1. #include <iostream>

using namespace std;

int main() {

int arr[] = {1, 2, 3, 4, 5};

cout << arr[5] << endl;

return 0;

}

What will be the output of the above code?

**a) Garbage value**

b) 1

c) 5

d) Compiler error

2. What is the time complexity to access an element in an array by its index?

a) O(n)

**b) O(1)**

c) O(log n)

d) O(n^2)

3. #include <iostream>

using namespace std;

int main() {

int arr[] = {4, 7, 2, 9, 1};

int n = sizeof(arr) / sizeof(arr[0]);

int key = 7;

int index = -1;

for (int i = 0; i < n; i++) {

if (arr[i] == key) {

index = i;

break;

}

}

cout << index;

return 0;

}

What will be the output when the above code is executed with key = 7?

a) 2

b) 3

c) 1

**d) -1**

4. When an element is deleted from an array, the remaining elements are shifted to fill the empty space. The time complexity for deletion from the end of the array is:

a) O(1)

**b) O(1) (amortized)**

c) O(n)

d) O(log n)

5. #include <iostream>

using namespace std;

int main() {

int arr1[] = {2, 5, 8, 10};

int arr2[] = {1, 4, 6, 9};

int n1 = sizeof(arr1) / sizeof(arr1[0]);

int n2 = sizeof(arr2) / sizeof(arr2[0]);

int merged[8];

// Merge the two arrays in sorted order

int i = 0, j = 0, k = 0;

while (i < n1 && j < n2) {

if (arr1[i] < arr2[j])

merged[k++] = arr1[i++];

else

merged[k++] = arr2[j++];

}

while (i < n1)

merged[k++] = arr1[i++];

while (j < n2)

merged[k++] = arr2[j++];

for (i = 0; i < n1 + n2; i++) {

cout << merged[i] << " ";

}

return 0;

}

What will be the output when the above code is executed?

a) 2 5 8 10 1 4 6 9

**b) 1 2 4 5 6 8 9 10**

c) 10 8 6 4 2 9 5 1

d) 1 4 6 9 2 5 8 10

6. The time complexity of the Insertion Sort algorithm for sorting an array of n elements is:

a) O(n)

b) O(log n)

c) O(n log n)

**d) O(n^2)**

7. #include <iostream>

using namespace std;

int main() {

int arr[] = {3, 8, 2, 1, 5};

int n = sizeof(arr) / sizeof(arr[0]);

int key = 4;

int index = -1;

for (int i = 0; i < n; i++) {

if (arr[i] == key) {

index = i;

break;

}

}

cout << index;

return 0;

}

What will be the output when the above code is executed with key = 4?

a) 1

b) 2

c) 4

**d) -1**

8. Binary Search is more efficient than Linear Search because:

a) Binary Search uses less memory.

b) Binary Search has a lower time complexity.

**c)Binary Search reduces the search space by half with each iteration.**

d) Binary Search is suitable for unsorted arrays.

9. #include <iostream>

using namespace std;

void print(int arr[], int n) {

for (int i = 0; i < n; i++) {

if (arr[i] % 2 == 0) {

cout << arr[i] << " ";

}

}

}

int main() {

int arr[] = {5, 2, 8, 3, 1};

int n = sizeof(arr) / sizeof(arr[0]);

print(arr, n);

return 0;

}

What will be the output when the above code is executed?

**a) 2 8**

b) 5 2 8

c) 2 8 6

d) 5

10. Which data structure can be used to efficiently implement a queue using an array?

a) One-dimensional array

**b) Circular array**

c) Two-dimensional array

d) Linked List

11. #include <iostream>

using namespace std;

int main() {

int arr[5] = {5, 2, 9, 1, 5};

int key = 9;

for (int i = 0; i < 5; i++) {

if (arr[i] == key) {

cout << "Found" << endl;

break;

}

}

return 0;

}

What will be the output of the above code if the key is 9?

**a) Found**

b) Not Found

c) Compiler error

d) Undefined behavior

12. Which operation is used to access all the elements of an array in sequence?

a) Insertion

b) Deletion

**c) Traversal**

d) Sorting

13. #include <iostream>

using namespace std;

int main() {

int arr[] = {2, 4, 6, 8, 10};

int n = sizeof(arr) / sizeof(arr[0]);

int sum = 0;

for (int i = 0; i < n; i++) {

sum += arr[i];

}

cout << sum << endl;

return 0;

}

What will be the output of the above code?

a) 20

**b) 30**

c) 15

d) 10

14. Which sorting algorithm has the best time complexity in the average case and requires additional space for merging?

a) Bubble Sort

b) Quick Sort

**c) Merge Sort**

d) Heap Sort

15. #include <iostream>

using namespace std;

int main() {

int arr[] = {5, 2, 9, 1, 5};

int n = sizeof(arr) / sizeof(arr[0]);

int min = arr[0], max = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i] < min) {

min = arr[i];

}

if (arr[i] > max) {

max = arr[i];

}

}

cout << min << " " << max << endl;

return 0;

}

What will be the output of the above code?

**a) 1 9**

b) 2 9

c) 1 5

d) 2 5

16. The process of arranging elements of an array in a specific order is called:

a) Traversal

**b) Sorting**

c) Insertion

d) Deletion

17. #include <iostream>

using namespace std;

int main() {

int arr[] = {1, 2, 3, 4, 5};

int\* ptr = arr;

for (int i = 0; i < 5; i++) {

cout << \*ptr++ << " ";

}

return 0;

}

What will be the output of the above code?

**a) 1 2 3 4 5**

b) 5 4 3 2 1

c) 5 5 5 5 5

d) Compiler error

18. The time complexity for merging two sorted arrays of size m and n into a third array of size m+n is:

**a) O(m + n)**

b) O(m \* n)

c) O(m log n)

d) O(log(m + n))

19. Which array operation requires shifting elements and has a worst-case time complexity of O(n)?

a) Traversal

**b) Deletion from the beginning**

c) Insertion at the end

d) Searching for an element

20. #include <iostream>

using namespace std;

int binarySearch(int arr[], int n, int key) {

int low = 0, high = n - 1, mid;

while (low <= high) {

mid = (low + high) / 2;

if (arr[mid] == key) {

return mid;

}

else if (arr[mid] < key) {

low = mid + 1;

}

else {

high = mid - 1;

}

}

return -1;

}

int main() {

int arr[] = {1, 2, 3, 4, 5};

int n = sizeof(arr) / sizeof(arr[0]);

int key = 3;

int result = binarySearch(arr, n, key);

cout << result << endl;

return 0; }

What will be the output of the above code if the key is 3?

a) 1

**b) 2**

c) 3

d) 4