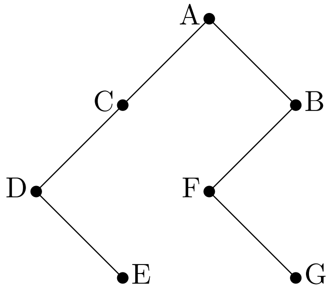
Math 211 Final
May 6, 2002

Professor L. Taylor

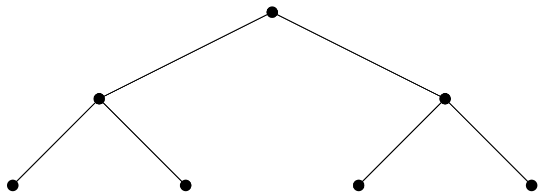
Name: ____

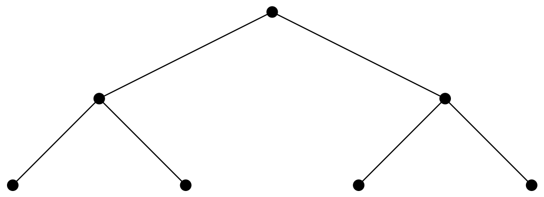
1. Consider the binary tree below whose nodes are labelled. On the line provided for your answer, write the node–labels in the order given by the tree order. The first letter you write should be the letter of the smallest node, followed by the next smallest and so on. The last letter on the line should be the largest node.

Answer ____



The binary tree has seven nodes as does the binary tree below whose nodes are unlabelled. Label the nodes in this tree so that the tree order for this tree is the same as your order on the answer line above. A second blank tree is provided for your scratch work so your final answer should be legible.





. . . .

Recall that the rule is “everything to the left, then me, then everything to the right”. Let’s work from the ‘A’. Since ‘C’, ‘D’ and ‘E’ are all to the left, they come before ‘A’. Since ‘D’ and ‘E’ are to the left of ‘C’, ‘C’ immediately preceeds ‘A’. Since ‘E’ is to the left of ‘D’, ‘E’ comes after ‘D’, so the first part of the sting is ‘D’, ‘E’, ‘C’, ‘A’.

Since ‘B’, ‘F’ and ‘G’ are all to the right of ‘A’, they all come after ‘A’. Both ‘F’ and ‘G’ are to the left of ‘B’ so they both come before ‘B’.Since ‘G’ is to the right of ‘F’, ‘G’ comes after ‘F’, so the whole string is

‘D’ ‘E’ ‘C’ ‘A’ ‘F’ ‘G’ ‘B’

For the second part, note there are three nodes to the left of the root and there are three letters that occur on the line before ‘A’. There are also three letters occurring after ‘A’ and three nodes to the right of the root, so ‘A’ goes at the root. Working left, we have to put ‘D’, ‘E’ and ‘C’ at the three nodes to the left of the root. The level-one node has a left node and a right node, so if we put ‘E’ at the level-one node and then ‘D’ at the left hand level-two node and then ‘C’ at the level-two right node we get the first part of the correctly ordered. The right hand branch is done similarly. The middle of the dtring ‘F’, ‘G’, ‘B’ goes at the level-one node and then ‘F’ goes to the left at level-two and ‘B’ goes to the right.

2. Consider the following short program

```
main(int argc, char *argv[]) {
char  c;
c=*((*(argv+2))+3);
putchar(c);
}
```

If you compile this program and run it as

```
a.out  The  quick  bown  fox  jumped
```

what character appears on the screen? What is the value of `argc`?

. . . .

First recall that `argc` is the number of words on the command line so in this example `argc`=6: the `a.out` counts. The value in `c` takes a bit of work to unravel. We work through the parentheses working form the inside out: `argv` is a pointer to an array of pointers, so `argv+2` points to the third pointer in the array. This is a pointer to the C-string `quick`. The `*` just before the opening parenthesis says take the value of that pointer. Then we add 3 to that, so it now points to the fourth character of `quick`. This is an 'c' so the `putchar` puts an 'c' on the screen.

If instead you ran

```
a.out  All  for  one  and  one  for  all
```

what is the value in `c` at the end of the program?

. . . .

This time, `*(argv+2)` points to the start of the C-string `for`; `(*(argv+2))+3` points to the fourth character in this C-string. There are only three characters there, but C-strings end with a 0, so the value in `c` is 0.

3. Consider the following two functions in C++. Assume they have been properly declared.

```
short rr(short v) {
return(v*v);
}

long rr(long v) {
return(2*v);
}
```

Discuss what happens when the following two lines execute. Be sure to include what value is returned by the call to `rr`.

```
short ss=5;
long LL=rr(ss);
```

. . . .

This is an example of C++ function overloading. The program determines which version of the function is used by the type of the variable passed to it. Specifically, `ss` is a short so `rr(ss)` uses the first version of the function and so `rr` returns 25. It returns a short which is to be stored in `LL`, which is a long, but this is an example of the sort of automatic promotion C and C++ handle for you. Hence 25 ends up in `LL`.

Discuss what happens when the following two lines execute. Be sure to include what value is returned by the call to `rr`.

```
long ss=5;
long LL=rr(ss);
```

. . . .

Now `ss` is a long, so the second version of the function is used and so a 10 gets stored in `LL`.

4. Finally, a multiple choice question! Given the declarations

```
short i, j, k;
```

consider the values in these three variables after the following statements execute:

```
i=k=7; j=2; j*=++k;j%=i--;i-=k; k/=8;
```

Which value is the value of `i`?
(a) -2 (b) -1 (c) 0 (d) 1 (e) 2

Which value is the value of `j`?
(a) -2 (b) -1 (c) 0 (d) 1 (e) 2

Which value is the value of `k`?
(a) 0 (b) 1 (c) 2 (d) 3 (e) 4

We work through the statements from left to right.

After the first statement

`i` is 7; `j` is undefined; and `k` is 7

After the scond statement

`i` is 7; `j` is 2; and `k` is 7

To evaluate the third statement we first increment `k` so it is now 8 and then we multiply the value of `j` by 8:

`i` is 7; `j` is 16; and `k` is 8

To evaluate the fourth statement we calculate the remainder of `j` when divided by `i`, so `j` is now 2 and then we decrement `i` so it is now 8 and then we multiply the value of `j` by 8:

`i` is 6; `j` is 2; and `k` is 8

To evaluate the fifth statement we subtract `k` from `i`

`i` is -2; `j` is 2; and `k` is 8

To evaluate the sixth statement we divide `k` by 8

`i` is -2; `j` is 2; and `k` is 1

Now we read off the answers: (a), (e) and (b).