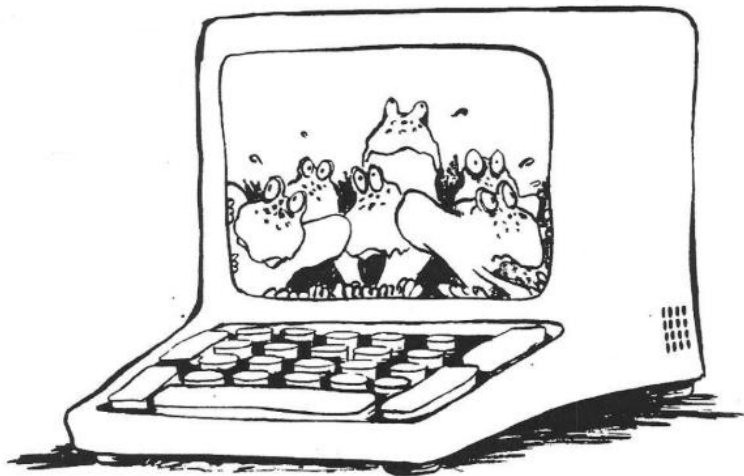


*AP Computer
Science A
Questions
of the
Day*



The College Board released AP Computer Science A Multiple Choice items for the following exam years:

Pascal: 1984, 1988, 1992

C++: 1999

Java: 2004, 2009, 2015

So, what can we do with those items from non-Java years? Translate them from Pascal or C++ to Java.

The following pages contain MC items that were translated from the 1988, 1992 and 1999 exams (thank you Dave Wittrey). Keep in mind that the content of the exam in Pascal and C++ was much different than it is today, but some of the items are still good to view. Some of the items were revised to reflect the changes in paradigm.

The items in this document are grouped by content focus.

And, what can we do with computers that we can't use anymore?

1992 # 1

If addition had higher precedence than multiplication, then the value of the expression

$$1 + 2 * 3 + 4 * 5$$

would be which of the following?

- (A) 27
- (B) 47
- (C) 65
- (D) 69
- (E) 105



Top-down programming is illustrated by which of the following?

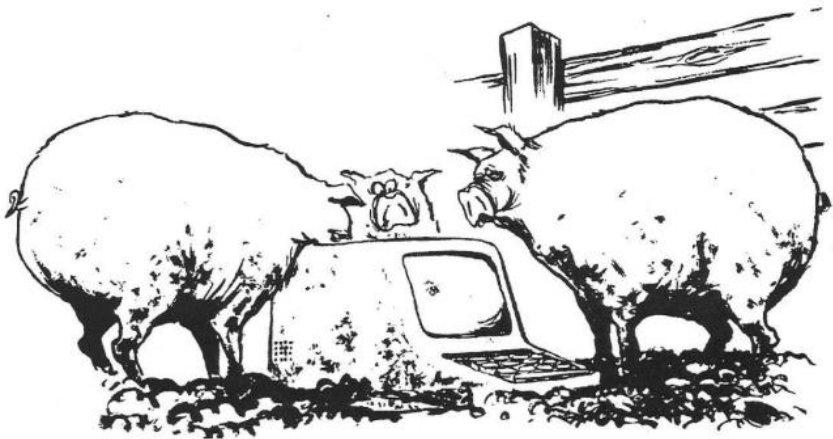
- (A) writing a program in the top-down dialect of Java
- (B) writing a program without using goto's
- (C) writing the first lines of a program before writing the last lines
- (D) writing the program in terms of abstract operations and then refining those abstract operations
- (E) writing and testing the lowest-level routines first and then combining these routines to form appropriate abstract operations



A Java source file has been compiled into a .class file. It will run on several different types of computers, ranging from microcomputers to mainframes. Which of the following might be different on the different machines?

- I. The number of bits used to represent an int while the .class file is being executed
- II. The number of reserved words available to the source file during compilation
- III. The Java virtual machine used to execute the .class file

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



Consider the following code segment:

```
x = !y;  
y = !x;
```

Assume that `x` and `y` are initialized variables of type `boolean`. Which of the following statements is (are) true?

- I. The final value of `x` is the same as the initial value of `x`.
- II. The final value of `y` is the same as the initial value of `y`.
- III. The final value of `x` is the same as the initial value of `y`.

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

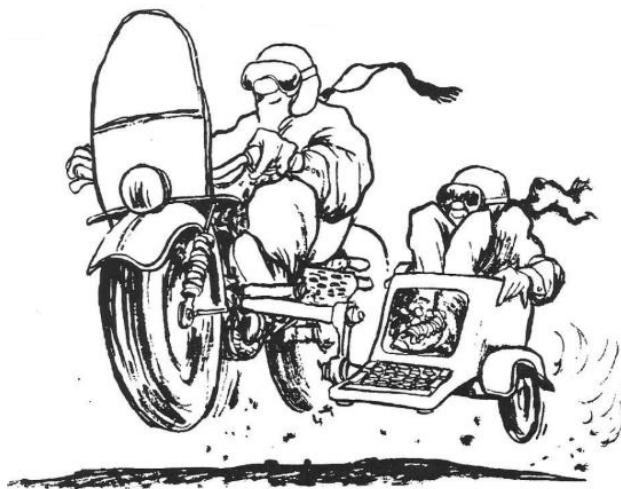


1988 # 23.

If `b` is a boolean variable, then the statement

`b = b == false;` has what effect?

- (A) It causes a compile-time error message.
- (B) It causes a run-time error message.
- (C) It causes `b` to have the value `false` regardless of its value just before the statement was executed.
- (D) It always changes the value of `b`.
- (E) It changes the value of `b` if and only if `b` had value `true` just before the statement was executed.



The Boolean expression

$(\text{num} > \text{max}) \ || \ !(\text{max} < \text{num})$

can be simplified to

- (A) $\text{max} != \text{num}$
- (B) $\text{max} == \text{num}$
- (C) $(\text{num} < \text{max}) \ \&\& \ !(\text{max} < \text{num})$
- (D) `false`
- (E) `true`



Refer to the following Boolean expression.

```
((i <= n) && (a[i]== 0)) || ((i >= n) && (a[i - 1]== 0))
```

26.

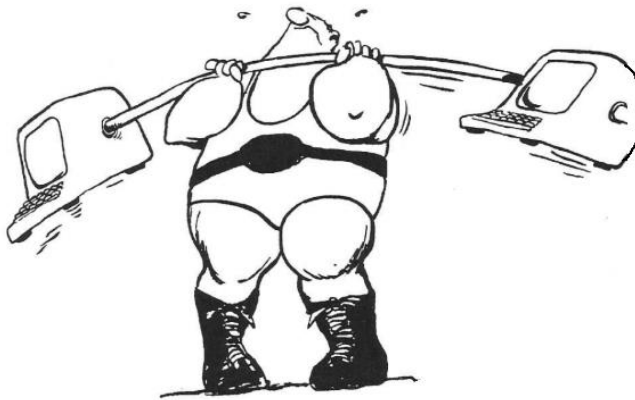
Under which of the following conditions must the Boolean expression have the value `true`?

- (A) `(i <= n) || (i >= n)`
- (B) `(a[i] == 0) && (a[i - 1] == 0)`
- (C) `i == n`
- (D) `i < n`
- (E) `i > n`

27.

Evaluation of the Boolean expression is guaranteed to cause a run-time error under which of the following conditions?

- (A) `i < 0`
- (B) Neither `a[i]` nor `a[i - 1]` has the value zero.
- (C) array `a` is of size `n`.
- (D) array `a` is of size 2.
- (E) None of the above



Consider the following code segment. Assume that neither *<condition 1>*, *<condition 2>*, nor *<condition 3>* changes the value of *k*.

```
int k = 0;  
if (<condition 1>)  
    k++;  
if (<condition 2>)  
    k++;  
if (<condition 3>)  
    k++;
```

What are the possible final values of *k* after the code segment executes?

- (A) 0 only
- (B) 1 only
- (C) 0 or 1 only
- (D) 1, 2, or 3 only
- (E) 0, 1, 2, or 3



1992 # 12,13

Refer to the following information. The code

```
if (n == 1)
    k -= 1;
else if (n == 2)
    k -= 2;
```

is rewritten in the form

```
if (<condition>)
    <assignment statement>;
```

where *<condition>* and *<assignment statement>* are chosen so that the rewritten code performs the same task as the original code. Assume that both *n* and *k* are integer variables.

12.

Which of the following could be used as *<condition>*?

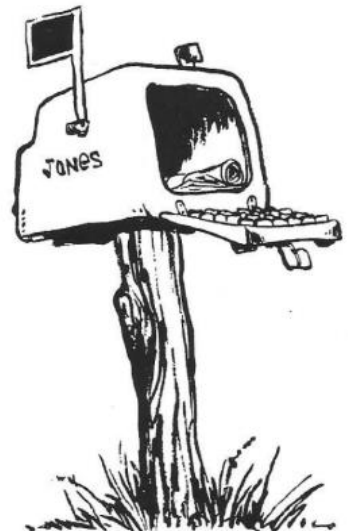
- I. $(n == 1) \ || \ (n == 2)$
- II. $(n == 1) \ \&\& \ (n == 2)$
- III. $(n \geq 1) \ \&\& \ (n \leq 2)$

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

13.

Assume that evaluating *<condition>* changes neither *n* nor *k*. Which of the following could be used as *<assignment statement>*?

