

1. _____ Refer to these declarations:

```
Integer k = new Integer(8);  
Integer m = new Integer(4);
```

Which test will *not* generate an error?

- I if (k.intValue() == m.intValue())...
- II if ((k.intValue()).equals(m.intValue()))...
- III if ((k.toString()).equals(m.toString()))...

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

2. _____ One of the rules for converting English to Pig Latin states: If a word begins with a consonant, move the consonant to the end of the word and add "ay". Thus "dog" becomes "ogday," and "crisp" becomes "rispcay". Suppose *s* is a *String* containing an English word that begins with a consonant. Which of the following creates the correct corresponding word in Pig Latin? Assume the declarations

```
String ayString = "ay";  
String pigString;
```

- (A) pigString = s.substring(0, s.length()) + s.substring(0,1)
+ ayString;
- (B) pigString = s.substring(1, s.length()) + s.substring(0,0)
+ ayString;
- (C) pigString = s.substring(0, s.length()-1) + s.substring(0,1)
+ ayString;
- (D) pigString = s.substring(1, s.length()-1) + s.substring(0,0)
+ ayString;
- (E) pigString = s.substring(1, s.length()) + s.substring(0,1)
+ ayString;

3. — The following program segment is intended to find the index of the first negative integer in `arr[0] ... arr[N-1]`, where `arr` is an array of `N` integers.

```
int i = 0;
while (arr[i] >= 0)
{
    i++;
}
location = i;
```

This segment will work as intended

- (A) always.
 - (B) never.
 - (C) whenever `arr` contains at least one negative integer.
 - (D) whenever `arr` contains at least one nonnegative integer.
 - (E) whenever `arr` contains no negative integers.
4. — Refer to the following code segment. You may assume that `arr` is an array of `int` values.

```
int sum = arr[0], i = 0;
while (i < arr.length)
{
    i++;
    sum += arr[i];
}
```

Which of the following will be the result of executing the segment?

- (A) Sum of `arr[0], arr[1], ..., arr[arr.length-1]` will be stored in `sum`.
- (B) Sum of `arr[1], arr[2], ..., arr[arr.length-1]` will be stored in `sum`.
- (C) Sum of `arr[0], arr[1], ..., arr[arr.length]` will be stored in `sum`.
- (D) An infinite loop will occur.
- (E) A run-time error will occur.

5. ——— Refer to the following code segment. You may assume that array `arr1` contains elements `arr1[0]`, `arr1[1]`, ..., `arr1[N-1]`, where `N = arr1.length`.

```
int count = 0;
for (int i = 0; i < N; i++)
    if (arr1[i] != 0)
    {
        arr1[count] = arr1[i];
        count++;
    }
int[] arr2 = new int[count];
for (int i = 0; i < count; i++)
    arr2[i] = arr1[i];
```

If array `arr1` initially contains the elements 0, 6, 0, 4, 0, 0, 2 in this order, what will `arr2` contain after execution of the code segment?

- (A) 6, 4, 2
- (B) 0, 0, 0, 0, 6, 4, 2
- (C) 6, 4, 2, 4, 0, 0, 2
- (D) 0, 6, 0, 4, 0, 0, 2
- (E) 6, 4, 2, 0, 0, 0, 0

6. ——— Which of the following initializes an 8×10 matrix with integer values that are perfect squares? (0 is a perfect square.)

I `int[][] mat = new int[8][10];`

II `int[][] mat = new int[8][10];`
`for (int r = 0; r < mat.length; r++)`
`for (int c = 0; c < mat[r].length; c++)`
`mat[r][c] = r * r;`

III `int[][] mat = new int[8][10];`
`for (int c = 0; c < mat[r].length; c++)`
`for (int r = 0; r < mat.length; r++)`
`mat[r][c] = c * c;`

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

7. ——— Consider a class that has this private instance variable:

```
private int[] [] mat;
```

The class has the following method, alter.

```
public void alter(int c)
{
    for (int i = 0; i < mat.length; i++)
        for (int j = c + 1; j < mat[0].length; j++)
            mat[i][j-1] = mat[i][j];
}
```

If a 3×4 matrix mat is

```
1 3 5 7
2 4 6 8
3 5 7 9
```

then alter(1) will change mat to

(A) 1 5 7 7
2 6 8 8
3 7 9 9

(B) 1 5 7
2 6 8
3 7 9

(C) 1 3 5 7
3 5 7 9

(D) 1 3 5 7
3 5 7 9
3 5 7 9

(E) 1 7 7 7
2 8 8 8
3 9 9 9

Questions 8 - 9 refer to the following `BingoCard` class declaration.

```
public class BingoCard
{
    private int[] myCard;

    /* Default constructor: Creates BingoCard with
     * 20 random digits in the range 1 - 90. */
    public BingoCard()
    { /* implementation not shown */ }

    /* Display BingoCard. */
    public void display()
    { /* implementation not shown */ }

    ...
}
```

A program that simulates a bingo game declares an array of `BingoCard`. The array has `NUMPLAYERS` elements, where each element represents the card of a different player. Here is a code segment that creates all the bingo cards in the game:

```
/* declare array of BingoCard */
/* construct each BingoCard */
```

