

## 11.1.1

```

1  #include <stdio.h>
2
3  v void swap(int *xp, int *yp) {
4      —> int temp = *xp;
5      —> *xp = *yp;
6      —> *yp = temp;
7  }
8
9  v void selectionSort(int arr[], int n) {
10     —> int i, j, max_idx;
11     v —> for (i = n-1; i > 0; i--) {
12         —> —> max_idx = i;
13         —> —> for (j = i-1; j >= 0; j--)
14             —> —> —> if (arr[j] > arr[max_idx])
15                 —> —> —> max_idx = j;
16         —> —> swap(&arr[max_idx], &arr[i]);
17     }
18 }
19 v void printArray(int arr[], int size) {
20     —> int i;
21     —> for (i = 0; i < size; i++)
22         —> —> printf("Value of a[%d] = %d\n", i, arr[i]);
23 }
24
25 v int main() {
26     —> int n;
27     —> printf("Enter value of n: ");
28     —> scanf("%d", &n);
29     —> int arr[n];
30     v —> for (int i = 0; i < n; i++) {
31         —> —> printf("Enter element for a[%d] : ", i);
32         —> —> scanf("%d", &arr[i]);
33     }
34     —> printf("Before sorting the elements in the array are\n");
35     —> printArray(arr, n);
36     —> selectionSort(arr, n);
37     —> printf("After sorting the elements in the array are\n");
38     —> printArray(arr, n);
39
40     —> }

```

## 10.1.2

```
1  #include <stdio.h>
2
3  v void swap(int *xp, int *yp) {
4      —>int temp = *xp;
5      —>*xp = *yp;
6      —>*yp = temp;
7  }
8
9  v void selectionSort(int arr[], int n) {
10     —>int i, j, min_idx;
11  v —>for (i = 0; i < n-1; i++) {
12     —>—>min_idx = i;
13     —>—>for (j = i+1; j < n; j++)
14     —>—>—>if (arr[j] < arr[min_idx])
15     —>—>—>—>min_idx = j;
16     —>—>swap(&arr[min_idx], &arr[i]);
17     —>}
18 }
19 v void printArray(int arr[], int size) {
20     —>int i;
21     —>for (i = 0; i < size; i++)
22     —>—>printf("Value of a[%d] = %d\n", i, arr[i]);
23 }
24
25 v int main() {
26     —>int n;
27     —>printf("Enter value of n :: ");
28     —>scanf("%d", &n);
29     —>int arr[n];
```

```
30  v —>for (int i = 0; i < n; i++) {
31     —>—>printf("Enter element for a[%d] :: ", i);
32     —>—>scanf("%d", &arr[i]);
33     —>}
34     —>printf("Before sorting the elements in the array are\n");
35     —>printArray(arr, n);
36     —>selectionSort(arr, n);
37     —>printf("After sorting the elements in the array are\n");
38     —>printArray(arr, n);
39
40     —>}
```

## 11.1.3

```

1  #include <stdio.h>
2
3  void insertionSort(int arr[], int n) {
4      int i, key, j;
5      for (i = 1; i < n; i++) {
6          key = arr[i];
7          j = i - 1;
8          while (j >= 0 && arr[j] > key) {
9              arr[j + 1] = arr[j];
10             j = j - 1;
11         }
12         arr[j + 1] = key;
13     }
14 }
15
16 void printArray(int arr[], int size) {
17     int i;
18     for (i = 0; i < size; i++)
19         printf("Value of a[%d] = %d\n", i, arr[i]);
20 }
21 int main() {
22     int n;
23     printf("Enter value of n : ");
24     scanf("%d", &n);
25     int arr[n];
26     for (int i = 0; i < n; i++) {
27         printf("Enter element for a[%d] : ", i);
28         scanf("%d", &arr[i]);
29     }

```

```

30     printf("Before sorting the elements in the array are\n");
31     printArray(arr, n);
32     insertionSort(arr, n);
33     printf("After sorting the elements in the array are\n");
34     printArray(arr, n);
35
36 }

```

### 11.2.1

```
1  #include <stdio.h>
2  #include <string.h>
3
4  v void shellSort(char arr[]) {
5      —> int n = strlen(arr);
6  v —> for (int gap = n/2; gap > 0; gap /= 2) {
7  v —> —> for (int i = gap; i < n; i += 1) {
8      —> —> —> char temp = arr[i];
9      —> —> —> int j;
10     —> —> —> for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)
11     —> —> —> —> arr[j] = arr[j - gap];
12     —> —> —> arr[j] = temp;
13     —> —> }
14     —> }
15 }
16
17 v int main() {
18     —> char str[100];
19     —> printf("Enter a string:");
20     —> scanf("%s", str);
21     —> shellSort(str);
22     —> printf("The sorted string is: %s\n", str);
23
24 }
```