- (A) $k -= n;$
- (B) $k -= 1;$
- (C) $k -= 2;$
- (D) $k += n;$
- (E) $k = n - k;$



Consider the following program segment.

```
int x, y;  
x = IO.readInt();    // integer is read  
while (x != 0)  
{  
    y = IO.readInt();    // integer is read  
    if (y == 0)  
        System.out.println("yes");  
    else  
        System.out.println("no");  
    x = IO.readInt();    // integer is read  
}
```

Which of the following input sequences ensures that every statement of the program is executed at least once?

- (A) 1 1 1 2 0
- (B) 1 0 1 1 0
- (C) 1 0 1 0 1 0 0
- (D) 1 0 0
- (E) 1 1 0



Consider the following program segment:

```
int k, sum;
System.out.print("Enter a nonnegative number to sum, ");
System.out.println("negative number to quit");
k = input.readInt();           // read an integer
while (k >= 0)
{
    sum += k;
    System.out.print("Enter a nonnegative number to sum, ");
    System.out.println("negative number to quit");
    k = input.readInt();       // read an integer
}
System.out.println(sum);
```

The segment contains the incorrect use of an uninitialized variable. Which of the following is the first line in which this occurs?

- (A) k in "k = input.readInt()"
- (B) k in "(k >= 0)"
- (C) sum in "sum += k"
- (D) k in "sum += k"
- (E) sum in "System.out.println(sum)"



Consider the following code segments.

Segment 1

```
x = IO.readInt();           // read in an integer
while (x > 0)
{
    System.out.println(x);
    x --;
}
while (x > 0)
{
    System.out.println(x);
    x --;
}
```

Segment 2

```
x = IO.readInt();           // read in an integer
while (x > 0)
{
    System.out.println(x);
    x --;
}
```

Under which of the following conditions will the two code segments produce exactly the same output?

- I. The value read in the first line is less than zero.
- II. The value read in the first line is greater than zero.
- III. The value read in the first line is equal to zero.

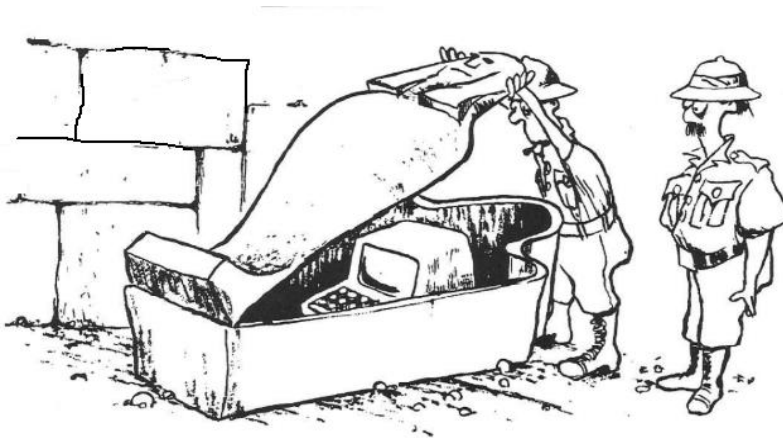
- (A) I only
- (B) II only
- (C) III only
- (D) I and III only
- (E) I, II, and III



If the value of $f(x, y, z)$ is always an integer, which of the following conditions ensures that the loop below terminates?

```
while (f(x, y, z) != 0)
    <some computation>
```

- (A) x , y , and z are each increased during each iteration
- (B) x , y , and z are each decreased during each iteration
- (C) the sum of x , y , and z is decreased during each iteration
- (D) the value of $f(x, y, z)$ is decreased during each iteration
- (E) the value of $\text{Math.pow}(f(x, y, z), 2)$ is decreased during each iteration



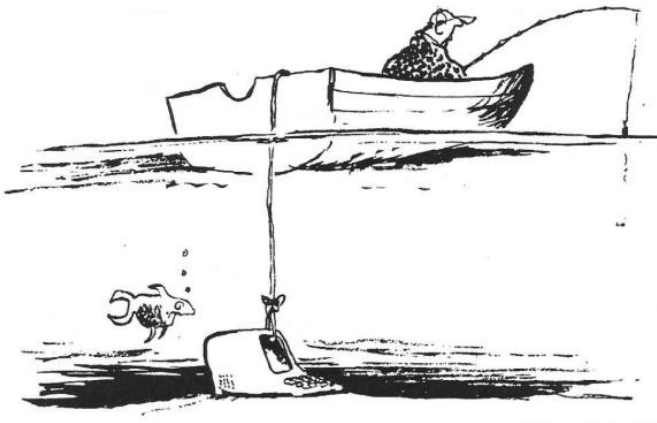
Consider the following two program segments.

```
//SEGMENT 1
i = first;
while (i <= last)
{
    System.out.println("while loop output");
    i++;
}
```

```
//SEGMENT 2
for (int i = first; i <= last; i++)
    for (int j = 1; j <= i; j++)
        System.out.println("for loop output");
```

If $\text{first} = -1$ and $\text{last} = 1$, how many times will the output lines “while loop” and “for loop output”, respectively, be written?

- (A) 2, 1
- (B) 2, 6
- (C) 3, 1
- (D) 3, 3
- (E) 3, 6



During one iteration of the outer for loop of the program segment below, how many times is the body of the inner loop executed?

```
for (int i = 1; i <= n-1; i++)  
    for (int j = n; j >= i+1; j--)  
        if (a[j-1] > a[j])  
        {  
            temp = a[j-1];  
            a[j-1] = a[j];  
            a[j] = temp;  
        }
```

- (A) $i + 1$
- (B) $n - i + 1$
- (C) $i - n + 1$
- (D) $n(i - 1)$
- (E) $n - i$



1992 # 2, 3

Refer to the following methods:

```
public int add(int x, int y) {...}  
// postcondition: returns x + y  
  
public int multiply(int x, int y) {...}  
// postcondition: returns x * y
```

2.

What is the value of the expression `multiply(3, add(4, 5))`?

- (A) 12
- (B) 17
- (C) 23
- (D) 27
- (E) 60

3.

Consider the following expression:

```
multiply(2, add(multiply(a, b), add(multiply(a, c), multiply(b, c))))
```

Which of the following corresponds to this expression?

- (A) $2 * a * b + a * c + b * c$
- (B) $a * b + a * c + b * c + 2$
- (C) $2 * (a * b + (a * c + b * c))$
- (D) $a * b + (a * c + b * c)$
- (E) $a * b + (a * c + b * c) * 2$



Consider the following incomplete method:

```
public int fun(int x, int y)
// precondition: (x * y) > 0
// postcondition: returns a value > 0
{
    <body of fun>
}
```

Assume that *<body of fun>* is replaced with code so that fun meets the specification defined by its precondition and its postcondition.

7.

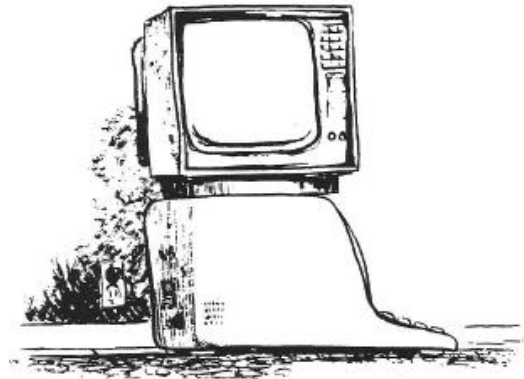
What can be assumed about the value returned by the call `fun(0, 0)`?

- (A) The value is 0.
- (B) The value is not equal to 0.
- (C) The value is less than or equal to 0.
- (D) The value is greater than or equal to 0.
- (E) No assumption can be made about the value returned.

8.

What can be assumed about the value returned by the call `fun(-1, 1)`?

- (A) The value is -1.
- (B) The value is 0.
- (C) The value is 1.
- (D) The value is greater than or equal to 0.
- (E) No assumption can be made about the value returned.



1999 #27.

Consider the following method:

```
public boolean somethingDifferent(boolean p, boolean q)
{
    return ((p || q) && !(p && q));
}
```

What does method `somethingDifferent` return?

- (A) `somethingDifferent` always returns false.
- (B) `somethingDifferent` always returns true.
- (C) `somethingDifferent` returns true whenever `p` is false.
- (D) `somethingDifferent` returns true whenever `q` is false.
- (E) `somethingDifferent` returns true whenever `p` is not equal to `q`.



Consider the method `total` below.

