AP Computer Science Prep - Question Catagory - loop

Catagory	Questions in 40	Percentage
datatype	2	5%
operator	3	7.5
if-else	5	12.5%
loop	16	40%
method	14	35%

Questions related to **loop** can be divided by

- for loop
- while loop
- embedded loop
- loop array
- loop multiple-dimension array
- loop ArrayList

1. Choose the correct option to complete lines 3 and 4 such that str2 will contain the letters of str1 in reverse order.

```
String str1 = "banana";
String str2 = "";
// 3. missing code
// 4. missing code
{
    str2 += str1.substring(i, i + 1);
    i--;
}
System.out.println(str2);
```

```
(A) int i = 0;
    while(i<str1.length);
(B) int i = str1.length();
    while (i >=0);
(C) int i = str1.length() - 1;
    while (i >=0);
(D) int i = str1.length();
    while (i>0);
(E) int i = str1.length() - 1;
    while (i>0)
```

2. Consider the following code excerpt:

```
int n = // some integer greater than zero
int count = 0;
int p = 0;
int q = 0;
for (p=1; p<n; p++)
    for(q=1; q<=n; q++)
        count++;
System.out.println(count);
```

(A) n^n

(B) n^2-1

(C) $(n-1)^2$

(D) $n(n-1)$

(E) n^2

3. Given the following code excerpt, determine the output:

```
int x = 0;
for (int j = 1; j < 4; j++) {
    if (x != 0 && j / x > 0)
        System.out.print(j / x + " ");
    else
        System.out.print(j * x + " ");
}
```

```
(A) 0 0 0
(B) 0 0 0 0
(C) 1 2 3
(D) 1 0 2 0 3 0
(E) ArithmeticException: Divide by zero
```

4. Consider the following code:

```
String space = " ";
String symbol = "*";
int num = 5;
for (int i = 1; i <= num; i++) {
    System.out.print(symbol);
}
System.out.println();
```

```
    for (int i = 1; i <= num; i++) {
        for (int j = num - i; j > 0; j--) {
            System.out.print(space);
        }
        System.out.println(symbol);
    }
    for (int i = 1; i <= num; i++) {
        System.out.print(symbol);
    }
```

```
(A) *****
    ****
    ***
    **
    *
    *****
(B) *****
    ****
    ***
    **
        *
    *****
(C) *****
        *
    *
    *
    *
    *****
(D) *****
    *
    *
    *
    *
    *****
(E) *****
    *
    **
    ***
    ****
    *****
```

5. What will be printed as a result of the following code excerpt?

```
    int sum = 0;
    for (int i=1; i<2; i++)
        for(int j=1; j<=3; j++)
            for(int k=1; k<4; k++)
                sum += (i*j*k);
    System.out.println(sum);
```

```
(A) 18
(B) 36
(C) 45
(D) 60
(E) 108
```

6. Consider the following code:

```
    int j = 0;
    String s = "map";
    while (j < s.length()) {
        int k = s.length();
        while (k > j) {
            System.out.println(s.substring(j, k));
            k--;
        }
        j++;
    }
```

```
(A) map
    ma
    m
    ap
    a
(B) map
    ma
    m
    ap
    a
    p
(C) map
    ap
    p
    ap
    p
    p
(D) m
    ma
    map
    a
    ap
    p
(E) p
    ap
    p
    map
```

```
ma
m
```

7. A factorial is shown by an exclamation point(!) following a number. The factorial of 5 or 5! is calculated by (5)(4)(3)(2)(1)=120.

