The exam consists of 20 multiple choice questions. For each question you are asked to select a single answer, i.e. the best / most complete answer. Each question is worth 5 points.
CS177 Spring 2012 Midterm 2 – April 9th

/*
* Last name, First name:
* Your Purdue mail account:
* Your Purdue ID:
* Your recitation section:
* Signature:

INSTRUCTIONS

Fill in the header of the FIRST page of the assignment text with the required information:

- Your Last Name and First Name
- Your purdue mail account (e.g. lmartino)
- Your Recitation Section: find your recitation section in the table below, and put the value of the last column in the header
- Your Purdue ID
- Signature: your signature

Fill in the bubble sheet fields as follows:

- student ID: use the 10 digit ID number on your student ID card. DO NOT USE YOUR SOCIAL SECURITY NUMBER!
- Last Name and First Name
- Test/Quiz: put 02
- Course: 177
- Section number: find your recitation section in the table below and put in the scantron card the value of the last column

<table>
<thead>
<tr>
<th>Recitation Section</th>
<th>Day and Time</th>
<th>TA</th>
<th>Recitation section to be used in the header and in the scantron card</th>
</tr>
</thead>
<tbody>
<tr>
<td>R01</td>
<td>R 3:30PM</td>
<td>Bin Shen</td>
<td>0001</td>
</tr>
<tr>
<td>R02</td>
<td>F 1:30PM</td>
<td>Bin Shen</td>
<td>0002</td>
</tr>
<tr>
<td>R03</td>
<td>F 7:30</td>
<td>Gnana Surendra, Youhan Fang</td>
<td>0003</td>
</tr>
<tr>
<td>R04</td>
<td>R 8:30</td>
<td>Youhan Fang</td>
<td>0004</td>
</tr>
<tr>
<td>R05</td>
<td>F 3:30PM</td>
<td>Rajul Jain</td>
<td>0005</td>
</tr>
<tr>
<td>R06</td>
<td>R 9:30</td>
<td>Youhan Fang</td>
<td>0006</td>
</tr>
<tr>
<td>R07</td>
<td>F 2:30PM</td>
<td>Rajul Jain</td>
<td>0007</td>
</tr>
<tr>
<td>R08</td>
<td>F 10:30</td>
<td>Gnana Surendra</td>
<td>0008</td>
</tr>
<tr>
<td>R09</td>
<td>F 12:30PM</td>
<td>Gnana Surendra</td>
<td>0009</td>
</tr>
<tr>
<td>R10</td>
<td>R 7:30</td>
<td>Bin Shen, Rajul Jain</td>
<td>0010</td>
</tr>
</tbody>
</table>
/*
* Last name, First name:
* Your Purdue mail account:
* Your Purdue ID:
* Your recitation section:
* Signature:

1. Python is a high-level programming language
   a. *Yes
   b. No

2. Programs are written with specialized editors, for example Python programs can be written with the editor in the IDLE environment. Such specialized editors have the following advantages:
   a. They provide syntactic highlighting, making the code easier to read.
   b. They provide help with the syntax by providing suggestions according to what is currently typed.
   c. They help with program indentation.
   d. None of the above
   e. *a, b, and c

3. Which of the following statements regarding pseudocode are true:
   a. Pseudocode can be translated automatically, by a computer, into Python code.
   b. *Pseudocode can be translated easily, by a programmer, into Python code.
   c. Pseudocode is Python specific.
   d. a and c.
   e. b and c.
   f. None of the above.
4. What will the Python interpreter return if you type:

6 == \texttt{eval}('6')

a. ... NameError: name 'eval' is not defined
b. ... NameError: name '6' is not defined
c. True
d. False
e. c and d

5. What will the value of variable C be after the following Python code is executed:

\begin{verbatim}
A = 'class'
B = 'team'
\end{verbatim}

a. 'ttc'
b. 'lee'
c. 'ctt'
d. '*eel'
e. 310

6. Files are useful because:

a. Unlike main memory, they provide fast access to data.
b. *Unlike main memory, they continue to store data after the computer is powered off.
c. Unlike main memory, they store data in compact form.
d. a and c
e. a and b
CS177 Spring 2012 Midterm 2 – April 9\textsuperscript{th}

