```
stuct node
{ virtual string kind() = NULL;
  virtual int get_value() = NULL;
  virtual node * get_pointer(int n) = NULL;
  virtual string get_name() = NULL;
  virtual string get_operator() = NULL; };



struct  var_node: public node
{ string name;

  var_node(string s)
  { name=s; }

  virtual string kind()
  { return “variable”; }

  virtual string get_name()
  { return name; } };



struct  number_node: public node
{ int value;

  number_node(int u)
  { value=i; }

  virtual string kind()
  { return “integer”; }

  virtual int get_value()
  { return value; } };
```

```
struct node_with_ptrs: public node
{ int nptrs;
  node * * pointers;

  virtual string kind() = NULL;

  node_with_ptrs(node * p1 = NULL, node * p2 = NULL,
                 node * p3 = NULL, node * p4 = NULL)
  { nptrs = 0;
    if (p1!=NULL) nptrs=1;
    if (p2!=NULL) nptrs=2;
    if (p3!=NULL) nptrs=3;
    if (p4!=NULL) nptrs=4;
    pointers = new (node *) [nptrs];
    if (nptrs>0)
      pointers[0]=p1;
    if (nptrs>1)
      pointers[1]=p2;
    if (nptrs>2)
      pointers[2]=p3;
    if (nptrs>3)
      pointers[3]=p4; }

  virtual node * get_pointer(int i)
  { if (i<0 || i>=nptrs)
       return NULL;
    return pointers[i]; } };
```

```
struct arith_node: public node_with_pointers
{ string op;

  arith_node(node * a, string o, node * b):
                            node_with_pointers(a, b)
  { op=o; }

  virtual string kind()
  { return “arithmetic”; }

  virtual string get_operator()
  { return op; } };
```

```
struct if_node: public node_with_pointers
{ if_node(node * cond, node * truep, node * falsep = NULL):
         node_with_pointers(cond, truep, falsep)
  { }

  virtual string kind()
  { return “if”; } };
```