Recursion Tracing Exercises with Answers

**Question 1:  Can you Predict the Output of following Code?**

public void traceRecursion(int n) {  
    if (n <= 1)   
        System.out.print(n);  
    else{   
        traceRecursion(n / 2);  
        System.out.print(", " + n);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(1)  
2. traceRecursion(4)  
3. traceRecursion(16)  
4. traceRecursion(30)  
5. traceRecursion(100)  
  
  
**Answer**  
1. 1  
2. 1, 2, 4  
3. 1, 2, 4, 8, 16  
4. 1, 3, 7, 15, 30  
5. 1, 3, 6, 12, 25, 50, 100

**Question 2:  Can you Predict the Output of following Code?**

public void traceRecursion(int n) {  
    if (n > 100)  
        System.out.print(n);  
    else {  
        traceRecursion(2 \* n);  
        System.out.print(", " + n);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(10)  
2. traceRecursion(30)  
3. traceRecursion(42)  
4. traceRecursion(70)  
5. traceRecursion(113)  
  
**Answer**  
1. 160, 80, 40, 20, 10  
2. 120, 60, 30  
3. 168, 84, 42  
4. 140, 70  
5. 113

**Question 3:  Can you Predict the Output of following Code?**

public void traceRecursion(int n) {  
    if (n <= 0) {  
        System.out.print("\*");  
    } else if (n % 2 == 0) {  
        System.out.print("(");  
        traceRecursion(n - 1);  
        System.out.print(")");  
    } else {  
        System.out.print("[");  
        traceRecursion(n - 1);  
        System.out.print("]");  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(0)  
2. traceRecursion(1)  
3. traceRecursion(2)  
4. traceRecursion(4)  
5. traceRecursion(5)  
  
  
**Answer**  
1. \*  
2. [\*]  
3. ([\*])  
4. ([([\*])])  
5. [([([\*])])]

**Question 4:  Can you Predict the Output of following Code?**

public void traceRecursion(int x, int y) {  
    if (y == 1) {  
        System.out.print(x);  
    } else {  
        System.out.print(x \* y + ", ");  
        traceRecursion(x, y - 1);  
        System.out.print(", " + x \* y);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(4, 1)  
2. traceRecursion(4, 2)  
3. traceRecursion(8, 2)  
4. traceRecursion(4, 3)  
5. traceRecursion(3, 4)  
  
  
**Answer**  
1. 4  
2. 8, 4, 8  
3. 16, 8, 16  
4. 12, 8, 4, 8, 12  
5. 12, 9, 6, 3, 6, 9, 12

**Question 5:  Can you Predict the Return Value of following Code?**

public int traceRecursion(int x, int y) {  
    if (x < y)   
        return x;  
    return traceRecursion(x - y, y);  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(6, 13)  
2. traceRecursion(14, 10)  
3. traceRecursion(37, 10)  
4. traceRecursion(8, 2)  
5. traceRecursion(50, 7)  
  
  
**Answer**  
1. 6  
2. 4  
3. 7  
4. 0  
5. 1

**Question 6:  Can you Predict the Return Value of following Code?**

public int traceRecursion(int x, int y) {  
    if (x < 0) {  
        return -traceRecursion(-x, y);  
    } else if (y < 0) {  
        return -traceRecursion(x, -y);  
    } else if (x == 0 && y == 0) {  
        return 0;  
    } else {  
        return 100 \* traceRecursion(x / 10, y / 10) + 10 \* (x % 10) + y % 10;  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(5, 7)  
2. traceRecursion(12, 9)  
3. traceRecursion(-7, 4)  
4. traceRecursion(-23, -48)  
5. traceRecursion(128, 343)  
  
