Let's convert the provided code snippets from C to Java and solve them. I'll address each question step-by-step.

Code Conversion to Java and Corrections

1(b) Find the error(s) in the code snippet and correct them

C Code

```
#include <stdio.h>
int main() {
   int x;
   printf("Enter a value for x: ");
   scanf("%d", x);
   if (x > 10)
      printf("x is greater than 10\n");
   else
      printf("x is less than or equal to 10\n");
   return 0;
}
```

Errors:

- scanf("%d", x); is incorrect. The address of x should be passed: scanf("%d", &x);.

Corrected C Code

```
#include <stdio.h>
int main() {
   int x;
   printf("Enter a value for x: ");
   scanf("%d", &x);
   if (x > 10)
      printf("x is greater than 10\n");
   else
      printf("x is less than or equal to 10\n");
   return 0;
}
```

Converted Java Code

```
import java.util.Scanner;


public class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a value for x: ");
      int x = scanner.nextInt();


      if (x > 10) {
         System.out.println("x is greater than 10");
      } else {
         System.out.println("x is less than or equal to 10");
      }
   }
}
```

1(e) Write a function in C called sumAndAverage that takes two integers and prints their sum and average

C Code

```
#include <stdio.h>
void sumAndAverage(int a, int b) {
   int sum = a + b;
   float avg = sum / 2.0;
   printf("Sum = %d, Average = %.2f\n", sum, avg);
}


int main() {
   sumAndAverage(10, 20);
   return 0;
}
```

Converted Java Code

```
public class Main {
   public static void sumAndAverage(int a, int b) {
      int sum = a + b;
      float avg = sum / 2.0f;
      System.out.printf("Sum = %d, Average = %.2f%n", sum, avg);
   }

   public static void main(String[] args) {
      sumAndAverage(10, 20);
   }
}
```

2(c) Identify and correct the errors in the code snippet

C Code

```
#include <stdio.h>
int main() {
   int i = 0;
   while (i < 5)
      printf("Value of i: %d\n", i);
   i++;
   return 0;
}
```

Errors:

- The i++ is outside the while loop, resulting in an infinite loop.

Corrected C Code

```
#include <stdio.h>
int main() {
   int i = 0;
   while (i < 5) {
      printf("Value of i: %d\n", i);
```

```
        i++;
    }
    return 0;
}
```

Converted Java Code

```
public class Main {
    public static void main(String[] args) {
        int i = 0;
        while (i < 5) {
            System.out.println("Value of i: " + i);
            i++;
        }
    }
}
```

3(c) Predict the output of the recursive function

C Code

```
#include <stdio.h>
int recursiveSum(int n) {
    if (n <= 0)
        return 0;
    else
        return n + recursiveSum(n - 1);
}

int main() {
    printf("%d\n", recursiveSum(4));
    return 0;
}
```

Converted Java Code

```
public class Main {
```

```
    public static int recursiveSum(int n) {
        if (n <= 0) {
            return 0;
        } else {
            return n + recursiveSum(n - 1);
        }
    }

    public static void main(String[] args) {
        System.out.println(recursiveSum(4));
    }
}
```

Output: The recursive function calculates 4+3+2+1=10

Sure, I'll address the requested code snippets, solve them, and provide equivalent Java code. Let's start with the tasks. Let me know if you'd like me to focus on specific ones. I'll break them down below.

Question 2.a:

C Code with Errors (Correction):

```
#include <stdio.h>
int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    for (int i = 0; i < 5; i++) {
        printf("%d ", arr[i]); // Fixed 'arr(i)' to 'arr[i]'
    }
    return 0;
}
```

Converted Java Code:

```
public class ArrayExample {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5};
```

```
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

Question 2.d:

C Code Output Prediction and Conversion:

```
#include <stdio.h>
int main() {
    int x = 5;
    while (x > 0) {
        printf("%d ", x);
        x--;
    }
    return 0;
}
```

Output: 5 4 3 2 1

Converted Java Code:

```
public class Countdown {
    public static void main(String[] args) {
        int x = 5;
        while (x > 0) {
            System.out.print(x + " ");
            x--;
        }
    }
}
```

Question 2.f:

Function to Find Minimum (C Code):

```
int findMinimum(int arr[], int size) {
    int min = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] < min) {
            min = arr[i];
        }
    }
    return min;
}
```

Converted Java Code:

```
public class FindMinimum {
    public static int findMinimum(int[] arr) {
        int min = arr[0];
        for (int i = 1; i < arr.length; i++) {
            if (arr[i] < min) {
                min = arr[i];
            }
        }
        return min;
    }

    public static void main(String[] args) {
        int[] arr = {4, 2, 7, 1, 9};
        System.out.println("Minimum value: " + findMinimum(arr));
    }
}
```

Question 3.c:

C Code with Errors (Reversing a String):

```
#include <stdio.h>
#include <string.h>
```

```
void reverseString(char *str) {
    int len = strlen(str);
    for (int i = 0; i < len / 2; i++) {
        char temp = str[i];
        str[i] = str[len - i - 1];
        str[len - i - 1] = temp;
    }
}

int main() {
    char str[] = "Hello";
    reverseString(str);
    printf("%s\n", str);
    return 0;
}
```

Converted Java Code:

```
public class ReverseString {
    public static void reverseString(char[] str) {
        int len = str.length;
        for (int i = 0; i < len / 2; i++) {
            char temp = str[i];
            str[i] = str[len - i - 1];
            str[len - i - 1] = temp;
        }
    }

    public static void main(String[] args) {
        char[] str = "Hello".toCharArray();
        reverseString(str);
        System.out.println(new String(str));
```

```
    }
}
```

Let me know which specific task you'd like me to prioritize or if you'd like me to clarify any of these!