   Assuming n is an integer greater than 1. Choose the method that will return n!

```
I.  public static int f(int n) {
        int factorial = 1;
        for (int i = n; i > 0; i--) {
            factorial *= n;
        }
        return factorial;
    }

II. public static int f(int n) {
        int factorial = 1;
        int j = 1;
        while (j <= n) {
            factorial *= j;
            j++;
        }
        return factorial;
    }

III.public static int f(int n) {
        if (n == 1)
            return n;
        return n * f(n - 1);
    }
```

```
(A) I only
(B) II only
(C) II only
(D) II and III only
(E) I, II and III
```

8. Given the following code excerpt:

```
int[] nums = {11, 22, 33, 44, 55, 66};
for (int i=0; i<nums.length; i++)
    nums[nums[i]/11] = nums[i];
```

```
(A) 1, 2, 3, 4, 5
(B) 11, 11, 33, 33, 55, 55
(C) 11, 11, 22, 33, 44, 55
(D) 11, 22, 22, 33, 33, 55
(E) 11, 22, 33, 44, 55, 66
```

9. Given the following code excerpt:

```
    int[] arr1 = {1, 2, 3, 4, 5, 6};
    int[] arr2 = arr1;
    int last = arr1.length - 1;

    for(int i=0; i<arr1.length; i++)
        arr2[i] = arr1[last-i];

    for(int i=0; i<arr1.length; i++)
        System.out.print(arr1[i] + " ");

    System.out.println(" ");

    for (int i=0; i<arr2.length; i++)
        System.out.print(arr2[i] + " ");
```

```
(A) 1, 2, 3, 4, 5, 6
    1, 2, 3, 4, 5, 6
(B) 1, 2, 3, 4, 5, 6
    6, 5, 4, 4, 5, 6
(C) 6, 5, 4, 3, 2, 1
    6, 5, 4, 4, 5, 6
(D) 6, 5, 4, 4, 5, 6
    1, 2, 3, 4, 5, 6
(E) 6, 5, 4, 4, 5, 6
    6, 5, 4, 4, 5, 6
```

10. Given the following code excerpt:

```
    int[] arr3 = {1, 2, 3, 4, 5, 6};

    for (int element : arr3) {
        element *=2;
        System.out.print(element + " ");
    }
    System.out.println("");

    for(int element: arr3)
```

```
        System.out.print(element + " ");
}
```

```
(A) 1, 2, 3, 4, 5, 6
    1, 2, 3, 4, 5, 6
(B) 2, 4, 6, 8, 10, 12
    1, 2, 3, 4, 5, 6
(C) 2, 4, 6, 8, 10, 12
    2, 4, 6, 8, 10, 12
(D) A compiler error will occur;
(E) A run-time exception will occur;
```

11. Given an array numbers containing a variety of integers and the following code excerpt:

```
    int holdSmallest = Integer.MAX_VALUE;
    int holdLargest = 0;
    int a = 0;
    int b = 0;
    for(int i=0; i<numbers.length; i++) {
        if (numbers[i] <= holdSmallest) {
            holdSmallest = numbers[i];
            a = i;
        }
        if(numbers[i] >= holdLargest) {
            holdLargest = numbers[i];
            b = i;
        }
    }
    System.out.println(a + " " + b);
```

Determine the statement below that reflects the most successful outcome.

(A) The code will print the smallest and largest values in the numbers array.

(B) The code will print the locations of the smallest and largest values in the numbers array.

(C) The code will print the locations of the smallest and largest non-negative values in the numbers array.

(D) The code will print the location of the smallest value in the numbers array and the largest non-negative value in the numbers array.

(E) The code will print the location of the smallest non-negative value in the numbers array and the largest value in the numbers array.

12. Given the following code excerpt:

```
    ArrayList<Integer> alist1 = new ArrayList<Integer>();
    int[] a1 = { 2, 4, 6, 7, 8, 10, 11 };
    for (int a : a1) {
        alist1.add(a);
    }
    for (int i = 0; i < alist1.size(); i++) {
        if (alist1.get(i) % 2 == 0) {
            alist1.remove(i); // dangerous
        }
    }
    System.out.println(alist1);
```

```
(A) [4, 7, 10, 11]
(B) [2, 4, 7, 10, 11]
(C) [2, 7, 10, 11]
(D) [7, 11]
(E) An IndenxOutOfBoundsException will occur
```

