1. What does the following code print when called with i=1234, j=5?

public int compute(int i, int j) {

// base case

if (i < j) return i;

// recursive case

return compute(i - j, j);

}

1. What does the following code below print when called with value = 10 ?

public static int compute(int value) {

// base case

if (value == 0) return 0;

// compute term

int term = value ;

// recursive case

return term + compute(value - 2);

}

1. What does the following code below print when called with value = 9 ?

public static int compute(int value) {

// base case

if (value == 0) return 0;

// compute term

int term = value ;

// recursive case

return term + compute(value - 2);

}

1. What does the following code below print when called with d = 100.0 and n = 6?

public static double compute(double d, int n) {

// base case

if (n == 0) return 0.0;

// compute term

double term = d;

// recursive call

return term + compute(d / 10.0, n - 1);

}

1. What does the following code below print when called with number = 13 and base = 2?

public static void convert(int number, int base) {

int remainder = number % base;

int quotient = number / base;

if (quotient > 0) convert(quotient, base);

System.out.print(remainder);

}

1. What does the following code below print when called with s = ”aabbccddeeff”?

public static String munge(String s) {

if (s == null || s.length() <= 1) // base case

return s;

else if (s.charAt(0) == s.charAt(1))

return munge(s.substring(1, s.length()));

else

return s.charAt(0) + munge(s.substring(1, s.length()));

}

1. What does the following code print when i = 1?

public static void rec1 (int i) {

if (i==0)

System.*out.print(i + " ");*

for (int j=0; j<2; j++) {

*rec1(i-1);*

*rec1(i-1);*

}

}

1. What does the following code print when i = 1?

public static void rec2 (int i) {

if (i==0) {

System.*out.print(i + " ");*

} else {

for (int j=0; j<2; j++) {

*rec2 (i-1);*

*rec2 (i-1);*

}

}

}

1. What does rec return when list = {1,3,5,7,9} ?

public int rec(int [] list){

return rec3(list, 0);

}

public int rec3(int [] list, int start){

if (start == list.length - 1) {

return list[start];

} else {

return Math.max(list[start], rec3(list, start + 1));

}

}