/*
 * Last name, First name:
 * Your Purdue mail account:
 * Your Purdue ID:
 * Your recitation section:
 * Signature:

7. You are asked to program the collision detection module of a 2-D computer game. In particular, you have to keep a slowly moving square (side $r$ and center coordinates $x$ and $y$) away from a square obstacle (side $s$ and center coordinates $x_0$ and $y_0$). What condition do you need to test:

   a. $(x > x_0 + s/2 + r/2 \text{ and } x < x_0 - s/2 - r/2) \text{ and } (y > y_0 + s/2 + r/2 \text{ and } y < y_0 - s/2 - r/2)$
   b. $(x > x_0 + s/2 + r/2 \text{ or } x < x_0 - s/2 - r/2) \text{ and } (y > y_0 + s/2 + r/2 \text{ or } y < y_0 - s/2 - r/2)$
   c. $(x > x_0 + s/2 + r/2 \text{ and } x < x_0 - s/2 - r/2) \text{ or } (y > y_0 + s/2 + r/2 \text{ and } y < y_0 - s/2 - r/2)$
   d. *[$(x > x_0 + s/2 + r/2 \text{ or } x < x_0 - s/2 - r/2) \text{ or } (y > y_0 + s/2 + r/2 \text{ or } y < y_0 - s/2 - r/2)$]*
   e. None of the above

8. A computer stores colors using three channels: red, green, and blue. Each channel is encoded with 8 bits. How many different gray levels can the computer display, including black and white?

   a. 8
   b. $2^{32}$
   c. *256
   d. Infinity
   e. None of the above

9. A computer stores colors using three channels: red, green, and blue. Each channel is encoded with 8 bits. How many different colors can the computer display?

   a. $8 \times 8 \times 8$
   b. $256 \times 256 \times 256$
   c. $2^{24}$
   d. Infinity
   e. *b and c
10. What will the following Python code do:

```python
for v in range(0, image.getHeight()):
    image.setPixel(10, v, 'red')
```

a. *It will draw a vertical red line on the image.
b. It will draw a horizontal red line on the image.
c. It will set the image to red.
d. It will set the top half of the image to red.
e. It will set the bottom half of the image to red.

11. What will the value of variable `a` be at the end of the following Python code?

```python
def MyAbs(a):
    if a < 0:
        a = -a
    a = -10
MyAbs(a)
```

a. *-10
b. 10
c. -20
d. 0
e. None of the above
/*
* Last name, First name:
* Your Purdue mail account:
* Your Purdue ID:
* Your recitation section:
* Signature:

12. What will the value of array A be at the end of the following Python code?

def MyMod(A):
    A[0] = 0
    A = [1, 2, 3]
MyMod(A)

a. [0, 1, 2]
b. [1, 2, 3]
c. *[0, 2, 3]
d. [0, 0, 0]
e. None of the above

13. Which of the following statements are true:
   a. If a break statement is reached in a loop, the current loop and all loops containing the current loop are exited.
   b. If a break statement is reached in a loop, execution is continued immediately after the break statement.
   c. *If a break statement is reached in a loop, the loop is exited.
   d. If a break statement is reached in a for loop, execution is continued with the next iteration of the for loop after incrementing the for loop index variable.
   e. None of the above

14. Recursion is:
   a. *A general approach for designing algorithms.
   b. An algorithm for solving data processing problems.
   c. The only way some data processing problems can be solved.
   d. a and c.
   e. None of the above.
Given a tree $T$ encoded as an array of triples $[a, b, c]$, where $a$ is the data stored by the node, $b$ is the index of the left child (-1 if child is missing) and $c$ is the index of the right child (-1 if child is missing), how can one test whether a node with index $currNode$ is a leaf?