Question 29-30 refer to the following code expcerpt.

```
2   ArrayList<Integer> alist5 = new ArrayList<Integer>();
3   int[] a1 = { 21, 6, 2, 8, 1 };
4   for (int a : a1)
5   {
6       alist5.add(a);
7   }
8   for (int k = 0; k < alist5.size() - 1; k++)
9   {
10      for (int i = 0; i < alist5.size() - 2; i++)
11  {
12          if (alist5.get(i) > alist5.get(i + 1))
13          {
14              int hold = alist5.remove(i);
15              alist5.add(i + 1, hold);
16          }
17      }
18  }
19  System.out.println(alist5);
```

13. How many times will line 12 be executed?

```
(A) 6 times
(B) 12 times
(C) 15 times
(D) 16 times
(E) 20 times
```

14. What will be the final output after the code executes?

```
(A) [21, 8, 6, 2, 1]
(B) [6, 21, 2, 8, 1]
(C) [6, 2, 8, 21, 1]
(D) [2, 6, 8, 21, 1]
(E) [1, 2, 6, 8, 21]
```

15. Given nums—a rectanglular, but not necessarily square, two-dimensional array of integers, choose the code to correctly print the array:

```
    int[][] arr2d = { { 1, 2, 3, 4 }, { 5, 6, 7, 8 } };
    String s = "";
    for (int a = 0; a < arr2d[0].length; a++) {
        for (int b = 0; b < arr2d.length; b++) {
            s += arr2d[b][a] + " ";
        }
        s += "\n";
    }
    System.out.print(s);
```

Determine the resulting output.

```
(A) 1  2 3 4
    5 6 7 8
(B) 1 5 2 6
    3 7 4 8
(C) 1 2
    3 4
    5 6
    7 8
(D) 1 5
    2 6
    3 7
    4 8
(E) 1
    2
    3
    4
    5
    6
    7
    8
```

16. Given nums—a rectangular, two-dimensional array of integers, choose the code to print the entire array.

```
    // I.
    for (int r = 0; r < nums.length; r++) {
        for (int c = 0; c < nums[0].length; c++) {
            System.out.print(nums[r][c] +" ");
        }
        System.out.print("\n") ;
    }

    System.out.println();
    // II.
    for (int[] row : nums) {
        for (int col: row) {
            System.out.print(col +" ");
        }
        System.out.print("") ;
    }
    System.out.println();

    // III.
    for (int r = 0; r < nums[0].length; r++) {
        for (int c = 0; c < nums.length; c++) {
            System.out.print(nums[r][c] +" ");
        }
        System.out.print("\n") ;
    }
```

```
(A) I only
(B) I and II only
(C) I and III only
(D)II and III only
(E) I, II, and III
```

17. Consider the following code segment:

```
for (int i = 200 ; i > 0; i /= 3) {
    if (i % 2 == 0 )
        System.out.print(i +" ");
}
```

What is the output as a result of executing the code segment?

```
(A) 200 66 22 7 2
(B) 66 22 72
```

```
(C) 200 66 22 2
(D) 200 66 22
(E) 7
```

18. Consider the following code segment.

```
int vail = 2, val2 = 22, val3 = 78;
while (val2 % vail == 0 || val2 % 3 == 0 ){
    val3++;
    val2--;
}
```

What will val3 contain after the code segment is executed?

```
(A) 77
(B) 78
(C) 79
(D) 80
(E) None of the above
```

19. What will be the output when the following code is evaluated?

```
for (int k = 0; k< 3; k++ ) {
    for (int j = 1; j <4; j++){
        System.out.printin (j + " ");
    }
    System,out.println();
}
```

```
(A) 1 2 3 4
    1 2 3 4
    1 2 3 4
(B) 0 1 2
    0 1 2
    0 1 2
    0 1 2
(C) 1 2 3
    1 2 3
    1 2 3
(D) 1 2 3
    1 2 3
    1 2 3
    1 2 3
(E) 1 2 3 4
    1 2 3 4
```

```
1 2 3 4
1 2 3 4
```