

C MULTI CHOICE ANSWERS AND EXPLANATIONS

1. Answer C
2. Answer C
3. Answer: A
Explanation: The assignment `x = 0` always evaluates to true (non-zero), making the condition true.
4. Answer: B, trying to access **index** 5 instead of **element** 5.
5. Answer: C
6. Answer: A
7. Answer: A
Explanation: Array indices in C start at 0, so `arr[i]` is out-of-bounds when `i = 5`.
8. Answer: B
9. Answer: A
10. Answer: A
Explanation: The file is opened in read mode, but writing is attempted.

C++ MULTI CHOICE ANSWERS AND EXPLANATIONS

1. Answer: D
2. Answer: A
3. Answer: A
4. Answer: A
5. Answer: B
6. Answer: A
Explanation: The correct syntax is `delete[] ptr;` for arrays.
7. Answer: B
8. Answer: D
9. Answer: B `delete[] ptr;`
10. Answer: B

C CREATIVE SOLUTION

```
1 #include <stdio.h>
2
3 // Function to perform linear search
4 int linearSearch(int arr[], int size, int value) {
5     for (int i = 0; i < size; i++) {
6         if (arr[i] == value) {
7             return i; // Return index if found
8         }
9     }
10    return -1; // Return -1 if not found
11 }
12
13 // Function to find maximum and minimum in the array
14 void findMaxAndMin(int arr[], int size, int *max, int *min) {
15     *max = arr[0];
16     *min = arr[0];
17     for (int i = 1; i < size; i++) {
18         if (arr[i] > *max) {
19             *max = arr[i];
20         }
21         if (arr[i] < *min) {
22             *min = arr[i];
23         }
24     }
25 }
26
27 // Function to calculate sum and average
28 void calculateSumAndAverage(int arr[], int size, int *sum, float *average) {
29     *sum = 0;
30     for (int i = 0; i < size; i++) {
31         *sum += arr[i];
32     }
33     *average = (float)(*sum) / size;
34 }
35
36 int main() {
37     // Define and initialize the array
38     int arr[10] = {12, 45, 78, 34, 23, 56, 89, 90, 67, 33};
39     int size = 10;
40     int searchValue, index, max, min, sum;
41     float average;
42
43     // Print the array
44     printf("Array elements: ");
45     for (int i = 0; i < size; i++) {
46         printf("%d ", arr[i]);
47     }
48     printf("\n");
49
50     // Perform linear search
51     printf("Enter a value to search: ");
52     scanf("%d", &searchValue);
53     index = linearSearch(arr, size, searchValue);
54     if (index != -1) {
55         printf("Value found at index %d.\n", index);
56     } else {
57         printf("Value not found in the array.\n");
58     }
59
60     // Continued on the next page
61 }
```

```

62 // Find and print the maximum and minimum values
63 findMaxAndMin(arr, size, &max, &min);
64 printf("Maximum value: %d\n", max);
65 printf("Minimum value: %d\n", min);
66
67 // Calculate and print the sum and average
68 calculateSumAndAverage(arr, size, &sum, &average);
69 printf("Sum of array elements: %d\n", sum);
70 printf("Average of array elements: %.2f\n", average);
71
72 return 0;
73 }

```

Task	Marks	Explanation
Define and initialize the array	4	The array must contain the specified values and be printed correctly.
Implement <code>linearSearch</code> function	8	Function must iterate through the array and return the correct index or <code>-1</code> .
Use <code>linearSearch</code> in main and print results	8	User input must be handled, and results displayed correctly.
Implement <code>findMaxAndMin</code> function	6	Function must calculate and return the correct maximum and minimum values.
Implement sum and average calculation	4	Function must return and print the sum and average correctly.

Table 1: Marking scheme for the C program question

C++ CREATIVE SOLUTION

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 // Student class definition
6 class Student {
7 private:
8     string name;
9     float grades[5];
10    int rollNumber;
11
12 public:
13     // Constructor to initialize attributes
14     Student(string studentName, float studentGrades[], int studentRollNumber) {
15         name = studentName;
16         rollNumber = studentRollNumber;
17         for (int i = 0; i < 5; i++) {
18             grades[i] = studentGrades[i];
19         }
20     }
21
22     // Function to calculate average grade
23     float calculateAverage() {
24         float sum = 0.0;
25         for (int i = 0; i < 5; i++) {
26             sum += grades[i];
27         }
28         return sum / 5;
29     }
30
31     // Function to check pass/fail
32     bool isPass() {
33         return calculateAverage() >= 50;
34     }
35
36     // Function to display student details
37     void displayDetails() {
38         cout << "Name: " << name << endl;
39         cout << "Roll Number: " << rollNumber << endl;
40         cout << "Grades: ";
41         for (int i = 0; i < 5; i++) {
42             cout << grades[i] << " ";
43         }
44         cout << endl;
45         cout << "Average: " << calculateAverage() << endl;
46         cout << (isPass() ? "Status: Pass" : "Status: Fail") << endl;
47     }
48 };
49
50 int main() {
51     float grades1[5] = {70, 80, 90, 60, 50};
52     Student student1("John Doe", grades1, 101);
53
54     // Display details and status
55     student1.displayDetails();
56
57     return 0;
58 }
```

Task	Marks	Explanation
Define the class with required members	5	Class should include <code>name</code> , <code>grades</code> , <code>rollNumber</code> , and appropriate access.
Implement <code>calculateAverage</code> function	5	Function must calculate the average of the grades correctly.
Implement <code>isPass</code> function	5	Logic for checking if average is ≥ 50 should be correct.
Implement <code>displayDetails</code> function	5	Should display all required fields in a formatted manner.
Correctly instantiate objects in <code>main</code>	5	Objects must be created with provided details.
Call functions and test program	5	Program output should match the requirements.

Table 2: Marking scheme for the C++ class question