1) ITERATION

n = 4;
for XXX1XXX
    for XXX2XXX
        fprintf('%1d', XXX3XXX);
    end
    fprintf('
');
end

Which code fragments complete the method such that the output is:

1
22
333
4444

a) XXX1XXX = index = 0:n
   XXX2XXX = jndex = 1:index
   XXX3XXX = index

b) XXX1XXX = index = 0:n
   XXX2XXX = jndex = 1:index
   XXX3XXX = jndex

c) XXX1XXX = index = 1:n
   XXX2XXX = jndex = 1:index
   XXX3XXX = index

d) XXX1XXX = index = 1:n
   XXX2XXX = jndex = 0:index
   XXX3XXX = index

e) XXX1XXX = index = 1:n
   XXX2XXX = jndex = 0:index
   XXX3XXX = jndex
2) CONDITIONAL

```c
x = 2;
y = 16;

if ( 1 && 0 ) || 1
    if  x < y
        x = x * x;
    else
        y = x + y;
    end
end
if ( 0 || 1 ) && 0
    if x <= y
        x = x + 1;
    else
        y = y + y;
    end
else
    x = x * x;
    if x <= y
        y = x + y;
    end
end

fprintf('x = %f
', x);
fprintf('y = %f
', y);
```

What is the output of this code fragment?

a) x = 4
   y = 22

b) x = 3
   y = 19

c) x = 4
   y = 20

d) x = 4
   y = 23

e) x = 16
   y = 32
3) BST

Assume that a BSTNode class has been defined with fields:

    data
    left
    right

and methods:

    getData
    getLeft
    getRight

Further assume that a BST class had been defined with a root field and a getRoot method.

Now examine the following methods of the BST class:

function traverse(bst)
    traverse_helper(getRoot(bst));
    fprintf('
');

function traverse_helper(current)
    if ~isempty(current)
        fprintf('%f ', getData(current));
        traverse_helper(getLeft(current));
        traverse_helper(getRight(current));
    end

What is the output of the “traverse” method on the above binary search tree (BST), where 8 is passed as the root?

a)  3
    5
    4
    2
    6
    12
    15
    13
    12
    8

b)  2
    3
    4
    5
    6
    12
    13
    15
    3
    5

c)  8
    6
    12
    2
    15
    4
    13
    3
    5
    3

d)  8
    6
    2
    4
    3
    5
    12
    15
    13
    3

e)  8
    12
    15
    13
    6
    2
    4
    3
    5
    3
4) ARRAY

array1 = [ 4, 5, 3, 6, 2, 7, 1 ];
array2 = [ 7, 4, 2, 1 ];
array1(4) = array1(6);
array1(3) = array2(3);
array1(5) = array2(4) + 5;
array1(7) = array1(4);
if array1(2) > array2(2)
    array1(2) = array1(2) + 2;
end

What is the value of array1 after this code is executed?

a) [ 4, 4, 2, 7, 2, 2, 1 ]
b) [ 4, 7, 3, 6, 7, 7, 7 ]
c) [ 4, 7, 2, 7, 6, 7, 7 ]
d) [ 4, 7, 2, 7, 2, 2, 1 ]
e) [ 4, 7, 2, 6, 6, 7, 6 ]
5) HASH TABLE

Which is the resulting Hash Table that is made when the following < key, data > pairs are added? The hash code form of the key is given in the parenthesis after the pair. The HashTable is of size 5 and uses external chaining.

< “Fluffy”, “Cat” > ( 54 )
< “Fido”, “Dog” > ( 33 )
< “Mr. Bubbles”, “Fish” > ( 79 )
< “Bugs”, “Rabbit” > ( 12 )
< “Robbie”, “Robot” > ( 10 )
< “Puff”, “Dragon” > ( 25 )
< “Tux”, “Penguin” > ( 31 )

a) Fluffy
    Fido
    Mr.Bubbles
    Bugs
    Robbie

d) Robot -> Cat
    Rabbit -> Fish
    Dragon
    Dog
    Penguin

b) Robot -> Dragon
    Penguin
    Rabbit
    Dog
    Cat -> Fish

e) Fish -> Cat
    Dog
    Rabbit
    Penguin
    Dragon -> Robot

c) Robbie -> Puff
    Tux
    Bugs
    Fido
    Fluffy -> Mr.Bubbles
6) ELEMENT COMPARISON FOR SORTING

The following function “isSorted” should return true if the vector “ x “ is sorted in ascending order. Otherwise, the function should return false:

function result = isSorted( x )
    %missing code

Which of the following code fragments is the missing code?