  
**Answer**  
1. 57  
2. 1029  
3. -74  
4. 2438  
5. 132483

**Question 7:  Can you Predict the Return Value of following Code?**

public static int traceRecursion(int n, int k) {  
    if (k == 0 || k == n)  
        return 1;  
    else if (k > n)  
        return 0;  
    else  
        return traceRecursion(n - 1, k - 1) + traceRecursion(n - 1, k);  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(7, 1)  
2. traceRecursion(4, 2)  
3. traceRecursion(4, 3)  
4. traceRecursion(5, 3)      
5. traceRecursion(5, 4)  
  
**Answer**  
1. 7  
2. 6  
3. 4  
4. 10  
5. 5

**Question 8:  Can you Predict the Return Value of following Code?**

public int traceRecursion(int n) {  
    if (n < 0) {  
        return -traceRecursion(-n);  
    } else if (n < 10) {  
        return n;  
    } else {  
        return traceRecursion(n / 10 + n % 10);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(6)  
2. traceRecursion(17)  
3. traceRecursion(259)  
4. traceRecursion(977)      
5. traceRecursion(-479)  
  
**Answer**  
1. 6  
2. 8  
3. 7  
4. 5  
5. -2

**Question 9:  Can you Predict the Return Value of following Code?**

public int traceRecursion(int n) {  
    if (n < 0)  
        return traceRecursion(-n);  
    else if (n < 10)  
        return n;  
    else  
        return n % 10 + traceRecursion(n / 10);  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(8)  
2. traceRecursion(74)  
3. traceRecursion(-52)  
4. traceRecursion(3052)      
5. traceRecursion(82534)

**Answer**  
1. 8  
2. 11  
3. 7  
4. 10  
5. 22

**Question 10:  Can you Predict the Return Value of following Code?**

public void traceRecursion(int x) {  
    if (x < 10)  
        System.out.print(x);  
    else {  
        int y = x % 10;  
        System.out.print(y);  
        traceRecursion(x / 10);  
        System.out.print(y);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(7)  
2. traceRecursion(38)  
3. traceRecursion(194)  
4. traceRecursion(782)  
5. traceRecursion(3842)  
6. traceRecursion(3)  
7. traceRecursion(42)  
8. traceRecursion(293)  
9. traceRecursion(402)  
10. traceRecursion(2468)

**Answer**  
1. 7  
2. 838  
3. 49194  
4. 28782  
5. 2483842  
6. 3  
7. 242  
8. 39293  
9. 20402  
10. 8642468

**Question 11:  Can you Predict the Return Value of following Code?  [SKIP IT - PROBLEM HAS SOME ISSUE]**

public int traceRecursion(int n){  
    if(n<0)  
        return -traceRecursion(-n);  
    else if(n>10)  
        return (n+1)%10;  
    else  
        return 10 \*traceRecursion(n/10) + (n+1) %10;  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(7)      
2. traceRecursion(42)      
3. traceRecursion(385)      
4. traceRecursion(-790)      
5. traceRecursion(89294)

**Answer**  
1. 8  
2. 3  
3. 6  
4. -1  
5. 5

**Question 12:  Can you Predict the Output of following Code?**

public void traceRecursion(int x, int y){  
    if(x > y)  
        System.out.print("\*");  
    else if(x == y)  
        System.out.print("="+ y + "=");  
    else{  
        System.out.print(y + " ");  
        traceRecursion(x+1, y-1);  
        System.out.print(" " + x);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(3, 3)      
2. traceRecursion(5, 1)      
3. traceRecursion(1, 5)      
4. traceRecursion(2, 7)      
5. traceRecursion(1, 8)

**Answer**  
1. =3=  
2. \*  
3. 5 4 =3= 2 1  
4. 7 6 5 \* 4 3 2  
5. 8 7 6 5 \* 4 3 2 1

**Question 13:  Can you Predict the Output of following Code?**

public void traceRecursion(int n){  
    if(n<0){  
        System.out.print("-");  
        traceRecursion(-n);  
    }  
    else if(n<10)  
        System.out.print(n);  
    else{  
        int two = n%100;  
        System.out.print(two/10);  
        System.out.print(two%10);  
        traceRecursion(n/100);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(7)      
2. traceRecursion(825)      
3. traceRecursion(38947)      
4. traceRecursion(612305)      
5. traceRecursion(-12345678)