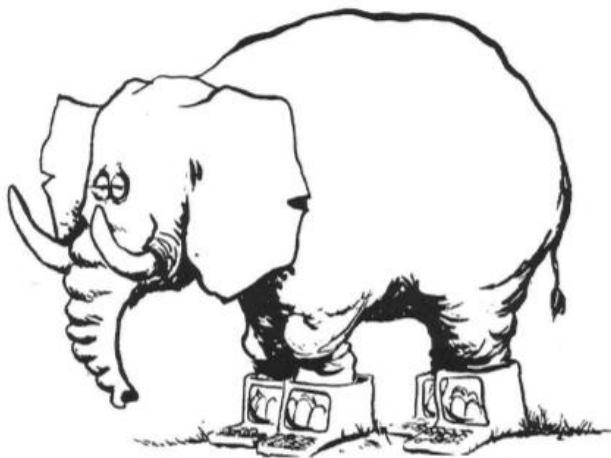
```
public int total(int result, int a, int b)
{
    if (a == 0)
    {
        if (b == 0)
            return result * 2;
        }
    else
        return result * 3;
}
```

The assignment statement

```
x = total (5, 0, 1);
```

must result in

- (A) x being assigned the value 0
- (B) x being assigned the value 5
- (C) x being assigned the value 10
- (D) x being assigned the value 15
- (E) none of the above



Consider the following code segment to print a calendar.

```
int month, year;
year =          // read in a valid year

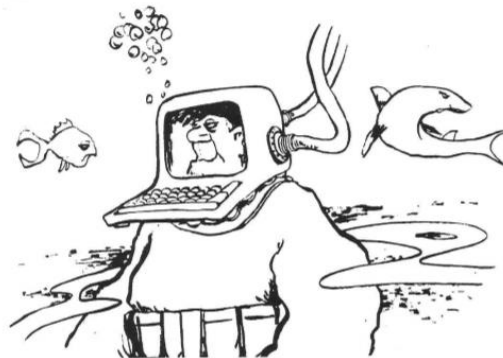
for(month = 1; month <= 12; month++)
{
    printHeading(month, year);
    printDays(month, year);
}
```

Consider the following method.

```
public void printDays(int month, int year)
{
    int day;
    printSpaces(month, year);
    for (day = 1; day <= numDaysIn(month, year); day++)
    {
        System.out.println(day + " ");
        if (endOfWeek(day, month, year))
        {
            System.out.println();
        }
    }
}
```

Suppose that when the program is run, every month is printed correctly except for February, for which only a heading and some white space is printed. Of the following methods, which is most likely to contain the error?

- (A) numDaysIn
- (B) printSpaces
- (C) endOfWeek
- (D) printHeading
- (E) printDays



Consider the following program:

```
public class Scope
{
    private static int x, y;

    public static void one(int a, int b)
    {
        a = b + 1;
        b = a + 2;
    }
    public static void two(int a, int b)
    {
        a = b + 1;
        b = a + 2;
        x = a;
        y = b;
    }

    public static void main(String[] args)
    {
        int x, y;
        x = 1;
        y = 2;
        one(x, y);
        two(x, y);
        System.out.println(x + " " + y);
    }
}
```

What is the output of the program?

- (A) 1 2
- (B) 4 4
- (C) 6 5
- (D) 6 8
- (E) 9 8



The purpose of a method's precondition is to

- (A) initialize the local variables of the method
- (B) describe the conditions under which the compiler is to abort compilation
- (C) describe the conditions under which the method may be called so that it satisfies its postcondition
- (D) describe the algorithm used by the method
- (E) describe the effect(s) of the method on its postcondition



Consider the following code segment to print a calendar

```
int month, year;
year = < code to get a valid year >;

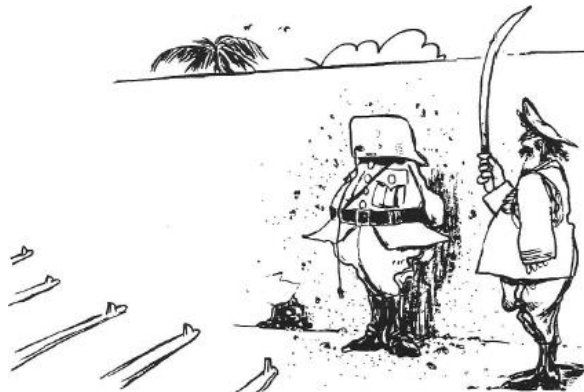
for(month = 1; month <= 12; month++)
{
    printHeading(month, year);
    printDays(month, year);
}
```

Consider the following method.

```
public void printDays(int month, int year)
{
    int day;
    printSpaces(month, year);
    for (day = 1; day <= numDaysIn(month, year); day++)
    {
        System.out.println( day + " " );
        if (endOfWeek(day, month, year))
        {
            System.out.println();
        }
    }
}
```

Suppose that when the program is run, every week on the calendar printed has eight days, which is most likely to contain the error?

- (A) printHeading
- (B) printDays
- (C) printSpaces
- (D) numDaysIn
- (E) endOfWeek



Consider the following code segment.

```
public void transfer(Input sourceFile, Output largeNumbers,
                    Output smallNumbers){
    // precondition:sourceFile contains one integer per line and is open
    // for reading at the beginning of the file; the other two files
    // are open for writing and are initially empty.
    // postcondition: All elements of sourceFile larger than the program
    // constant LARGE_INT form the contents of largeNumbers, and the
    // other elements of sourceFile form the contents of smallNumbers.
    int item;
    while (sourceFile.hasMoreToRead()) {
        item = sourceFile.getItem();
        if (item > LARGE_INT)
            // copy item to the end of largeNumbers
            { }
        else
            // copy item to the end of smallNumbers
            { }
    }
}
```

You may assume that the program constant `LARGE_INT` is available at any point of the program. You may also assume that `Input` and `Output` are valid classes which enable reading and writing to/from files.

15.

Which of the following statements is true about the code?

- (A) It will cause an error message when processed by a standard Java compiler.
- (B) It provides an example of a stage of top-down design.
- (C) In order to be considered pseudocode, the syntax must be changed.
- (D) It will compile, but if an attempt is made to execute it, an execution error will occur.
- (E) It will not compile and will not execute.

16.

If the code is completed in such a way that (1) the in-line comments together with existing code completely characterize the final code and (2) the postcondition is accurate, which of the following could be added to the postcondition without destroying its accuracy?

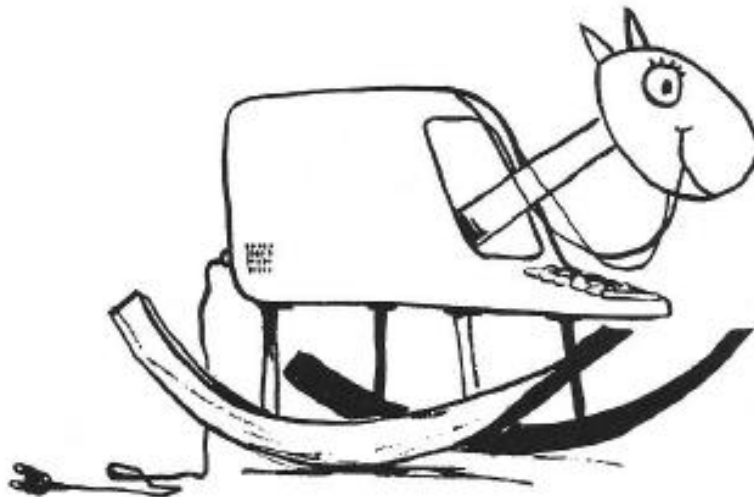
- I. `largeNumbers` and `smallNumbers` are ordered numerically.
- II. The elements of `largeNumbers` are in the same order in which they appear in `sourceFile`.
- III. The elements of `smallNumbers` are in the same order in which they appear in `sourceFile`.

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

1999 #21

The program for a video game will use several graphics routines. The program design team plans to place these routines in a graphics class, which can be compiled separately from the video game program. Which of the following would be an advantage of this plan?

- I. The graphics routines can be tested independently of the video game program, thus making it easier to locate errors in both the graphics routines and the video game program.
 - II. The programmers assigned to write the video game program can focus on the issues of the program without spending time considering how the graphics routines will be implemented.
 - III. The graphics class will be available for use in other programs.
- (A) I only
 - (B) I and II only
 - (C) I and III only
 - (D) II and III only
 - (E) I, II, and III



Java classes can include both public and private instance variables and methods. Which of the following statements represents the best design decision regarding the public and private members of a class?

- (A) All instance variables should be public to make it easier for client programs to use such data.
- (B) All methods should be public to facilitate future changes to parameter lists of methods.
- (C) All instance variables should be private to minimize the dependency between client programs and the manner in which data is stored in the class.
- (D) Some methods should be private to minimize memory usage.
- (E) All instance variables should be public and all methods should be private to make it easier to modify the class without requiring changes in the code of the client program.



Which of the following statements would best characterize a well-designed program?

- I. Methods can be tested independently before integrating them into the final program.
- II. Client programs know about, and take advantage of, implementation details of abstract data types.
- III. The algorithmic details of the abstract data types can be altered without changing client programs.

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



1992 #28, 29 (revised)

Assume that variables of type `LongIntType` can represent integers of up to 500 digits. Consider the following declaration for `LongIntType`.

```
public class LongIntType implements Comparable{
    public void readLongInt ()    {...}
    //reads a long integer from input and assigns it to this object
    public void writeLongInt ()  {...}
    //writes this long integer object to output
    public LongIntType addLongInt (LongIntType a) {...}
    //postcondition: returns this object's long integer plus a
    public int compareTo (Object obj) {...}
    //postcondition: compares two objects of type LongIntType and
    // returns -1,0, or 1 as would be expected for <, ==, and >
    public void assignIntToLong (int num) {...}
    //postcondition: assigns num as this object's long integer
    // constructors and other private and public methods and data not shown...
}
```

28.

Of the following pairs of methods, which should be coded and tested first in order to facilitate the debugging of the other methods? (Assume the runtime system provides no debugging facilities.)

- (A) `readLongInt` and `addLongInt`
- (B) `writeLongInt` and `addLongInt`
- (C) `compareTo` and `addLongInt`
- (D) `assignIntToLong` and `writeLongInt`
- (E) `readLongInt` and `assignIntToLong`

29.

In order to simulate the loop

```
for (int i = 1; i <= (n-1); i++)
    System.out.println(i);
```

the following code is used (where `i`, `One`, and `N` are of type `LongIntType`).

```
i.assignIntToLong(1);
one.assignIntToLong(1);
N.assignIntToLong(n);
while (i.compareTo(n) < 0)
{
    i.writeLongInt();
    <statement>
}
```

Which of the following should take the place of `<statement>` in order to simulate the loop correctly?