a) result = 1;
   for in = 1:length(x)-1
       if x(in) > x(in + 1)
           result = 0;
       else
           result = 1;
       end
   end

b) result = 1;
   for in = 1:length(x)-1
       if x(in) > x(in+1)
           result = 0;
       end
   end

c) result = 0;
   for in = 1: length(x)-1
       if x(in) > x(in+1)
           result = 0;
       end
   end

d) b = 0;
   for in = 0:length(x)-1
       if x(in) > x(in+1)
           result = 1;
       end
   end

e) result = 0;
   for in = 0:length(x)+1
       if x(in) > x(in):
           result = 1;
       end
   end
Given the following class definition:

```matlab
function result = Trace(value)
    if nargin == 0
        result.myNumber = 0;
        result = class(result, 'Trace');
    elseif isa(value, 'Trace')
        result = value;
    else
        result.myNumber = value;
        result = class(result, 'Trace');
    end

function ret = getMyNumber(obj)
    ret = obj.myNumber;

function setMyNumber(obj, n)
    obj.myNumber = n;
    assignin('caller', inputname(1), obj);

function ret = char(obj)
    ret = ['Trace Object containing ' num2str(myNumber)];

function display(obj)
    disp(char(obj));

function myFunction()
    a = Trace(5);
    b = Trace(6);
    c = Trace(-3);
    myArray{1} = b;
    myArray{2} = c;
    myArray{3} = a;
    for index = myArray
        disp(getMyNumber(index{1}));
    end
    setMyNumber(b, getMyNumber(c));
    setMyNumber(a, getMyNumber(c));
    b
    fprintf('%s
', char(myArray{1}));
```
What is the output when the function “myFunction” (on the preceding page) is called?

a) 5
   6
   -3
   -3
   6

b) 5
   6
   -3
   Trace Object containing -3
   Trace Object containing 6

c) 6
   -3
   5
   Trace Object containing 6
   Trace Object containing 6

d) 6
   -3
   5
   Trace Object containing -3
   Trace Object containing 6

e) Trace Object containing 6
   Trace Object containing -3
   Trace Object containing 5
   -3
   6
8) LOOPING

```
x = [ 2, 1, 4, 5, 7 ];
limit = 7;
index = 1;
sum = 0;
while sum < limit && index < length(x)
    sum = sum + x(index);
    index = index + 1;
end
```

What is the value of the variable "index" after the code is executed?

a) 1
b) 3
c) 4
d) 5
e) 7
9) SORTING

The following function “sortArray” should return an array “x” sorted in ascending order:

function ret = sortArray(x)
    %missing code

Which of the following code fragments is the missing code?

a) for index = 1:length(x)
    min = index;
    for jndex = index+1: length(x)
        if x(jndex) < x(min)
            min = jndex;
        end
    end
    temp = x(min);
    x(min) = x(index);
    x(index) = temp;
end
ret = x;

b) for index = 1:length(x)
    min = index;
    for jndex = index+1: length(x)
        if x(min) < x(jndex)
            min = jndex;
        end
    end
    temp = x(min);
    x(min) = x(index);
    x(index) = temp;
end
ret = x;

c) for index = 1:length(x)
    min = index;
    for jndex = index+1: length(x)
        if x(jndex) < x(min)
            min = jndex;
        end
    end
    temp = x(min);
    x(index) = x(min);
    x(index) = temp;
end
ret = x;
d) for index = 1:length(x)
    min = index;
    for jndex = index+1: length(x)
        if x(min) < x(jndex)
            min = jndex;
        end
    end
    temp = x(min);
    x(index) = x(min);
    x(index) = temp;
end
ret = x;


e) for index = 1:length(x)
    min = index;
    for jndex = index: length(x)
        if x(jndex) < x(min)
            min = jndex;
        end
    end
    temp = x(min);
    x(index) = x(min);
    x(index) = temp;
end
ret = x;
10) ITERATION

n = 5
for XXX1XXX
  sum = 0;
  fprintf( 'sum ( ' );
  for XXX2XXX
    sum = sum + jndex;
    fprintf('%d ', XXX3XXX);
  end
  fprintf(') = %d
', XXX4XXX);
end

Which code fragments complete the method such that the output is:

sum ( 1 ) = 1
sum ( 1 2 ) = 3
sum ( 1 2 3 ) = 6
sum ( 1 2 3 4 ) = 10
sum ( 1 2 3 4 5 ) = 15

a) XXX1XXX = index = 1:n
   XXX2XXX = jndex = 0:index
   XXX3XXX = jndex
   XXX4XXX = sum

b) XXX1XXX = index = 1:n
   XXX2XXX = jndex = 1:index
   XXX3XXX = jndex
   XXX4XXX = sum
c) XXX1XXX = index = 0:n
   XXX2XXX = jndex = 0:index
   XXX3XXX = jndex
   XXX4XXX = sum
d) XXX1XXX = index = 1:n
   XXX2XXX = jndex = 1:index
   XXX3XXX = sum
   XXX4XXX = jndex
e) XXX1XXX = index = 0:n
   XXX2XXX = jndex = 1:index
   XXX3XXX = sum
   XXX4XXX = sum
function ret = tryMe( s, c, value)
    if length(s) == 0
        ret = value;
    elseif s(1) == c
        value = value * 2;
        ret = tryMe( s(2:end), c, value );
    else
        value = value + 1;
        ret = tryMe( s(2:end), c, value );
    end
end