**Answer**  
1. 7  
2. 258  
3. 47893  
4. 0523610  
5. -785634120

**Question 14:  Can you Predict the Output of following Code?**

public int traceRecursion(int n){  
    if(n<0)  
        return -traceRecursion(-n);  
    else if(n==0)  
        return 0;  
    else  
        return traceRecursion(n/10) \* 10 + 9 - (n % 10);  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(7)      
2. traceRecursion(825)      
3. traceRecursion(38947)      
4. traceRecursion(612305)      
5. traceRecursion(-12345678)

**Answer**  
1. 2  
2. 174  
3. 61052  
4. 387694  
5. -87654321

**Question 15:  Can you Predict the Output of following Code?**

public void traceRecursion(int n){  
    if(n%2==1)  
        System.out.print(n);  
    else{  
        System.out.print(n + ", ");  
        traceRecursion(n/2);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(13)      
2. traceRecursion(42)      
3. traceRecursion(40)      
4. traceRecursion(60)      
5. traceRecursion(48)

**Answer**  
1. 13  
2. 42, 21  
3. 40, 20, 10, 5  
4. 60, 30, 15  
5. 48, 24, 12, 6, 3

**Question 16:  Can you Predict the Output of following Code?**

public void traceRecursion(int n){  
    if(n<=1)  
        System.out.print(": ");  
    else{  
        System.out.print((n % 2) + " ");  
        traceRecursion(n/2);  
        System.out.print(n + " ");  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(8)      
2. traceRecursion(25)      
3. traceRecursion(46)

**Answer**  
1. 0 0 0 : 2 4 8  
2. 1 0 0 1 : 3 6 12 25  
3. 0 1 1 1 0 : 2 5 11 23 46

**Question 17:  Can you Predict the Output of following Code?**

public void traceRecursion(int x, int y){  
    if(y<=0)  
        System.out.print("0 ");  
    else if(x > y){  
        System.out.print(x + " ");  
        traceRecursion(x - y, y);  
    }  
    else{  
        traceRecursion(x, y - x);  
        System.out.print(y + " ");  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(6, 3)      
2. traceRecursion(2, 3)      
3. traceRecursion(5, 8)      
4. traceRecursion(21, 12)      
5. traceRecursion(3, 10)

**Answer**  
1. 6 0 3  
2. 2 0 1 3  
3. 5 2 0 1 3 8  
4. 21 9 6 0 3 12  
5. 3 2 0 1 4 7 10

**Question 18:  Can you Predict the Output of following Code?**

public static int traceRecursion(int x, int y) {  
    if (x < 0)  
        return -traceRecursion(-x, y);  
    else if (y < 0)  
        return -traceRecursion(x, -y);  
    else if (y < x)  
        return 0;  
    else  
        return 1 + traceRecursion(x, y - x);  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(10, 28)      
2. traceRecursion(5, 17)      
3. traceRecursion(2, 10)      
4. traceRecursion(4, -15)      
5. traceRecursion(-3, -23)

**Answer**  
1. 2  
2. 3  
3. 5  
4. -3  
5. 7

**Question 19:  Can you Predict the Output of following Code?**

public int traceRecursion(int x, int y) {  
    if (x > y)  
        return 0;  
    else  
        return traceRecursion(x + 1, y) + 2 \* x - 1;  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(1. 3)      
2. traceRecursion(4, 4)      
3. traceRecursion(3, 5)      
4. traceRecursion(1, 5)      
5. traceRecursion(4, 7)

**Answer**  
1. 9  
2. 7  
3. 21  
4. 25  
5. 40

**Question 20:  Can you Predict the Output of following Code?**

public void traceRecursion(int n) {  
    if (n == 1)  
        System.out.print(n);  
    else {  
        System.out.print(n + ", ");  
        if (n % 2 == 0)  
            traceRecursion(n / 2);  
        else  
            traceRecursion(3 \* n + 1);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(1)      
2. traceRecursion(3)      
3. traceRecursion(4)      
4. traceRecursion(5)      
5. traceRecursion(20)