- (A) `i++;`
- (B) `one.addLongInt(i);`
- (C) `i.addLongInt(one);`
- (D) `i.addLongInt(i);`
- (E) `one.addLongInt(one);`



1999 #5

Consider the following incomplete method:

```
public int total(int[] scores)
// precondition: The sentinel -999 occurs somewhere in
// scores
{
    int k = 0;
    int sum = 0;

    while (scores[k] != -999)
    {
        <program statements>
    }
    return sum;
}
```

Method `total` is intended to return the sum of the integers in parameter `scores`, beginning with the first integer in `scores` and up to, but not including the sentinel `-999` (which occurs somewhere in `scores`). Which of the following code segments could be used to replace `<program statements>` so that `total` will work as intended?

- (A) `sum += scores[k];`
`k++;`
- (B) `k++;`
`sum += scores[k];`
- (C) `k++;`
`sum += k;`
- (D) `sum += k;`
`k++;`
- (E) `if (scores[k] != -999)`
`{`
`sum += scores[k];`
`}`



Consider the following code segment:

```
a = initialize(100); // a is an array containing 100 integers
b = initialize(100); // b is an array containing 100 integers

int k = 0;
while (k < a.length && !a[k].equals(b[k]))
{
    k++;
}
```

Which of the following must be true after the while loop terminates?

- (A) $k \geq a.length$
- (B) $k < a.length$
- (C) $(k < a.length) \ \&\& \ (!a[k].equals(b[k]))$
- (D) $(k \geq a.length) \ || \ (!a[k].equals(b[k]))$
- (E) $(k \geq a.length) \ || \ (a[k].equals(b[k]))$



Assume `a` is defined as follows:

```
int[] a = new int[5];
```

Consider the following code segment:

```
int k;  
for (k = a.length - 1; k > 0; k--)  
{  
    a[k] = a[k - 1];  
}
```

Assume `a` contains the following values before the code segment is executed:

0	1	2	3	4
10	15	20	25	30

What values will `a` contain after the code segment is executed?

(A)

10	10	10	10	10
----	----	----	----	----

(B)

10	10	15	20	25
----	----	----	----	----

(C)

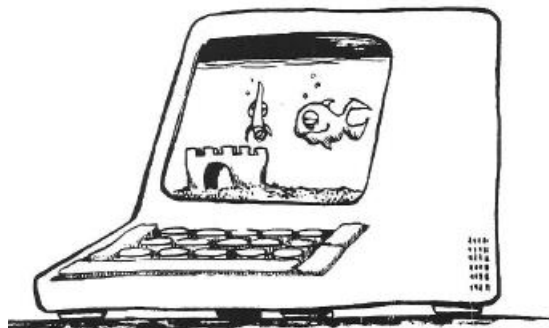
10	15	20	25	30
----	----	----	----	----

(D)

15	20	25	30	30
----	----	----	----	----

(E)

25	25	25	25	25
----	----	----	----	----

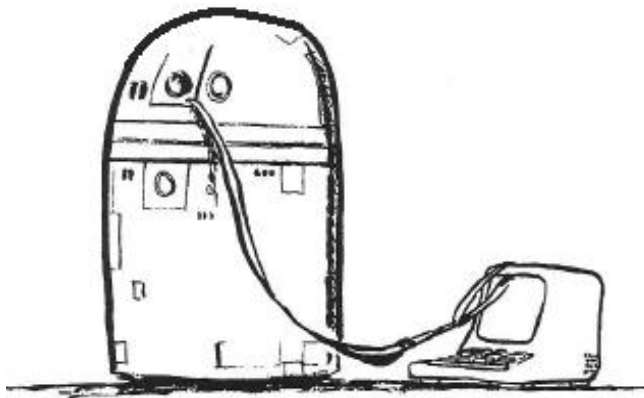


Consider the following definitions and code segment:

```
int[] a = new int[7];  
  
int x;  
  
for (x = 0; x < a.length; x++)  
{  
    a[x] = x;  
}  
for (x = 0; x < a.length; x++)  
{  
    a[x/3] = a[x];  
}
```

What values will a contain after the code segments is executed?

- (A) 0 0 1 1 1 2 2
- (B) 0 1 2 3 4 5 6
- (C) 1 2 0 1 2 0 1
- (D) 2 3 4 5 6 0 1
- (E) 2 5 6 3 4 5 6



Assume that a is an array of n integers and that variable k has a value in the range $0 < k < n$. Also assume that the following assertion is true:

for all j , $0 < j < k$, $a[j] < a[j+1]$

Which of the following is a valid conclusion?

- (A) All elements of a are in increasing order.
- (B) All elements of a are in decreasing order.
- (C) Elements 0 through k of a are in increasing order.
- (D) Elements 0 through k of a are in decreasing order.
- (E) The smallest value in a is stored in $a[0]$.

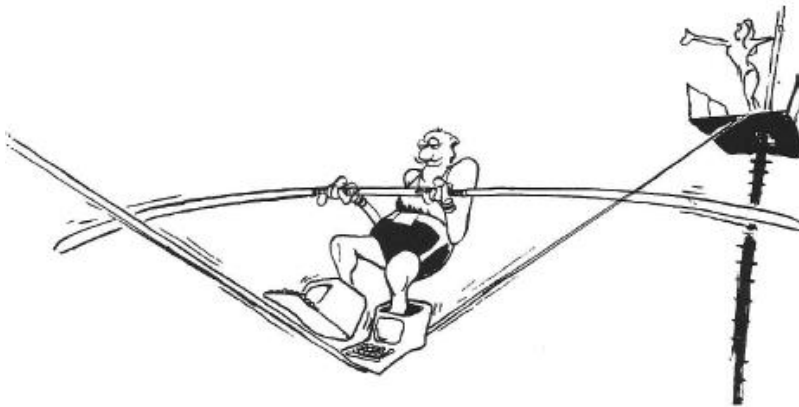


Consider the following code segment.

```
int[] a;  
a = initialize();    // initializes the elements of a  
  
int k;  
for (k = 0; k < a.length; k++)  
{  
    int pos1 = k;  
    int pos2 = a.length - k - 1;  
    swap(a, pos1, pos2 );  
}
```

Assume that method `swap` takes the array `a` and interchanges the values at locations `pos1` and `pos2`. Which of the following best characterizes the effect of the for loop?

- (A) It sorts the elements of `a`.
- (B) It reverses the elements of `a`.
- (C) It reverses the order of the first half of `a` and leaves the second half unchanged.
- (D) It reverses the order of the second half of `a` and leaves the first half unchanged.
- (E) It leaves all of the elements of `a` in their original order.



A program is to be written that will manipulate 100 names of 30 characters each. The program will perform several tasks, including reading names, sorting names, and printing names. Of the following definitions, which will be most appropriate for the program?

- (A) `String[] names = new String[30];`
- (B) `String[] names = new String[100];`
- (C) `String[][]names = new String[30][100];`
- (D) `String[][]names = new String[100][30];`
- (E) `char[][]names = new char[100][30];`

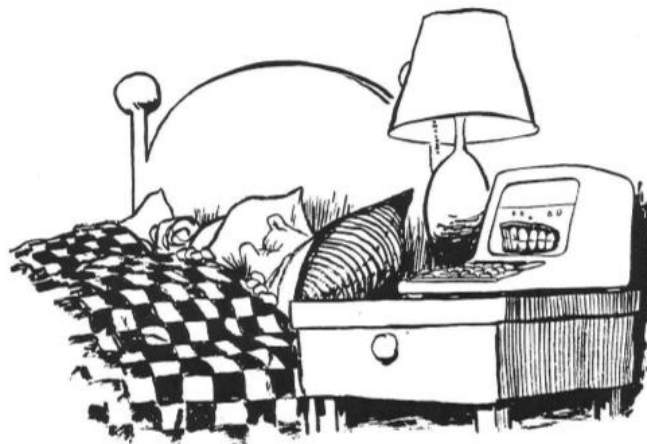


Assume that A is an array of n positive integers and that the following assertion is true.

$A[0] > A[k]$ for all k such that $0 < k < n$

Which of the following is a valid conclusion?

- (A) The array is sorted in ascending order.
- (B) The array is sorted in descending order.
- (C) All values in the array are identical.
- (D) A[0] holds the smallest value in the array.
- (E) A[0] holds the largest value in the array.



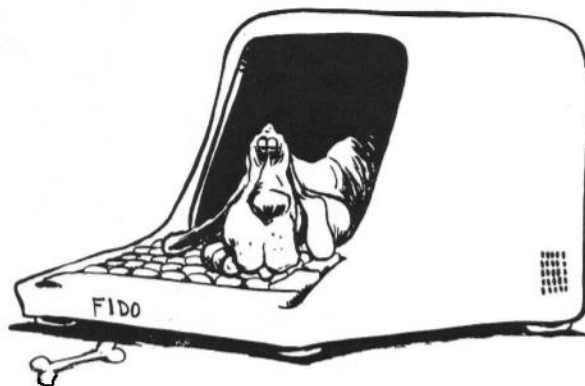
Consider the following declarations.

```
public class ListType{
private static int MAX_ITEMS = <some positive number>;
    // maximum # of elements that nums would contain
private int[] nums ; // holds length number of integers
private int length; // holds the current number of valid
// items in nums (held in positions 0..length-1)

public double averageOf ()
{
    int sum = 0, count;
    for (count = 0; count < length; count++)
        <statement 1>;
    <statement 2>;
}
// constructors and other private and public methods and
// data not shown.
}
```

Which of the following substitutions for <statement 1> and <statement 2> will cause averageOf to return the average of the values in nums?

- | | <statement 1> | <statement 2> |
|-----|------------------------------|------------------------------|
| (A) | sum += nums[MAX_ITEMS-1] | return(double) sum/count |
| (B) | sum += nums[count] | return(double) sum/length |
| (C) | sum += nums[nums.length - 1] | return(double) sum/MAX_ITEMS |
| (D) | sum += nums[nums.length -1] | return(double) sum/MAX_ITEMS |
| (E) | sum += nums[count] | return sum/count |



The following code is designed to set `index` to the location of the first occurrence of `goal` in the array `A`, and to set `index` to -1 if `goal` does not occur in `A`.

```
index = 0;
while (A[index] != goal)
    index++;
if (A[index] != goal)
    index = -1;
```

Which of the following describes the condition under which this program segment will fail to perform the task described?

- (A) Whenever `goal` is the first element of the array
- (B) Whenever `goal` is the last element of the array
- (C) Whenever `goal` is not present in the array
- (D) Whenever `goal` is -1
- (E) Whenever `goal = A[goal]`



1999 #12, 13

Refer to the following data structure, designed to store information about hiking trails in the United States.

```
public class Trail{
    public boolean goodMonths(int k) {...}
    // goodMonths(k) == true if this trail is good for
    // hiking in month k, false otherwise
    public String getName() {...}
    // returns the name of the trail
    private String name;           // name of trail
    private String state;          // location of trail
    private distance;              // length of trail in kilometers
    private boolean[] myGoodMonths; // constructor
    // initializes to size 12 and assigns values
    // constructors and other private and public methods and
    // data not shown.
}
```

#12 Assume that `t` is an instantiated object of class `Trail` and the integers `j` and `k` represent valid months. Which of the following expressions always evaluates to true if the trail represented by `t` is good for hiking during month `j` or `k`?

- (A) `t.goodMonths(j) && t.goodMonths(k)`
- (B) `t.goodMonths(j) || t.goodMonths(k)`
- (C) `(t.goodMonths() == j) || (t.goodMonths() == k)`
- (D) `(t.goodMonths() == j) && (t.goodMonths() == k)`
- (E) `t.myGoodMonths[j] || t.myGoodMonths[k]`

#13 Consider the incomplete method `printTrails` given below. `printTrails` should print the names of the trails in its parameter `trailArray` that are good for hiking in the month specified by parameter `month`.

```
public void printTrails(Trail[]trailArray, int month){
    int k;
    for (k = 0; k < trailArray.length; k++)
    {
        <loop body>
    }
}
```

Which of the following could be used to replace `<loop body>` so that `printTrails` works as intended?

- (A) `if (trailArray[k].goodMonths(month))`
 `System.out.println(trailArray[k].getName());`
- (B) `if (trailArray[month].goodMonths(k))`
 `System.out.println(trailArray[month].getName());`
- (C) `if (trailArray[k] == month)`
 `System.out.println(trailArray[k].getName());`
- (D) `if (trailArray.goodMonths == month)`
 `System.out.println(trailArray.goodMonths.getName());`
- (E) `if (trailArray.goodMonths[k] == month)`
 `System.out.println(trailArray.goodMonths[k].getName());`

Refer to the following information. Consider the following declaration:

```
public class Restaurant {
    public String name() {...} // returns the restaurant's name
    public double price() {...} // returns the price of a meal
    // (all meals in a specific in a restaurant cost the same)
    public int capacity() {...} // returns the maximum number of
    // customers the restaurant can serve at one time
    // other methods not shown

    private String myName; // the restaurant's name
    // other data members not shown
}
```

37. Assume that a client program declares and initializes rList as follows:

```
Restaurant[] rList;
rList = initialize(); // sizes rList and initializes
                      // its elements
```

Which of the following code segments correctly prints the names of all the restaurants whose meal price is under \$10.00?

- I. int r;
 for (r = rList.length - 1; r >= 0; r--)
 if (rList[r].price() < 10.00)
 System.out.println(rList[r].name());

- II. int r;
 for (r = 0; r < rList.length; r++)
 if (rList[r].price() < 10.00)
 System.out.println(rList[r].name());

- III. int r;
 for (r = 0; r < rList.length; r++)
 if (rList[r].price() < 10.00)
 System.out.println(rList[r].myName);

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

1999 #38.

Consider the following method.

```
public void printSomeRestaurants(Restaurant[] rList)
// precondition:rList.length > 0
{
    int r;
    int numRests = rList.length;
    double sum = 0.0;
    double average;

    for (r = 0; r < numRests; r++)
    {
        sum += rList[r].price();
    }
    average = sum / numRests;
    for (r = 0; r < numRests; r++)
    {
        if ((rList[r].capacity() >= 50) &&
            (rList[r].price() < average))
        {
            System.out.println(rList[r].name());
        }
    }
}
```

Of the following, which best describes the behavior of `printSomeRestaurants`?

- (A) It prints the name of the first restaurant whose meal price is below the average of all restaurants.
- (B) It prints the name of the first restaurant whose capacity is at least 50 and whose meal price is less than the average for the meal prices for all restaurants.
- (C) It prints the names of all restaurants whose capacity is at least 50 and whose meal price is below the average for the meal prices for all restaurants.
- (D) It prints the average of the meal prices of all restaurants.
- (E) It prints the average of the meal prices of all restaurants whose capacity is at least 50.



Consider a Java method named `match` that is indicated by the following method header.

```
public boolean match (int[] a, int[] b)
//precondition: a.length == b.length
```

The method compares the elements of two arrays of integers and returns the value `true` if the elements in corresponding positions are equal, and returns `false` otherwise.

23.

In Java, the formal parameters `a` and `b` are pass-by-value. If `match` is invoked with a statement such as:

```
match (x, y)    // where x and y are integer arrays
               // and x.length == y.length
```

which of the following statements about parameter passing in Java is true?

- (A) the contents of `a` and `b` can not be changed during execution of `match(x, y)`
- (B) the contents of `x` and `y` can not be changed during execution of `match(x, y)`
- (C) the lengths of `a` and `b` can not be changed during execution of `match(x, y)`
- (D) the lengths of `x` and `y` can not be changed during execution of `match(x, y)`
- (E) value parameters protect against side effects, but the drawback is that it requires time to be spent in copying all of the array elements every time the method is called

24.

Which of the following is code for the body of the method that fits the specification given above?

- (A)

```
int i = 0;
while (i < a.length && a[i] == b[i])
    i++;
return (i == a.length);
```
- (B)

```
temp = false;
for (int i = 0; i < a.length; i++)
    if (a[i] == b[i])
        temp = true;
return temp;
```
- (C)

```
for (int i = 0; i < a.length; i++)
    temp = (a[i] == b[i]);
return temp;
```
- (D)

```
int i = 0;
while (i < a.length && a[i] == b[i])
    i++;
return (i > a.length);
```
- (E)

```
int i = 0;
while (a[i] == b[i] && i < a.length)
    i++;
return (i == a.length);
```



Consider the following declarations.

```
public class SomeClass
{
    private boolean found;
    private int loc;
    private int[] list;
    public void find (int num)
    {
        found = false;
        loc = 0;
        while (!found && loc < list.length )
        {
            if (list[loc] == num)
                found = true;
            loc++;
        }
    }
    // constructors and other private and public methods and data not
    shown...
}
```

Which of the following is a correct postcondition for method `find`?

- (A) `found`
- (B) `found && (loc >= list.length)`
- (C) `(list[loc] == num) || (loc == list.length)`
- (D) `loc == list.length`
- (E) `!found && (loc < list.length)`

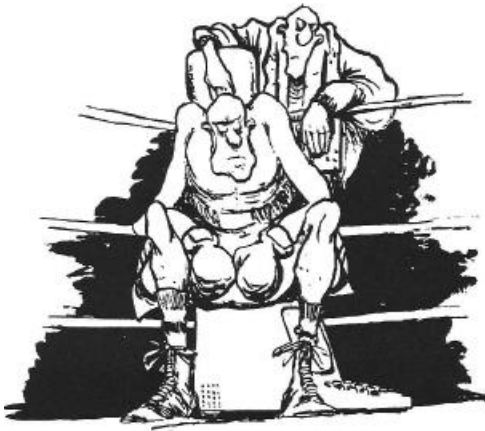


Consider the following definitions.

```
public boolean state (int[] list, int value)
{
    int counter;
    boolean flag = false;
    for (counter = 0; counter < list.length; counter++)
        flag = (list[counter] == value);
    return flag;
}
```

Under which of the following conditions must the method above return true?

