LAB REPORT 实验报告

Lab Title	Linked List				Lab No.	12
Stud. Name		Major	CST		Class	
Student ID			Date			

Lab description/objectives:

1. Watch the following video: Print elements of a linked list in forward and reverse order using recursion

2. Add two functions printForward(), printReverse() to Program 13.7 that print elements of the linked list in forward, and reverse order using recursion.

Source code:

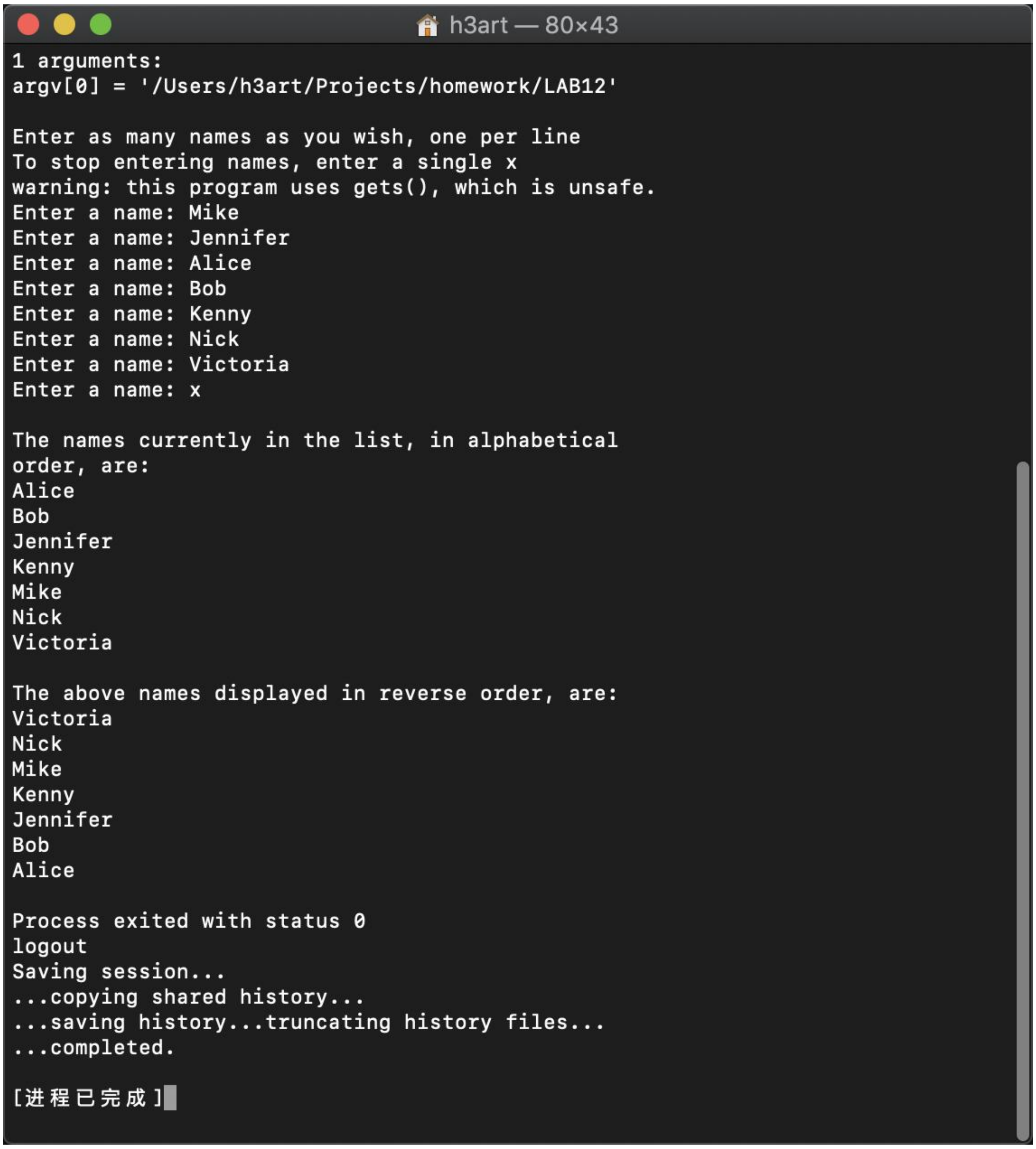
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXCHARS 30
#define DEBUG 0

