDE with a file named 'main.py'. The code implements a Bubble Sort algorithm on an array [1, 4, 5, 6, 2, 4, 8, 7, 9, 2, 3, 6, 4, 5]. The code consists of 9 lines: line 1 initializes the array; lines 3-7 are a nested loop for sorting; line 8 iterates through the sorted array; and line 9 prints each element. Below the code editor, there are buttons for 'Run', 'Share', and 'Command Line Arguments'. The output terminal shows the sorted array '1 2 2 3 4 4 4 5 5 6 6 7 8 9', a success message '\*\* Process exited - Return Code: 0 \*\*', and a prompt 'Press Enter to exit terminal'.

```
main.py +
1 arr = [1,4,5,6,2,4,8,7,9,2,3,6,4,5]
2
3 for i in range(len(arr)):
4     for j in range(i + 1, len(arr)):
5         if arr[i] > arr[j]:
6             arr[i], arr[j] = arr[j], arr[i]
7
8 for num in arr:
9     print(num, end=" ")

Ln: 8, Col: 15

Run Share Command Line Arguments

1 2 2 3 4 4 4 5 5 6 6 7 8 9
** Process exited - Return Code: 0 **
Press Enter to exit terminal
>_ |
```

**Problem Statement :** Write a Program in Python to apply Insertion Sort Algorithm

**Objective :** To Write a Program in Python for Insertion Sort Algorithm

**Source Code & Input-Output :**

```
main.py +
1  n = 9
2  arr = [4,2,5,1,3,7,9,12,36]
3
4  for i in range(1, n):
5      key = arr[i]
6      j = i - 1
7      while j >= 0 and arr[j] > key:
8          arr[j + 1] = arr[j]
9          j -= 1
10     arr[j + 1] = key
11
12 for i in range(n):
13     print(arr[i], end=" ")
14
```

Ln: 2, Col: 16

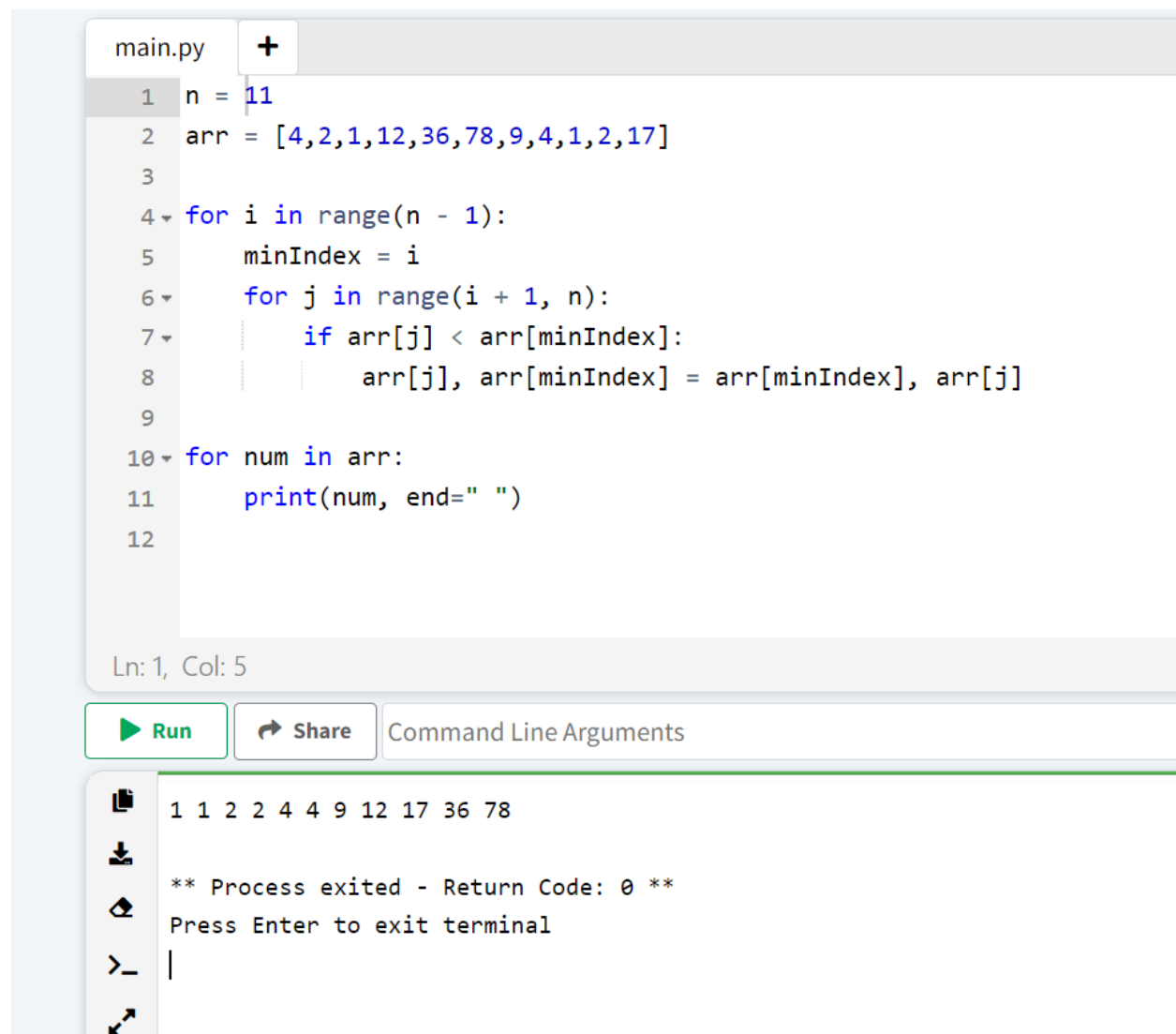
[Run](#) [Share](#) Command Line Arguments

```
1 2 3 4 5 7 9 12 36
** Process exited - Return Code: 0 **
Press Enter to exit terminal
>_ |
```

**Problem Statement :** Write a Program in Python to apply Selection Sort Algorithm

**Objective :** To Write a Program in Python for Selection Sort Algorithm



**Source Code & Input-Output :**





The screenshot shows a Python IDE with a file named 'main.py'. The code implements the Selection Sort algorithm. It starts by setting `n = 11` and creating an array `arr = [4, 2, 1, 12, 36, 78, 9, 4, 1, 2, 17]`. The algorithm uses nested loops: an outer loop for `i` from 0 to `n-2`, and an inner loop for `j` from `i+1` to `n-1`. In the inner loop, it finds the minimum element and swaps it with the element at index `i`. Finally, it prints the sorted array. Below the code editor, there are buttons for 'Run' and 'Share', and a 'Command Line Arguments' field. The output of the program is displayed in a terminal window, showing the sorted array: `1 1 2 2 4 4 9 12 17 36 78`. The terminal also shows the message `** Process exited - Return Code: 0 **` and `Press Enter to exit terminal`.


```
main.py +
1 n = 11
2 arr = [4, 2, 1, 12, 36, 78, 9, 4, 1, 2, 17]
3
4 for i in range(n - 1):
5     minIndex = i
6     for j in range(i + 1, n):
7         if arr[j] < arr[minIndex]:
8             arr[j], arr[minIndex] = arr[minIndex], arr[j]
9
10 for num in arr:
11     print(num, end=" ")
12
```


Ln: 1, Col: 5

 Run  Share Command Line Arguments

 1 1 2 2 4 4 9 12 17 36 78

 \*\* Process exited - Return Code: 0 \*\*

 Press Enter to exit terminal

 >\_ |