- (A) Under all conditions
- (B) Under the condition that $\text{value} = \text{list}[\text{list.length}-1]$
- (C) Under the condition that $\text{value} = \text{list}[i]$ for some i such that $0 \leq i < \text{list.length}$
- (D) Under the condition that $\text{value} \neq \text{list}[i]$ for all i such that $0 \leq i < \text{list.length}$
- (E) Under no conditions

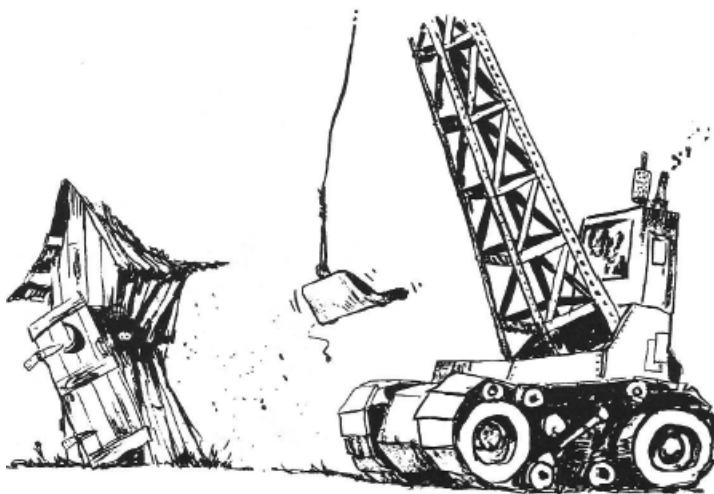


The following program segment is intended to sum $a[0]$ through $a[n-1]$, where $n = a.length$

```
sum = 0;
i = 0;
n = a.length;
while (i != n)
{
    i++;
    sum += a[i];
}
```

In order for this segment to perform as intended, which of the following modifications, if any, should be made?

- (A) no modification is necessary
- (B) the segment `sum = 0; i = 0;` should be changed to `sum = a[1]; i = 1;`
- (C) the segment `sum = 0; i = 0;` should be changed to `sum = a[0]; i = 1;`
- (D) the segment `while (i != n)` should be changed to `while (i <= n)`
- (E) the segment `i++;` should be interchanged with `sum += a[i];`



All of the entries in the array `a` with position indices `spot` to `last`, inclusive, ($0 \leq \text{spot} \leq \text{last}$) are to be shifted one position so that a new entry can be inserted at `a[spot]`. Which of the following statements correctly shifts the entries? (Assume that `last < (a.length - 1)`)

- (A)

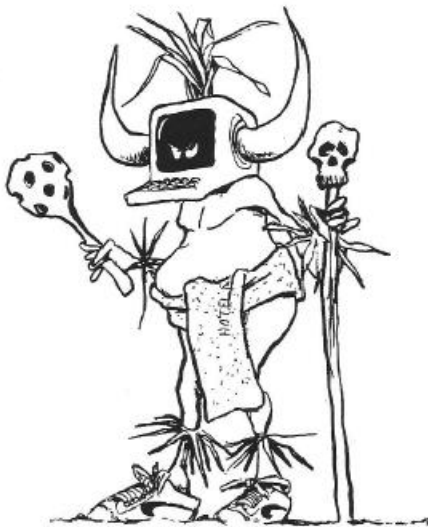
```
for (int i = spot; i <= last; i++)  
    a[i+1] = a[i];
```
- (B)

```
for (int i = last; i >= spot; i--)  
    a[i+1] = a[i];
```
- (C)

```
for (int i = spot+1; i <= last; i++)  
    a[i+1] = a[i];
```
- (D)

```
for (int i = last-1; i >= spot; i--)  
    a[i+1] = a[i];
```
- (E)

```
for (int i = spot; i <= last-1; i++)  
    a[i+1] = a[i];
```



1999 #6

Consider the following code segment.

```
int[][] m = new int[4][4];
m = initialize(); // initialize() will initialize its elements
int sum = 0;
int k;

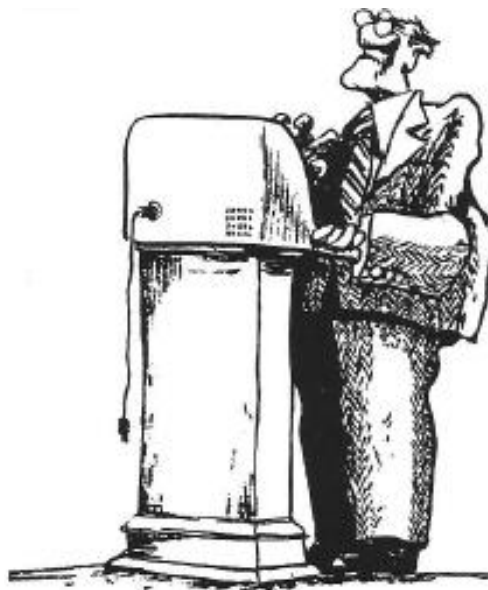
for (k = 0; k < m.length; k++)
{
    sum += m[k][m.length - k - 1];
}
```

Assume that after the call to `initialize`, `m` represents the matrix shown below

	0	1	2	3
0	1	1	1	1
1	1	2	3	4
2	2	2	2	2
3	2	4	6	8

What value will `sum` contain after the code segment is executed?

- (A) 4
- (B) 8
- (C) 13
- (D) 20
- (E) 42



Consider the following code segment.

```
int row, col;
int sum = 0;
int[][] a;
a = initialize();    // sizes a and initializes its elements

for (row = 0; row < a.length; row++)
{
    for (col = 0; col < a[0].length; col++)
    {
        sum += a[row][col];
    }
}
```

Which of the following best describes the result of executing the code segment?

- (A) Each element in the two-dimensional array `a` contains the value 0.
- (B) Each element in the two-dimensional array `a` contains the sum of its row number and its column number
- (C) Each element in the two-dimensional array `a` contains the sum of all preceding elements in two-dimensional array `a`.
- (D) The variable `sum` contains the sum of the values in the two-dimensional array `a`.
- (E) The variable `sum` contains the value `row * col`.



Consider the following code.

```
int[][] m;
m = initialize();    // sizes m to be a square matrix and
                     // initializes its elements
```

Which of the following code segments correctly sets a diagonal of the two-dimensional array `m` to contain all zeroes?

- (A)

```
int row = 0;
int col = 0;
while ((row < m.length) && (col < m[0].length)){
    m[row][col] = 0;
    row ++;
    row = col;
}
```
- (B)

```
int row = 0;
int col = 0;
while (row < m.length){
    m[row][col] = 0;
    row++;
    col = row;
    col++;
}
```
- (C)

```
int row = 0;
int col = 0;
while (row < m.length){
    m[row][col] = 0;
    row++;
}
```
- (D)

```
int row;
for (row = 0; row < m.length; row++){
    m[row][row] = 0;
}
```
- (E)

```
int row;
for (row = 1; row <= m.length; row++){
    m[row][row] = 0;
}
```

For each hour of the day, a weather station records temperature using integer values, and pressure, wind speed, and wind direction using values of type `double`. Of the following definitions of `dailyRecord`, which would be most suitable for recording these weather readings for one day?

- (A) `int[][] dailyRecord = new int[24][4];`
- (B) `int[] dailyRecord = new int[96];`
- (C)

```
public class WeatherInfo {
    public int temperature;
    public double pressure;
    public double windSpeed;
    public double windDir;
// constructors and other public and private methods and
// data not shown
}
WeatherInfo[] dailyRecord = new WeatherInfo[24];
```
- (D)

```
public class WeatherInfo {
    public void setTemperature(int temp){...}
    public void setPressure(double pressure){...}
    public void setWindSpeed(double speed){...}
    public void setWindDir(double dir){...}
// constructors and other public and private methods
// and data not shown
}
WeatherInfo[] dailyRecord = new WeatherInfo[24];
```
- (E)

```
public class WeatherInfo {
    private int temperature;
    private double pressure;
    private double windSpeed;
    private double windDir;
// constructors and other public and private methods
// and data not shown
}
WeatherInfo[] dailyRecord = new WeatherInfo[24];
```

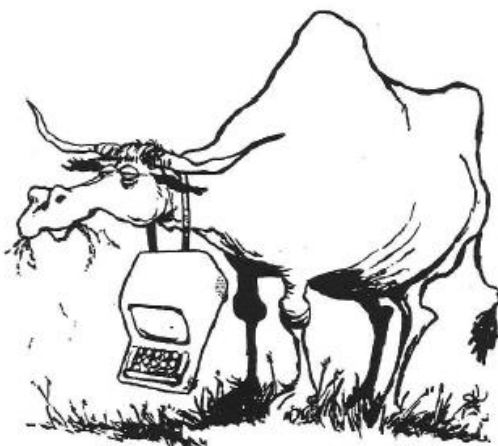
Consider designing a data structure to represent the positions of 50 game pieces on a 100 x 100 game board. (The position of a game piece is the row and column number of the square that it is on.) Two alternatives are described below.

Method 1: Use a two-dimensional array of `boolean` values indexed by row and column number, where each array element represents one square of the game board. If there is a game piece on that square, then the array element is `true`; otherwise, the array element is `false`.

Method 2: Use a one-dimensional array in which each element represents the position of one game piece (i.e., the row and column number of the square that it is on).

Which of the following is true?

- (A) Method 1 is not suitable if two game pieces can occupy the same square of the gameboard.
- (B) Method 2 is not suitable if two game pieces can occupy the same square of the gameboard.
- (C) Printing the positions of all game pieces can be done more efficiently by using Method 1 than by using Method 2.
- (D) Determining whether there is a game piece on a particular square (given the row and column numbers) can be done more efficiently by using Method 2 than by using Method 1.
- (E) Removing the game piece from a particular square (given the row and column numbers) can be done more efficiently by using Method 2 than by using Method 1.



