Chapter 8 – AP - Recursion

Chapter Objectives

- Explain the underlying ideas of recursion.
- Examine recursive methods and processing steps.
- Define infinite recursion and discuss ways to avoid it.
- Explain when recursion should and should not be used.
- Demonstrate the use of recursion to solve problems.
  Examine the use of recursion in sorting.

Chapter Objectives: Recursion is a valuable tool in designing algorithms for both simple data structures and calculations as well as more advanced data types. This chapter covers recursion through examples of some classic problems.

Multiple Choice: 8.1 – 8.8, 8.10

True/False: 8.1 – 8.10

Short Answer: 8.2 – 8.4, 8.6 – 8.7

AP Multiple Choice: 8.1 – 8.6

AP Style Free-Response: 8.1

Worksheets: Recursion, Merge Sort, Quick Sort, Review of all sorts

Programming Projects: 8.2, QuickSort into Strings

fact(4) = ???

```
public int  fact (int  n)
{
     if (1 = = n)
       return  1;
     else
       return  n * fact (n - 1);
}
```

Name_____

Recursion

Find the output to the following code segments.

1.
```
int result = identity(10);

System.out.println("The final answer is " + result);

public int identity(int num)
{
   if (num < 1)
            return 10;
   else
            return num + identity(num - 2);
}
```

2.
```
int result2 = negative(-3);

System.out.println("The final answer is " + result2);

public int negative(int num)
{
   if (num >= 20)
      return -5;
   else
      return negative(num + 4) + 2 * num;
}
```

3.
```
int result3 = product(1);

System.out.println("The final answer is " + result3);

public int product(int num)
{
   if (num > 20)
      return -1;
   else
      return num * product(-2 * num);
}
```


4. What does mystery(4) print?

```
public void mystery (int x)
{
if (x <= 0)
  return;
else
 {
  System.out.print( x + " ");
  mystery( x - 2);
 }
}
```

5. What does numbuh(4) print?

```
public void numbuh (int x)
{
if (x <= 0)
 return;
else
 {
  numbuh (x - 2);
  System.out.print(x + " ");}
 }
```

6. What does value(4) return?

```
public int value (int x)
{
if (x <= 0)
  return 1;
else
  return x * value(x- 1);
}
```


7. What does strange(3543) return?

```
public int strange (int x)
{
if (x == 0)
  return 0;
else
  return ((x % 10) + strange(x/10));
}
```

8. What does whatisit(6) return?

```
public int whatisit(int x)
{
if (x == 1)
  return 2;
else
  return 2 * whatisit(x - 1);
}
```

9. What does nowwhat(6) return?

```
public int nowwhat(int x)
{
if (x == 0)
  return 0;
else
  return x + nowwhat ( x/2) + nowwhat ( x/4);
}
```


Name:_____

Merge Sort

1. Sort the following list using Merge Sort:

32 8 51 86 22 1 -3 5 -26

2. Practice sorting the following data using Merge Sort:

order the list you are currently reading using merge sort

Name_____

QUICKSORT

1. Practice sorting the following data using Quicksort.

28	42	69	8	100	33	36	3	84	21	40	5

2. Practice sorting the following data using Quicksort.

sort these words using quicksort until a recursive call is made

Name:_____

Review of Sorting Techniques

Sort the following list using each of the four techniques learned in class

45 12 31 76 33 55 72 24 6

Selection Sort:

Quick Sort:

Insertion Sort:

Merge Sort:

