22C:050 HW#3 Solutions

1.

```
char pool[poollim]      /* the string pool */

int pos = 0;      /* current position in string pool */

int store(symtype symbol)
/* add symbol to stringpool */
{
            int len, i;
            len = length(symbol);

            /* start */
            if(pos + len +1 > poollim)return -1;
            /* end */

            for(i = 0; i < len; i++) pool[pos+i] = symbol[i];
            pool[pos + len] = marker;
            i=pos;
            pos = pos + len +1;
            return i;
} /* store */
```

2.
(Memory overflow is a critical problem in system implementation. For instance in C programming language, we need to pay attention to pointer allocation/deallocation, and array index problems.)

There is a piece of code within function POOL_HANDLE pool_add(char * s, int l) in file stringpool.c:

```
            /* l, the length of the string to be added  */
        int val = pos;
        if ((pos + l + 1) >= POOLSIZE ) {
                /* throw a pool_full exception */
                EXCEPT_RAISE( pool_full );
        }
```

which will check the overflow problem before adding a string.

We do the same in handling table entry problem in function SYM_HANDLE sym_find(char * s, int l, SYM_HANDLE h) in file symboltable.c.

3.

There are at least two places where hash function is used.
 1) A hash function

```
SYM_HANDLE sym_hash( char c, SYM_HANDLE h )
{
return (((SYM_HANDLE)h + (unsigned long int)c)*7) % SYMSIZE;
}
```

is used to computer the handle of a string.

 2) A hash function is also used in function SYM_HANDLE sym_find(char * s, int l, SYM_HANDLE h) to find a table entry for insertion or matching.

4.

EAL V0.0 by Douglas W. Jones; Sun Sep 21 19:46:28 2003

```
    1              |
    2 0000: 0002  |        A: W C  ; two passes for C value
    3 0002: ????  |       C: W D  ; we need chaining for D value
                             ^
undefined
    4              |       D = F   ;
    5              |       F = 5
    6              |
```

5.

a) The first design encapsulate all the details of chaining implementation in the symbol table package and leave users the interface to handle the table. This
Design follows the abstract data byte or object oriented design philosophies
to a large degree.

b) The second design only provides some basic data structure and operations,
leaving and exposing much space for users. Hence users have a lot of flexibilities in realizing what they want, including those actions mentioned in the question.