8. — Which of the following is a correct replacement for

```
/* declare array of BingoCard */?
```

- (A) `int[] BingoCard = new BingoCard[NUMPLAYERS];`
- (B) `BingoCard[] players = new int[NUMPLAYERS];`
- (C) `BingoCard[] players = new BingoCard[20];`
- (D) `BingoCard[] players = new BingoCard[NUMPLAYERS];`
- (E) `int[] players = new BingoCard[NUMPLAYERS];`

9. — Assuming that `players` has been declared as an array of `BingoCard`, which of the following is a correct replacement for

```
/* construct each BingoCard */
```

```
I for (BingoCard card : players)
    card = new BingoCard();
```

```
II for (BingoCard card : players)
    players[card] = new BingoCard();
```

```
III for (int i = 0; i < players.length; i++)
    players[i] = new BingoCard();
```

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

10. — Suppose the characters 0, 1, ..., 8, 9, A, B, C, D, E, F are used to represent a hexadecimal (base-16) number. Here $A = 10$, $B = 11$, ..., $F = 15$. What is the largest base-10 integer that can be represented with a two-digit hexadecimal number, such as 14 or 3A?

- (A) 32
- (B) 225
- (C) 255
- (D) 256
- (E) 272

Go to the next page

Free Response

An electric car that runs on batteries must be periodically recharged for a certain number of hours. The battery technology in the car requires that the charge time not be interrupted.

The cost for charging is based on the hour(s) during which the charging occurs. A rate table lists the 24 one-hour periods, numbered from 0 to 23, and the corresponding hourly cost for each period. The same rate table is used for each day. Each hourly cost is a positive integer. A sample rate table is given below.

Hour	Cost	Hour	Cost	Hour	Cost
0	50	8	150	16	200
1	60	9	150	17	200
2	160	10	150	18	180
3	60	11	200	19	180
4	80	12	40	20	140
5	100	13	240	21	100
6	100	14	220	22	80
7	120	15	220	23	60

The class `BatteryCharger` below uses a rate table to determine the most economic time to charge the battery. You will write two of the methods for the `BatteryCharger` class.

```
public class BatteryCharger
{
    /** rateTable has 24 entries representing the charging costs for hours 0 through 23. */
    private int[] rateTable;

    /** Determines the total cost to charge the battery starting at the beginning of startHour.
     * @param startHour the hour at which the charge period begins
     *      Precondition:  $0 \leq \text{startHour} \leq 23$ 
     * @param chargeTime the number of hours the battery needs to be charged
     *      Precondition:  $\text{chargeTime} > 0$ 
     * @return the total cost to charge the battery
     */
    private int getChargingCost(int startHour, int chargeTime)
    { /* to be implemented in part (a) */ }

    /** Determines start time to charge the battery at the lowest cost for the given charge time.
     * @param chargeTime the number of hours the battery needs to be charged
     *      Precondition:  $\text{chargeTime} > 0$ 
     * @return an optimal start time, with  $0 \leq \text{returned value} \leq 23$ 
     */
    public int getChargeStartTime(int chargeTime)
    { /* to be implemented in part (b) */ }

    // There may be instance variables, constructors, and methods that are not shown.
}
```


- (a) Write the `BatteryCharger` method `getChargingCost` that returns the total cost to charge a battery given the hour at which the charging process will start and the number of hours the battery needs to be charged.

For example, using the rate table given at the beginning of the question, the following table shows the resulting costs of several possible charges.

Start Hour of Charge	Hours of Charge Time	Last Hour of Charge	Total Cost
12	1	12	40
0	2	1	110
22	7	4 (the next day)	550
22	30	3 (two days later)	3,710

Note that a charge period consists of consecutive hours that may extend over more than one day.

Complete method `getChargingCost` below.

```
/** Determines the total cost to charge the battery starting at the beginning of startHour.
 * @param startHour the hour at which the charge period begins
 *      Precondition:  $0 \leq \text{startHour} \leq 23$ 
 * @param chargeTime the number of hours the battery needs to be charged
 *      Precondition:  $\text{chargeTime} > 0$ 
 * @return the total cost to charge the battery
 */
private int getChargingCost(int startHour, int chargeTime)
```

- (b) Write the `BatteryCharger` method `getChargeStartTime` that returns the start time that will allow the battery to be charged at minimal cost. If there is more than one possible start time that produces the minimal cost, any of those start times can be returned.

For example, using the rate table given at the beginning of the question, the following table shows the resulting minimal costs and optimal starting hour of several possible charges.

Hours of Charge Time	Minimum Cost	Start Hour of Charge	Last Hour of Charge
1	40	12	12
2	110	0 or 23	1 0 (the next day)
7	550	22	4 (the next day)
30	3,710	22	3 (two days later)

Assume that `getChargingCost` works as specified, regardless of what you wrote in part (a).

Complete method `getChargeStartTime` below.

```
/** Determines start time to charge the battery at the lowest cost for the given charge time.
 * @param chargeTime the number of hours the battery needs to be charged
 *      Precondition: chargeTime > 0
 * @return an optimal start time, with  $0 \leq \text{returned value} \leq 23$ 
 */
public int getChargeStartTime(int chargeTime)
```