```c
struct stelem
{
 char sname[25];
 int  stype;
};
typedef struct stelem entry;


entry symtab[100];
int nsym;


void addtab( char *s)
{
 nsym++;
 strcpy( symtab[nsym].sname, s);
 symtab[nsym].stype = -1;
}

void showtab()
{
 int i;
 for (i = 1; i <= nsym; ++i)
   printf("%d: %s %d\n", i, symtab[i].sname, symtab[i].stype);
}

int intab( char *s)
{
 int i;
 for ( i = 1; i <= nsym; ++i)
 {
   if ( strcmp(symtab[i].sname, s) == 0)
    return 1;
 }
 return 0;

}
```

```
int addtype( char *s, int t)
{
 int i, loc = -1;
 for ( i = 1; i <= nsym; ++i)
 {
   if ( strcmp(symtab[i].sname, s) == 0)
    loc = i;
 }
 if (loc > 0)
  {
   //printf("Set type %s to %d\n", s, t);
   symtab[loc].stype = t;
  }
 else
 {
   //printf("Unable to set type %s to %d\n", s, t);
 }
}


int gettype( char *s)
{
 int t = -1;
 int i, loc = -1;
 for ( i = 1; i <= nsym; ++i)
 {
   if ( strcmp(symtab[i].sname, s) == 0)
    loc = i;
 }
 if (loc > 0)
  {
   t = symtab[loc].stype;
   //printf("Get type for %s to %d\n", s, t);
  }
 if (loc <= 0)
        ;
   //printf("gettype var %s not found\n", s);
 else if (t < 0)
        ;
   //printf("gettype var %s has bad type %d\n", s, t);
 else
        ;
   //printf("gettype var %s has type %d\n", s, t);

 return t;
}
```