

### Problem 37. (10 points):

Consider the following C declaration:

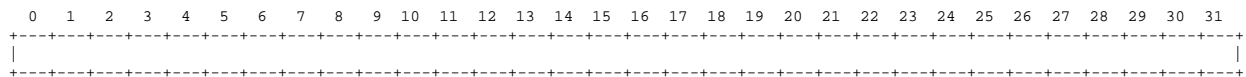
```
struct Node{
    char c;
    double value;
    struct Node* next;
    int flag;
    struct Node* left;
    struct Node* right;
};

typedef struct Node* pNode;

/* NodeTree is an array of N pointers to Node structs */
pNode NodeTree[N];
```

A. Using the template below (allowing a maximum of 32 bytes), indicate the allocation of data for a Node struct. Mark off and label the areas for each individual element (there are 6 of them). Cross hatch the parts that are allocated, but not used (to satisfy alignment).

Assume the Linux alignment rules discussed in Class 9. **Clearly indicate the right hand boundary of the data structure with a vertical line.**



B. For each of the four C references below, please indicate which assembly code section (labeled A – F) places the value of that C reference into register %eax. If no match is found, please write “NONE” next to the C reference.

The initial register-to-variable mapping for each assembly code section is:

```
%eax = starting address of the NodeTree array
%edx = i
```

-----  
C References:

1. \_\_\_\_\_ NodeTree[i]->flag
  2. \_\_\_\_\_ NodeTree[i]->left->left->c
  3. \_\_\_\_\_ NodeTree[i]->next->next->flag
  4. \_\_\_\_\_ NodeTree[i]->right->left->left
- 

Linux/IA32 Assembly:

|                                                                                                                                     |                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <p>A.</p> <pre>sall \$2, %edx leal (%eax,%edx),%eax movl 16(%eax),%eax</pre>                                                        | <p>B.</p> <pre>sall \$2,%edx leal (%eax,%edx),%eax movl (%eax),%eax movl 24(%eax),%eax movl 20(%eax),%eax movl 20(%eax),%eax</pre>  |
| <p>C:</p> <pre>sall \$2,%edx leal (%eax,%edx),%eax movl 20(%eax),%eax movl 20(%eax),%eax movsbl (%eax),%eax</pre>                   | <p>D:</p> <pre>sall \$2,%edx leal (%eax,%edx),%eax movl (%eax),%eax movl 16(%eax),%eax</pre>                                        |
| <p>E:</p> <pre>sall \$2, %edx leal (%eax,%edx),%eax movl (%eax),%eax movl 16(%eax),%eax movl 16(%eax),%eax movl 20(%eax),%eax</pre> | <p>F:</p> <pre>sall \$2, %edx leal (%eax,%edx),%eax movl (%eax),%eax movl 12(%eax),%eax movl 12(%eax),%eax movl 16(%eax),%eax</pre> |