Assume that methods `rowSum` and `colSum`, declared below, have been implemented correctly.

```
public int rowSum(int[][] a, int k) {...}
// precondition: a is a square matrix, 0 < k < a.length
// postcondition: returns the sum of the elements in row k
//               of array a

public int colSum(int[][] a, int k) {...}
// precondition: a is a square matrix, 0 < k < a[0].length
// postcondition: returns the sum of the elements in column
//               k of array a
```

Consider the partially written method `magicSquare`, shown below. `magicSquare` should return `true` if and only if every row and every column of its parameter `a` sums to the same value.

```
// precondition: a is a square matrix
public boolean magicSquare(int[][] a){
    int k;
    int sum = rowSum(a, 0);
    for (k = 0; k < a.length; k++)    {
        if (<condition>){
            <statement 1>
        }
    }
    <statement 2>
}
```

Which of the following could be used to replace `<condition>`, `<statement 1>`, and `<statement 2>` so that `magicSquare` will work as intended?

- | | <code><condition></code> | <code><statement 1></code> | <code><statement 2></code> |
|-----|---|----------------------------------|----------------------------------|
| (A) | <code>((rowSum(A, k) != sum) (colSum(A, k) != sum))</code> | <code>return false;</code> | <code>return true;</code> |
| (B) | <code>((rowSum(A, k) != sum) && (colSum(A, k) != sum))</code> | <code>return false;</code> | <code>return true;</code> |
| (C) | <code>((rowSum(A, k) != sum) && (colSum(A, k) != sum))</code> | <code>return true;</code> | <code>return false;</code> |
| (D) | <code>((rowSum(A, k) == sum) (colSum(A, k) == sum))</code> | <code>return true;</code> | <code>return false;</code> |
| (E) | <code>((rowSum(A, k) == sum) && (colSum(A, k) == sum))</code> | <code>return true;</code> | <code>return false;</code> |

1988#14, 15

Refer to the following program segment:

```
public static final int SIZE = 10;
public boolean yesOrNo (String[][] grid, int row, int colm, String mark)
{
    int i, count=0;
    boolean ok;
    for (i = 0; i < SIZE; i++)
        if (grid[i][colm].equals(mark))
            count++;
    ok = (count == SIZE);
    count = 0;
    for (i=0; i < SIZE; i++)
        if (grid[row][i].equals(mark) )
            count++;
    return (ok || (count == SIZE));
}
```

14. Which of the following conditions on a matrix `g` of type `String[][]` will by itself guarantee that

`yesOrNo (g, 0, 0, "*")` will have the value `true` when evaluated? (Assume the declaration `String[][] g = new String[SIZE][SIZE];`)

- I. the element in the first row and first column is “*”
- II. all elements in both diagonals are “*”
- III. all elements in the first column are “*”

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

15. Which of the following best describes what the method `yesOrNo` does?

- (A) counts the occurrences of `mark` in `grid` and returns `true` when the `count` is less than `SIZE`
- (B) determines whether there are exactly `SIZE` occurrences of `mark` in `grid`
- (C) determines whether it is true that all entries in row `row` are equal to `mark` or all entries in column `colm` are equal to `mark`
- (D) determines whether it is true that all entries in row `row` and all entries in column `colm` are equal to `mark`
- (E) counts the occurrences of `mark` on the main diagonal of `grid` and returns `true` when `count` equals `SIZE`

Suppose that `grid` is an n -by- n 2D array of integers, and a method is to be used to reflect the entries across the main diagonal. (The main diagonal goes from the upper left corner to the lower right corner). For example, if $n = 3$ and the 2D array contained

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

the method would change it to

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

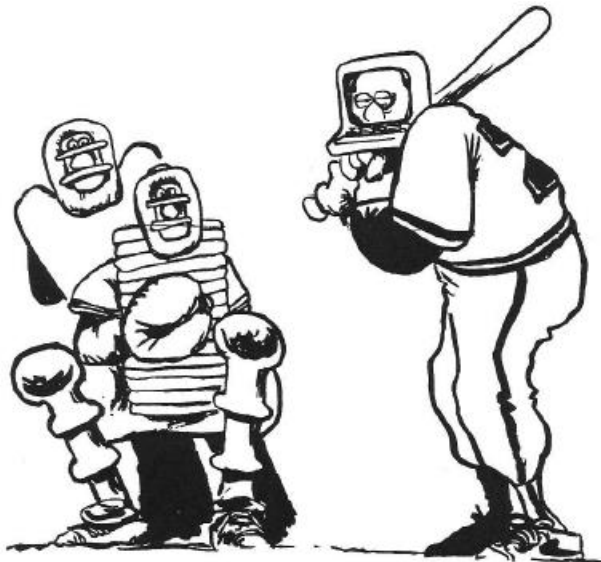
If `swap(grid, row, colm)` interchanges `grid[row][colm]` with `grid[colm][row]`, which of the following will NOT have the desired effect?

- (A) `for (int row = 0; row < n; row++)`
 `for (int colm = 0; colm < n; colm++)`
 `swap(grid, row, colm);`
- (B) `for (int row = 0; row < n; row++)`
 `for (int colm = 0; colm < row; colm++)`
 `swap(grid, row, colm);`
- (C) `for (int row = 1; row < n; row++)`
 `for (int colm = 0; colm < row; colm++)`
 `swap(grid, row, colm);`
- (D) `for (int row = 0; row < n - 1; row++)`
 `for (int colm = row + 1; colm < n; colm++)`
 `swap(grid, row, colm);`
- (E) `for (int row = 0; row < n - 1; row++)`
 `for (int colm = row; colm < n; colm++)`
 `swap(grid, row, colm);`

1999 #15

Assume that an array contains 100 integers sorted in increasing order. Two alternatives to search the array for a particular integer are a sequential and a binary search. When searching for a value that is in the array, which of the following best characterizes the greatest number of items in the array that will be examined during each kind of search?

	Sequential	Binary
(A)	100	1
(B)	100	7
(C)	100	50
(D)	50	7
(E)	50	25



A list of integers can be stored sequentially in an array. The list can be maintained in sorted order. Maintaining the list in sorted order in an array leads to inefficient execution for which of the following operations?

- I. Inserting and deleting elements
 - II. Printing the list
 - III. Computing the average of the elements
- (A) I only
 - (B) II only
 - (C) III only
 - (D) I and III only
 - (E) I, II, and III



An algorithm for searching a large sorted array for a specific entry x compares every fourth item in the array to x until it finds one that is larger than or equal to x . Whenever a larger item is found, the algorithm examines the preceding three entries. If the array is sorted smallest to largest, which of the following describes all cases when this algorithm might use fewer comparisons to find x than would a binary search?

- (A) it will never use fewer comparisons
- (B) when x is very close to the beginning of the array
- (C) when x is in the middle position in the array
- (D) when x is very close to the end of the array
- (E) it will always use fewer comparisons



The following declaration and incomplete method is intended to sort an array of unique integers in increasing order using the quicksort algorithm.

```
public void partition(int[]a, int first, int last, int
                    pivotPos);

public void quickSort(int[]a, int first, int last)
{
    int pivotPos;

    // If the subarray has at least 2 elements, partition
    //    and recursively sort the two partitions.

    if (first < last)
    {
        partition(a, first, last, pivotPos);
        <statements>
    }
}
```

The variables `first` and `last` are the indices of the first and last elements in the subarray of array `a` to be sorted. The method `partition` performs the task of splitting the array into two subarrays around a pivot point, `pivotPos`, chosen by partition. After the call to `partition`, the subarray from `a[first]` to `a[pivotPos - 1]` contains integers that are less than `a[pivotPos]`, and the subarray from `a[pivotPos + 1]` to `a[last]` contains integers greater than `a[pivotPos]`. The element `a[pivotPos]` is in its final sorted position.

Which of the following can be used to replace `<statements>` so that `quickSort` will work as intended?

- (A) `quickSort(a, first, pivotPos + 1);`
`quickSort(a, pivotPos - 1, last);`
- (B) `quickSort(a, last, pivotPos);`
`quickSort(a, pivotPos, first);`
- (C) `quickSort(a, first, pivotPos - 1);`
`quickSort(a, last, pivotPos + 1);`
- (D) `quickSort(a, first, pivotPos - 1);`
`quickSort(a, pivotPos + 1, last);`
- (E) `quickSort(a, pivotPos - 1, first);`
`quickSort(a, last, pivotPos + 1);`

Consider the following declarations:

```
public int search(int[] list, int key)
{
    //precondition:
    //postcondition:
    1) returns i such that  $0 \leq i < \text{list.length}$ 
    2) if  $i \geq 0$  is returned, then  $\text{list}[i] = \text{key}$ 
    3) if -1 is returned, then  $\text{key} \neq \text{list}[i]$  for all  $i < \text{list.length}$ 

    int i=0;

    while (i < list.length && list[i] < key)
        i++;
    if (list[i] == key)
        return i;
    else
        return -1;
}
```

Which of the following should be added to the precondition of search?

- (A) the value of key appears at least once in list
- (B) the value of key does not appear twice in list
- (C) list is sorted smallest to largest
- (D) list is sorted largest to smallest
- (E) list is unsorted



Merging two sorted lists to yield a single sorted list requires which of the following?

