

CS 2103

Assignment in Data Structures # 4

(**display()** and **insertLast()** for Linked List and Cursor-Based Implementation)

Submitted by:

**Robert Michael Lim, BSCS – 2**

Submitted to:

**Ms. Christine F. Peña**

**Linked List Implementation**

With the structure definition of:

typedef struct node{

char elem;

struct node \*next;

}ctype, \*List;

**display();**

void display(List L)

{

for(;L!=NULL;L=L->next){

printf("%c ", L->elem);

}

}

**insertLast();**

void insertLast(List \*L, char x)

{

List \*trav, temp;

for(trav=L;\*trav!=NULL;trav=&(\*trav)->next){}

temp = (List)malloc(sizeof(struct node));

if(temp!=NULL){

temp->elem = x;

temp->next = \*trav;

\*trav=temp;

}

}

**Cursor-Based Implementation**

With the structure definitions of:

typedef struct{

char FN[24], LN[16], MI;

}nametype;

typedef struct{

nametype elem;

int next;

}nodetype;

typedef struct{

nodetype NODES[SIZE];

int avail;

}VirtualHeap;

**display();**

void display(VirtualHeap VH, List L)

{

int x = L;

if(x==-1){

printf("List is full\n");

}

for(;x!=-1;x = VH.NODES[x].next){

printf("%s %s %c \n", VH.NODES[x].elem.FN, VH.NODES[x].elem.LN, VH.NODES[x].elem.MI);

}

}

**insertLast();**

void insertLast(VirtualHeap \*VH, List \*L, char fName[], char lName[], char middleInitial)

{

int temp, \*p;

if(VH->avail!=-1)

{

for(p = L; \*p!=-1 ;p = &VH->NODES[\*p].next){}

temp = VH->avail;

if(temp!=-1){

strcpy(VH->NODES[temp].elem.FN, fName);

strcpy(VH->NODES[temp].elem.LN, lName);

VH->NODES[temp].elem.MI = middleInitial;

VH->avail = VH->NODES[VH->avail].next;

VH->NODES[temp].next=\*p;

\*p=temp;

}

}

}