b. $T[currNode][2] == -1$ or $T[currNode][3] == -1$
c. $T[currNode][1] == -1$ or $T[currNode][2] == -1$
d. $T[currNode][1] == -1$ and $T[currNode][2] == -1$
e. None of the above

16. What does the following Python code do?

```python
def MBTR(T, currNode):
    if T[currNode][1] == -1 and T[currNode][2] == -1:
        return 1
    left = 0
    if T[currNode][1] != -1:
        left = MBTR(T, T[currNode][1])
    right = 0
    if T[currNode][2] != -1:
        right = MBTR(T, T[currNode][2])
    return left + right
```

a. It returns 1, no matter what the tree is.
b. It returns 0, no matter what the tree is.
c. It counts the number of internal nodes in a tree.
d. It counts the nodes in a tree.
e. It counts the leaves in a tree.
17. Consider the following two Python functions (note the presence and absence of `elif` statements).

```python
def EvalAEBTR(T, currNode):
    left = T[currNode][1]
    right = T[currNode][2]
    if left == -1:
        return T[currNode][0]
    if T[currNode][0] == '+':
        return EvalAEBTR(T, left) + EvalAEBTR(T, right)
    elif T[currNode][0] == '-':
        return EvalAEBTR(T, left) - EvalAEBTR(T, right)
    elif T[currNode][0] == '*':
        return EvalAEBTR(T, left) * EvalAEBTR(T, right)
    elif T[currNode][0] == '/':
        return EvalAEBTR(T, left) / EvalAEBTR(T, right)
```

and

```python
def EvalAEBTR(T, currNode):
    left = T[currNode][1]
    right = T[currNode][2]
    if left == -1:
        return T[currNode][0]
    if T[currNode][0] == '+':
        return EvalAEBTR(T, left) + EvalAEBTR(T, right)
    if T[currNode][0] == '-':
        return EvalAEBTR(T, left) - EvalAEBTR(T, right)
    if T[currNode][0] == '*':
        return EvalAEBTR(T, left) * EvalAEBTR(T, right)
    if T[currNode][0] == '/':
        return EvalAEBTR(T, left) / EvalAEBTR(T, right)
```

a. The functions never produce the same output.
b. The functions always produce the same output.
c. The functions sometimes produce the same output, sometimes they don’t.
d. a and b

e. b and d.
18. When drawing the following tree using the method discussed in class, what are the coordinates of nodes A, B, C, and D?

a. A(4, 0), B(2, 3), C(6, 2), D(7, 3)
b. A(0, 0), B(1, 1), C(2, 2), D(3, 3)
c. A(3, 0), B(1, 3), C(5, 2), D(6, 3)
d. A(4, 0), B(3, 3), C(5, 2), D(6, 3)
e. None of the above
19. Consider the following Python function. What will \texttt{Funct(6)} return?

\begin{verbatim}
def Funct(n):
    if n == 2:
        return 2
    return n * Funct(n-1)
\end{verbatim}

a. 6
b. 2
c. 24
d. 120
e. *720

20. Consider the following Python function for fast searching in a sorted array. What will the printouts be when you type \texttt{BinSearchR([1, 3, 4, 9, 10, 20], 0, 5, 10)}?

\begin{verbatim}
def BinSearchR(A, l, r, a):
    print(l, r)
    if l == r:
        return A[l] == a
    m = (l+r)//2
    if A[m] >= a:
        return BinSearchR(A, l, m, a)
    else:
        return BinSearchR(A, m+1, r, a)
\end{verbatim}

\begin{table}
\begin{tabular}{|c|c|c|c|c|}
\hline
a. *  & b.  & c.  & d.  & e.  \\
0 5 & 0 5 & 0 5 & 0 5 & 0 5 \\
3 5 & 3 5 & 0 3 & 0 3 & 0 2 \\
3 4 & 4 4 & 1 3 & 2 3 & 0 1 \\
4 4 &  & 2 2 & 2 2 & 0 0 \\
\hline
\end{tabular}
\end{table}