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

```c
#include <stdio.h>

// Function to perform linear search
int linearSearch(int arr[], int size, int value) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == value) {
            return i; // Return index if found
        }
    }
    return -1; // Return -1 if not found
}

// Function to find maximum and minimum in the array
void findMaxAndMin(int arr[], int size, int *max, int *min) {
    *max = arr[0];
    *min = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] > *max) {
            *max = arr[i];
        }
        if (arr[i] < *min) {
            *min = arr[i];
        }
    }
}

// Function to calculate sum and average
void calculateSumAndAverage(int arr[], int size, int *sum, float *average) {
    *sum = 0;
    for (int i = 0; i < size; i++) {
        *sum += arr[i];
    }
    *average = (float)(*sum) / size;
}

int main() {
    // Define and initialize the array
    int arr[10] = {12, 45, 78, 34, 23, 56, 89, 90, 67, 33};
    int size = 10;
    int searchValue, index, max, min, sum;
    float average;

    // Print the array
    printf("Array elements: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    // Perform linear search
    printf("Enter a value to search: ");
    scanf("%d", &searchValue);
    index = linearSearch(arr, size, searchValue);
    if (index != -1) {
        printf("Value found at index %d.\n", index);
    } else {
        printf("Value not found in the array.\n");
    }

    // Continued on the next page

```

```c
    // Find and print the maximum and minimum values
    findMaxAndMin(arr, size, &max, &min);
    printf("Maximum value: %d\n", max);
    printf("Minimum value: %d\n", min);

    // Calculate and print the sum and average
    calculateSumAndAverage(arr, size, &sum, &average);
    printf("Sum of array elements: %d\n", sum);
    printf("Average of array elements: %.2f\n", average);

    return 0;
}
```

Task	**Marks**	**Explanation**
Define and initialize the array	4	The array must contain the specified values and be printed correctly.
Implement `linearSearch` function	8	Function must iterate through the array and return the correct index or `-1`.
Use `linearSearch` in `main` and print results	8	User input must be handled, and results displayed correctly.
Implement `findMaxAndMin` 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

```cpp
#include <iostream>
#include <string>
using namespace std;

// Student class definition
class Student {
private:
    string name;
    float grades[5];
    int rollNumber;

public:
    // Constructor to initialize attributes
    Student(string studentName, float studentGrades[], int studentRollNumber) {
        name = studentName;
        rollNumber = studentRollNumber;
        for (int i = 0; i < 5; i++) {
            grades[i] = studentGrades[i];
        }
    }

    // Function to calculate average grade
    float calculateAverage() {
        float sum = 0.0;
        for (int i = 0; i < 5; i++) {
            sum += grades[i];
        }
        return sum / 5;
    }

    // Function to check pass/fail
    bool isPass() {
        return calculateAverage() >= 50;
    }

    // Function to display student details
    void displayDetails() {
        cout << "Name: " << name << endl;
        cout << "Roll Number: " << rollNumber << endl;
        cout << "Grades: ";
        for (int i = 0; i < 5; i++) {
            cout << grades[i] << " ";
        }
        cout << endl;
        cout << "Average: " << calculateAverage() << endl;
        cout << (isPass() ? "Status: Pass" : "Status: Fail") << endl;
    }
};

int main() {
    float grades1[5] = {70, 80, 90, 60, 50};
    Student student1("John Doe", grades1, 101);

    // Display details and status
    student1.displayDetails();

    return 0;
}
```

Task	Marks	Explanation
Define the class with required members	5	Class should include `name`, `grades`, `rollNumber`, and appropriate access.
Implement `calculateAverage` function	5	Function must calculate the average of the grades correctly.
Implement `isPass` function	5	Logic for checking if average is ≥ 50 should be correct.
Implement `displayDetails` function	5	Should display all required fields in a formatted manner.
Correctly instantiate objects in `main`	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