- (A) That the two lists be external files
- (B) That recursive techniques be used
- (C) That at least one list be stored in memory
- (D) That non-recursive techniques be used
- (E) None of the above



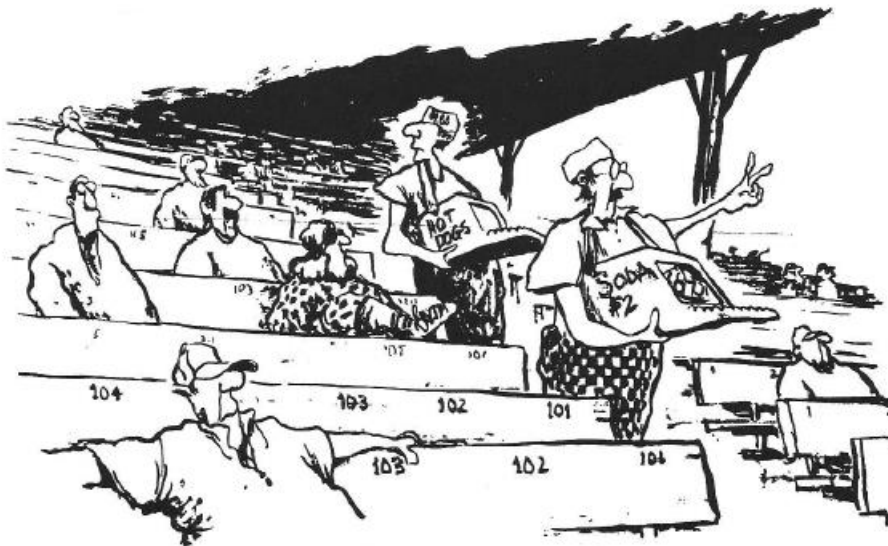
1999 #4

Consider the following method.

```
public int something(int a, int b)
{
    if (b <= 1)
    {
        return a;
    }
    else
    {
        return something(a, b-1);
    }
}
```

What value is returned by the call `something(4, 6)`?

- (A) 4
- (B) 6
- (C) 24
- (D) 1296
- (E) 4096



Consider the incomplete method `powerOf` given below.

The call `powerOf(n, x)` should return the quantity n^x .

```
public int powerOf(int base,int power)
// precondition: power >= 1
// postcondition: returns basepower
{
    int result;

    if (<expression1> == 1)
    {
        result = <expression2>;
    }
    else
    {
        result = <expression3> * powerOf(base, power - 1);
    }
    return result;
}
```

Which of the following could be used to replace *<expression1>*, *<expression2>*, and *<expression3>* so that `powerOf` will work as intended?

	<i><expression1></i>	<i><expression2></i>	<i><expression3></i>
(A)	power	base	result
(B)	power	base	power
(C)	power	base	base
(D)	base	power	result
(E)	base	power	base



Consider the following incomplete method.

```
public int mystery(int k)
{
    if (k <= 0)
    {
        return 0;
    }
    else
    {
        return <missing code>;
    }
}
```

Which of the following could be used to replace <missing code> so that the value of `mystery(5)` is 15?

- (A) `k + mystery(k - 1)`
- (B) `k * mystery(k - 1)`
- (C) `mystery(k - 1)`
- (D) `mystery(k + 1)`
- (E) `mystery(k - 1) * mystery(k + 1)`



Consider the following method.

```
public void print(int count)
{
    int k;
    if (count > 0)
    {
        k = IO.readInt();    // reads an integer
        print(count - 1);
        System.out.println(k);
    }
}
```

Of the following, which text best describes what is printed as a result of the call `print(10)`?

- (A) Nothing is printed because a run-time error occurs.
- (B) Nothing is printed because the if condition never evaluates to true.
- (C) Ten integers are printed in the same order in which they were read.
- (D) Ten integers are printed in the reverse order in which they were read.
- (E) Only the non-zero values that were read are printed: they are printed in the same order in which they were read.



1992 # 9, 10

Refer to the following method.

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

9.

What value does `answer(5)` return?

- (A) 2
- (B) 8
- (C) 10
- (D) 32
- (E) 120

10.

If n is a positive integer, how many times will `answer` be called to evaluate `answer(n)` (including the initial call)?

- (A) 2
- (B) n
- (C) $2n$
- (D) n^2
- (E) 2^n

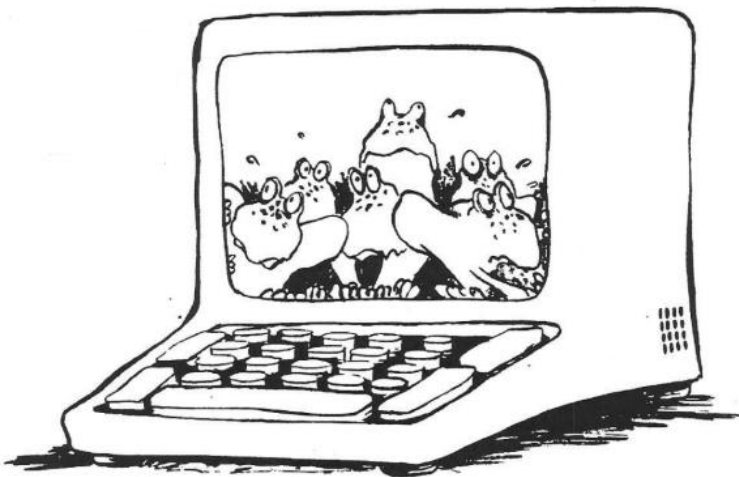


What happens when the program segment shown below is compiled and executed?

```
public int recurse(int x)
{
    x = 3;
    if (x==0)
        return 0;
    else
    {
        x --;
        return x + recurse(x);
    }
}
```

```
// this statement is located in some method in the same class as recurse()
System.out.print(recurse(0));
```

- (A) The program writes the value 5.
- (B) The program writes the value 6.
- (C) The program fails to compile because of illegal use of `recurse` on the right-hand side of an assignment statement.
- (D) The program fails to terminate because of infinite recursion.
- (E) The program fails to compile because the value returned by a method cannot be the actual parameter of a `System.out.print` call.



Consider the following method.

```
public void mystery(int n)
{
    if (n > 2)
        mystery (n % 3);
    System.out.print((n / 3) + " " );
}
```

The method call `mystery(38)` will yield as output which of the following sequences of numbers?

- (A) 0 12
- (B) 12 0
- (C) 1 1 0 2
- (D) 1 1 1 1
- (E) 2 0 1 1



1988 # 17, 18

Refer to the following method:

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

17.

What value is returned by `whatIsIt(4, 4)` ?

- (A) 8
- (B) 16
- (C) 24
- (D) 64
- (E) 256

18.

Which of the following is a necessary and sufficient condition for the method `whatIsIt` to return a value if it is assumed that the values of `n` and `x` are small in magnitude?

- (A) $n > 0$
- (B) $n \geq 0$
- (C) $n > 0$ and $x > 0$
- (D) $x \leq n$ and $n > 0$
- (E) $n \leq x$ and $n > 0$



1988 # 20

```
public void wow (int n)
{
    if (n > 1)
        wow (n / 2);
    System.out.print(n + " ");
}
```

The method call `wow (16)` will yield as output which of the following sequences of numbers?

- (A) 10 8 6 4 2
- (B) 16 8 4 2 1
- (C) 1 2 4 8 16
- (D) 32 16 8 4 2
- (E) 2 4 8 16 32

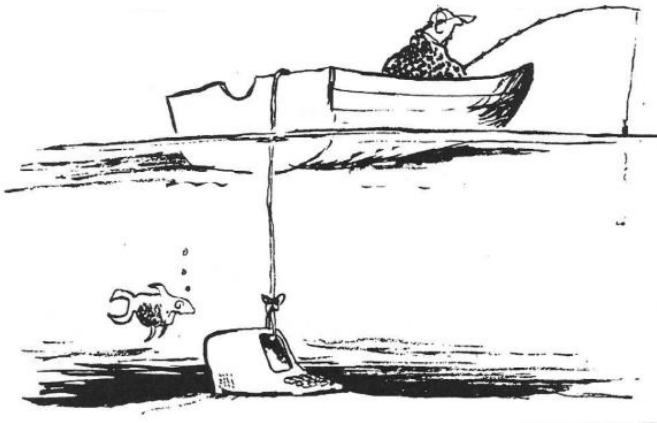


Consider the following method.

```
public int mult(int x, int y)
    //precondition:  x > 0
    //postcondition: returns x* y
{
    if (x == 1)
        <statement 1>
    else
        <statement 2>
}
```

Which of the following statement pairs properly completes the method?

- | | <statement 1> | <statement 2> |
|-----|---------------|---------------------------|
| (A) | return x * y | <none> |
| (B) | return y | return mult(x - 1, y + 1) |
| (C) | return y | return mult(x - 1, y + y) |
| (D) | return y | return mult(x - 1, y) + y |
| (E) | return y | return mult(x - 1, y) * y |



ANSWERS

1988 Answers

1	C	2	E	3	C	4	D	5	C
6	C	7	D	8	D	9	A	10	E
11	B	12	E	13	E	14	B	15	C
16	C	17	E	18	A	19	B	20	C
21	E	22	E	23	D	24	D	25	E
26	A	27	D	28	B	29	B	30	A
31	E	32	D	33	A	34	B	35	C

1992 Answers

1	E	2	A	3	D	4	C	5	E
6	B	7	D	8	E	9	D	10	B
11	B	12	D	13	A	14	D	15	B
16	D	17	A	18	B	19	A	20	C
21	E	22	E	23	D	24	A	25	E
26	B	27	E	28	D	29	C	30	D
31	C	32	E	33	C	34	B	35	B
36	C	37	E	38	D	39	D	40	A

1999 Answers

1	C	2	D	3	C	4	A	5	A
6	B	7	D	8	E	9	E	10	C
11	B	12	B	13	A	14	A	15	B
16	E	17	A	18	D	19	A	20	C
21	E	22	A	23	D	24	D	25	D
26	C	27	E	28	D	29	C	30	E
31	C	32	E	33	A	34	D	35	E
36	D	37	C	38	C	39	B	40	A