What is the value returned by this method call?
tryMe( 'mississippi', 'i', 1)

a) 60
b) 213
c) 44
d) 12
e) 30
12) POLYMORPHISM

Given the following class hierarchy:

```java
public abstract class Emotion
    has: public void express()

public interface Crying
    has: public void tears()

public class Joy extends Emotion implements Crying
    has: public void smile()
    has: public void tears()

public class PureJoy extends Joy
    has: public void exult()

public class Anger extends Emotion
    has: public void yell()
```

Determine whether the following statements will compile and run without errors.
If no errors write OK.
If compile error write COMP
If run time error write RUN

1) Emotion e = new Emotion();
2) Emotion my = new Joy();
3) Emotion great = new PureJoy();
   great.tears();
4) Emotion well = new Joy();
   (Anger)well.yell();
5) PureJoy lastone = new Joy();

<table>
<thead>
<tr>
<th>a) OK</th>
<th>b) COMP</th>
<th>c) COMP</th>
<th>d) OK</th>
<th>e) COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>COMP</td>
</tr>
<tr>
<td>RUN</td>
<td>OK</td>
<td>RUN</td>
<td>RUN</td>
<td>OK</td>
</tr>
<tr>
<td>OK</td>
<td>RUN</td>
<td>OK</td>
<td>COMP</td>
<td>OK</td>
</tr>
<tr>
<td>OK</td>
<td>COMP</td>
<td>OK</td>
<td>OK</td>
<td>COMP</td>
</tr>
</tbody>
</table>
13) SORTING

The following function “sortArray” should return an array “x” sorted in ascending order:

function ret = sortArray(x)

    %missing code

Which of the following code fragments is the missing code?

a) for index = 1:length(x)
    key = x(index);
    position = index;
    while (position > 1) && (key < x(position1))
        x(position) = x(position-1);
        position = position - 1;
    end
    x(position) = key;
end
ret = x;

b) for index = 1:length(x)
    key = x(index);
    position = index;
    while (position > 1) && (key < x(position1))
        x(position) = x(position-1);
        position = position - 1;
    end
    key = x(position);
end
ret = x;

c) for index = 1:length(x)
    key = x(index);
    position = index;
    while (position > 1) && (key < x(position-1))
        x(position) = x(position-1);
        position = position - 1;
    end
    key = x(position);
end
ret = x;
d) for index = 1:length(x)
    key = x(index);
    position = index;
    while (position > 1) && (key < x(position-1))
        x(position) = x(position-1);
        position = position - 1;
    end
    x(position) = key;
end
ret = x;

e) for index = 1:length(x)
    key = x(index);
    position = index;
    while (position > 1) && (key < x(position-1))
        x(position-1) = x(position);
        position = position - 1;
    end
    x(position) = key;
end
ret = x;
14) SEARCHING

array = [ 0, 1, 2, 3, 5, 9, 4, 8 ];

index = 1;

while index < (length( array ) ) && array(index+1) > array(index)
    index = index + 1;
end

disp(index)

What is the output of the code above?

  a) 5
  b) 6
  c) 8
  d) 4
  e) 9
15) DYNAMIC BINDING

Class Temp has
  function printer(obj)
    disp('Printer method in Temp class')

Class myTemp (subclass of Temp)
  function printer(obj)
    disp('Printer method in myTemp class')

Class aTemp (subclass of myTemp)
  function printer(obj)
    disp('Printer method in aTemp class')

  t = Temp;
  printer(t);
  at = aTemp;
  t = at;
  printer(t);
  mt = myTemp;
  printer(mt);
  mt = t;
  printer(mt);

What is the output of the preceding code?

a) Printer method in Temp class
   Printer method in Temp class
   Printer method in myTemp class
   Printer method in myTemp class

b) Printer method in Temp class
   Printer method in aTemp class
   Printer method in Temp class
   Printer method in aTemp class

c) Printer method in aTemp class
   Printer method in Temp class
   Printer method in Temp class
   Printer method in aTemp class

d) Printer method in aTemp class
   Printer method in aTemp class
   Printer method in myTemp class
   Printer method in myTemp class

e) Printer method in Temp class
   Printer method in aTemp class
   Printer method in myTemp class
   Printer method in aTemp class
16) LINKED LIST

Class ListNode Constructor

function ret = ListNode(data, next)
    if nargin == 0
        ret.data = 0;
        ret.next = [];
        ret = class(ret, 'ListNode');
    elseif isa(data, 'ListNode')
        ret = data;
    elseif nargin == 1
        ret.data = data;
        ret.next = [];
        ret = class(ret, 'ListNode');
    else
        ret.data = data;
        ret.next = next;
        ret = class(ret, 'ListNode');
    end

Assume that position is an object of class ListNode. Assume that head is the beginning of the linked list. Also assume that there is a getNext method.

Which of the following class changes position so that it is referencing the next item in the linked list and deletes the node at the beginning of the linked list?

a) position = getNext(position);
   position = getNext(head);
   head = position;

b) getNext(position) = getNext(head);
   head = getNext(position);

c) position = getNext(head);
   head = position;

d) position = getNext(head);
   head = getNext(position);

e) position = head;
   head = position;