/* here is the declaration of a linked list structure */
struct NameRec {
	char name[MAXCHARS];
	struct NameRec* nextAddr;
};

/* here is the definition of the first structure pointer */
struct NameRec* firstRec;

int main()
{
	void readInsert(); /* function prototypes */
	void printForward(struct NameRec*);
	void printReverse(struct NameRec*);
```

```
    firstRec = NULL; /* initialize list pointer */
    readInsert();
    printf("\nThe names currently in the list, in alphabetical");
    printf("\norder, are:\n");
    printForward(firstRec);
    printf("\nThe above names displayed in reverse order, are:\n");
    printReverse(firstRec);
    printf("\n");

    return 0;
}

/* get a name and insert it into the linked list */
void readInsert()
{
    char name[MAXCHARS];
    void insert(char*);

    printf("\nEnter as many names as you wish, one per line");
    printf("\nTo stop entering names, enter a single x\n");
    while (1) {
        printf("Enter a name: ");
        gets(name);
        if (strcmp(name, "x") == 0)
            break;
        insert(name);
    }
}

void insert(char* name)
{
```

```
struct NameRec* linearLocate(char*); /* function prototype */
struct NameRec *newaddr, *here; /* pointers to structure */
/* of type NameRec */

newaddr = (struct NameRec*)malloc(sizeof(struct NameRec));
if (newaddr == (struct NameRec*)NULL) /* check the address */
{
    printf("\nCould not allocate the requested space\n");
    exit(1);
}

/* locate where the new structure should be placed and */
/* update all pointer members */
if (firstRec == NULL) /* no list currently exists */
{
    newaddr->nextAddr = NULL;
    firstRec = newaddr;
}
else if (strcmp(name, firstRec->name) < 0) /* a new first structure */
{
    newaddr->nextAddr = firstRec;
    firstRec = newaddr;
}
else /* structure is not the first structure of the list */
{
    here = linearLocate(name);
    newaddr->nextAddr = here->nextAddr;
    here->nextAddr = newaddr;
}

strcpy(newaddr->name, name); /* store the name */
```

```
}


/* This function locates the address of where a new structure
   should be inserted within an existing list.
   It receives the address of a name and returns the address of a
   structure of type NameRec
*/
struct NameRec* linearLocate(char* name)
{
    struct NameRec *one, *two;
    one = firstRec;
    two = one->nextAddr;

    if (two == NULL)
        return (one);
    /* new structure goes after the existing single structure */
    while (1) {
        if (strcmp(name, two->name) < 0) /* if it is located within the list */
            break;
        else if (two->nextAddr == NULL) /* it goes after the last structure */
        {
            one = two;
            break;
        }
        else /* more structures to search against */
        {
            one = two;
            two = one->nextAddr;
        }
    } /* the break takes us here */
    return (one);
}

/* display names from the linked list(Forward) */
void printForward(struct NameRec* node)
{
    if (node == NULL) {return;}
    printf("%s\n", node->name);
    printForward(node->nextAddr);
}

/* display names from the linked list(Reverse) */
void printReverse(struct NameRec* node)
{
    if (node == NULL) {return;}
    printReverse(node->nextAddr);
    printf("%s\n", node->name);
}
```

Program outputs:



```
1 arguments:
argv[0] = '/Users/h3art/Projects/homework/LAB12'

Enter as many names as you wish, one per line
To stop entering names, enter a single x
warning: this program uses gets(), which is unsafe.
Enter a name: Mike
Enter a name: Jennifer
Enter a name: Alice
Enter a name: Bob
Enter a name: Kenny
Enter a name: Nick
Enter a name: Victoria
Enter a name: x

The names currently in the list, in alphabetical
order, are:
Alice
Bob
Jennifer
Kenny
Mike
Nick
Victoria

The above names displayed in reverse order, are:
Victoria
Nick
Mike
Kenny
Jennifer
Bob
Alice

Process exited with status 0
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

Discussion:

1. **Most difficult parts**

2. **Bugs and/or Errors**