**Answer**  
1. 1  
2. 3, 10, 5, 16, 8, 4, 2, 1  
3. 4, 2, 1  
4. 5, 16, 8, 4, 2, 1  
5. 20, 10, 5, 16, 8, 4, 2, 1

**Question 21:  Can you Predict the Output of following Code?**

public int traceRecursion(int x, int y) {  
    if (x % 2 == 1 || y % 2 == 1)   
        return 1;  
    else  
        return 2 \* traceRecursion(x / 2, y / 2);  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(4, 19)      
2. traceRecursion(32, 56)      
3. traceRecursion(12, 20)      
4. traceRecursion(4, 18)      
5. traceRecursion(48, 128)

**Answer**  
1. 1  
2. 8  
3. 4  
4. 2  
5. 16

**Question 22:  Can you Predict the Output of following Code?**

public void traceRecursion(int n) {  
    if (n >= 10)  
        traceRecursion(n / 10);  
    if (n % 2 == 0)  
        System.out.print("-");  
    else  
        System.out.print("\*");  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(5)      
2. traceRecursion(15)      
3. traceRecursion(304)      
4. traceRecursion(9247)      
5. traceRecursion(43269)

**Answer**

1. \*  
2. \*\*  
3. \*--  
4. \*--\*  
5. -\*--\*

**Question 23:  Can you Predict the Output of following Code?**

public int traceRecursion(int n, int m) {  
    if (n == 0 || m == 0)  
        return 0;  
    else if (n % 10 == m % 10)  
        return 1 + traceRecursion(n / 10, m / 10);  
    else  
        return traceRecursion(n / 10, m / 10);  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(18, 0)      
2. traceRecursion(8, 18)      
3. traceRecursion(25, 21)      
4. traceRecursion(305, 315)      
5. traceRecursion(20734, 1724)      
  
**Answer**  
1. 0  
2. 1  
3. 1  
4. 2  
5. 2

**Question 24:  Can you Predict the Output of following Code?**

public void traceRecursion(int n) {  
    if (n < 0)  
        traceRecursion(-n);  
    else if (n < 10)  
        System.out.print(n);  
    else {  
        traceRecursion(n / 10);  
        int digit = n % 10;  
        System.out.print(", " + digit % 3);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(7)      
2. traceRecursion(38)      
3. traceRecursion(749)      
4. traceRecursion(-6842)      
5. traceRecursion(954023)  
  
**Answer**  
1. 7  
2. 3, 2  
3. 7, 1, 0  
4. 6, 2, 1, 2  
5. 9, 2, 1, 0, 2, 0

**Question 25:  Can you Predict the Output of following Code?**

public void traceRecursion(int n) {  
    if (n <= 1)  
        System.out.print(n);  
    else {  
        System.out.print(n + ", ");  
        traceRecursion(n / 2);  
        System.out.print(", " + n);  
    }  
}

**What will be the output of above code with following calls?**  
1. traceRecursion(1)  
2. traceRecursion(3)  
3. traceRecursion(4)  
4. traceRecursion(6)  
5. traceRecursion(12)

**Answer**  
1. 1  
2. 3, 1, 3  
3. 4, 2, 1, 2, 4  
4. 6, 3, 1, 3, 6  
5. 12, 6, 3, 1, 3, 6, 12

**Question 26:  Can you Predict the Output of following Code?**

int traceRecursion(int n)   
{   
    if (n <= 1)   
        return 1;   
    
    int ans = 0;   
    for (int i=0; i<n; i++)   
        ans += traceRecursion(i) \* traceRecursion(n-i-1);   
    
    return ans;   
}

**What will be the output of above code with following calls?**  
2. traceRecursion(3)  
3. traceRecursion(4)  
4. traceRecursion(6)

traceRecursion(10)

**Answer**  
5

14

132

16796