```
import numpy as np

# Question 1: Array Creation
# 1.1 1D array from 5 to 25
arr1d = np.arange(5, 26)
print("1.1:", arr1d)

# 1.2 2D array (3x4), random integers between 10 and 50
arr2d = np.random.randint(10, 51, size=(3, 4))
print("1.2:\n", arr2d)

# Question 2: Array Attributes
print("2.1 Shape:", arr1d.shape)
print("2.1 Size:", arr1d.size)
print("2.1 Data type:", arr1d.dtype)

print("2.2 Shape:", arr2d.shape)
print("2.2 Size:", arr2d.size)
print("2.2 Data type:", arr2d.dtype)

# Question 3: Array Operations
array1 = np.array([2, 4, 6, 8, 10])
array2 = np.array([1, 3, 5, 7, 9])

print("3.2 Addition:", array1 + array2)
print("3.2 Subtraction:", array1 - array2)
print("3.2 Multiplication:", array1 * array2)
print("3.2 Division:", array1 / array2)

# Question 4: Broadcasting
arr_2d = np.arange(1, 10).reshape((3, 3))
print("4.1:\n", arr_2d)
print("4.2 Multiplied by 5:\n", arr_2d * 5)

# Question 5: Slicing and Indexing
arr4x4 = np.arange(10, 26).reshape((4, 4))
print("5.1:\n", arr4x4)
print("5.2 Second row:", arr4x4[1])
print("5.2 Last column:", arr4x4[:, -1])
arr4x4[0] = 0
print("5.3 First row replaced with zeros:\n", arr4x4)

# Question 6: Boolean Indexing
arr_bool = np.random.randint(20, 41, size=10)
print("6.1:", arr_bool)
print("6.2 Elements > 30:", arr_bool[arr_bool > 30])

# Question 7: Reshaping
arr_1d12 = np.arange(11, 23)
arr_2d3x4 = arr_1d12.reshape((3, 4))
print("7.1:", arr_1d12)
print("7.2 Reshaped (3x4):\n", arr_2d3x4)
```

```
# Question 8: Matrix Operations
A = np.array([[1, 2], [3, 4]])
B = np.array([[5, 6], [7, 8]])
print("8.1 Matrix A:\n", A)
print("8.1 Matrix B:\n", B)
print("8.2 Multiplication:\n", np.dot(A, B))
print("8.2 Transpose of A:\n", A.T)

# Question 9: Statistical Functions
arr_stats = np.random.randint(10, 61, size=15)
print("9.1 Array:", arr_stats)
print("9.2 Mean:", np.mean(arr_stats))
print("9.2 Median:", np.median(arr_stats))
print("9.2 Std Dev:", np.std(arr_stats))

# Question 10: Linear Algebra
A = np.array([[2, 1, 3], [0, 5, 6], [7, 8, 9]])
print("10.1 Matrix A:\n", A)
print("10.2 Determinant:", np.linalg.det(A))
print("10.2 Inverse:\n", np.linalg.inv(A))
eigvals, eigvecs = np.linalg.eig(A)
print("10.2 Eigenvalues:", eigvals)
print("10.2 Eigenvectors:\n", eigvecs)

# Question 11: Mobile Robot Euclidean Distance
positions = np.array([[0, 0], [2, 3], [4, 7], [7, 10], [10, 15]])
euclidean_distance = np.linalg.norm(positions[-1] - positions[0])
print("11.1 Euclidean distance:", euclidean_distance)

step_distances = np.linalg.norm(np.diff(positions, axis=0), axis=1)
total_distance = np.sum(step_distances)
print("11.2 Total distance step-by-step:", total_distance)
```