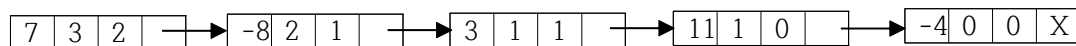


Review Questions

1. Linked lists are useful when you are not confident about the number of elements. Since the size of the array is determined and used, an error may occur if the maximum size is exceeded. But the connection list can be continuously added and used when needed, so there is relatively no limitation on capacity.
2. Since the double linked list has pointer about the forward and backward nodes, it is possible to search twice as efficiently.
3. We can begin at any node and traverse the list in any direction, forward or backward, until we reach the same node where we started. It is used to revisit a webpage you have visited, or to return to listen to a song you have heard from a music playlist.
4. Normally when we access the first node, we assign the first node to the starting point variable. However, if the header link list is used, the address of the next node in the starting node becomes the first node.

5.



6. The last node in a single linked list does not have a pointer to the next node and ends with a null value, but the circular linked list contains the pointer to the first node.
7. struct student

```
{  
    int roll_no;  
    char name[20];  
    char grade;  
    struct student *next;  
}
```

8.

```
#include <stdio.h>  
#include <malloc.h>  
#include <string.h>  
struct node  
{  
    int std_no;  
    char name[10];  
    struct node *next;  
};  
  
struct node *start1=NULL;  
struct node *create(struct node *start);  
struct node *display(struct node *start);
```

```

int main()
{
    int count;
    start1=create(start1);
    printf("\n\n");
    display(start1);
}

struct node *create(struct node *start)
{
    struct node *new_node, *ptr;
    int std;
    char name[10];
    int fee;
    printf("Enter -1 to end\n");
    printf("Enter the student_number : ");
    scanf("%d",&std);
    printf("Enter the name : ");
    scanf("%s",name);
    while(std!=-1)
    {
        new_node=(struct node*)malloc(sizeof(struct node));
        new_node->std_no=std;
        strcpy(new_node->name,name);

        if(start==NULL)
        {
            new_node->next=new_node;
            start=new_node;
        }
        else
        {
            ptr=start;
            while(ptr->next!=start)
                ptr=ptr->next;
            ptr->next=new_node;
            new_node->next=start;
        }
        printf("Enter -1 to end\n");
        printf("Enter the roll_number : ");
        scanf("%d",&std);
        if(std===-1)

```

```

        break;
        printf("Enter the name : ");
        scanf("%s",name);
    }
    return start;
}

struct node *display(struct node *start)
{
    struct node *ptr;
    ptr=start;
    while(ptr->next!=start)
    {
        printf("student_number = %d\n",ptr->std_no);
        printf("name = %s\n",ptr->name);
        printf("\n");
        ptr=ptr->next;
    }

    printf("roll_number = %d\n",ptr->std_no);
    printf("name = %s\n",ptr->name);

    return start;
}

```

```

C:\WINDOWS\system32\cmd.exe
Enter -1 to end
Enter the student_number : 1
Enter the name : amy
Enter -1 to end
Enter the roll_number : 2
Enter the name : billy
Enter -1 to end
Enter the roll_number : -1

student_number = 1
name = amy

roll_number = 2
name = billy
계속하려면 아무 키나 누르십시오 . . .

```

9.

```
#include <stdio.h>
#include <malloc.h>
#include <string.h>
struct student *create(struct student *);
struct student *display(struct student *);
struct student *delete_beg(struct student *);
struct student *delete_node(struct student *);
struct student
{
    int roll_no;
    char name[20];
    char grade;
    struct student *next;
};
struct student *start= NULL;
int main()
{
    start=create(start);
    start=display(start);
    start=delete_node(start);
    start=display(start);
}
struct student *delete_beg(struct student *start)
{
    struct student *ptr;
    ptr=start;
    start=start->next;
    free(ptr);
    return start;
}
struct student *delete_node(struct student *start)
{
    struct student *ptr, *preptr;
    int val;
    printf("Enter the value of the node which has to be deleted");
    scanf("%d",&val);
    ptr=start;
    if(ptr->roll_no==val)
    {
        start=delete_beg(start);
        return start;
    }
    else
```

```

        {
            while(ptr->roll_no!=val)
            {
                preptr=ptr;
                ptr=ptr->next;
            }
            preptr->next=ptr->next;
            free(ptr);
            return start;
        }
    }

    struct student *display(struct student *start)
    {
        struct student *ptr;
        ptr=start;
        while(ptr!=NULL)
        {
            printf("roll_no : %d\n",ptr->roll_no);
            printf("name : %s\n",ptr->name);
            printf("grade : %c\n",ptr->grade);
            ptr=ptr->next;
        }
        return start;
    }

    struct student *create(struct student *start)
    {
        struct student *new_node, *ptr;
        char grade,name[20];
        int roll;
        int i;

        for(i=0;i<5;i++)
        {
            printf("Enter the roll_num : ");
            scanf("%d",&roll);
            if(roll== -1)
                break;
            fflush(stdin);
            printf("Enter the name :");
            gets(name);
            printf("Enter the grade : ");

```

```

        scanf("%c",&grade);
new_node=(struct student *)malloc(sizeof(struct student));
new_node->roll_no=roll;
strcpy(new_node->name,name);
new_node->grade=grade;
    if(start==NULL)
    {
        new_node->next=NULL;
        start=new_node;
    }
    else
    {
        ptr=start;
        while(ptr->next!=NULL)ptr=ptr->next;
        ptr->next=new_node;
        new_node->next=NULL;
    }
}
return start;
}

```

```

C:\WINDOWS\system32\cmd.exe
Enter the roll_num : 1
Enter the name :amy
Enter the grade : c
Enter the roll_num : 2
Enter the name :billy
Enter the grade : a
Enter the roll_num : -1
roll_no : 1
name : amy
grade : c
roll_no : 2
name : billy
grade : a
Enter the value of the node which has to be deleted2
roll_no : 1
name : amy
grade : c
계속하려면 아무 키나 누르십시오 . . .

```

10.

```
#include <stdio.h>
#include <malloc.h>
struct alpha{
    char data;
    struct alpha *next;
};

struct alpha *start = NULL;
struct alpha *create_l(struct alpha *);
struct alpha *create_u(struct alpha *);

int main()
{
    start=create_l(start);
    start=create_u(start);
}

struct alpha *create_l(struct alpha *start)
{
    struct alpha *new_node, *ptr;
    char alpha;
    printf("Enter -1 to end\n");
    printf("Enter the data(lower case) : ");
    scanf("%c",&alpha);
    while(alpha>='a' && alpha<='z')
    {
        new_node=(struct alpha*)malloc(sizeof(struct alpha));
        new_node->data;
        if(start==NULL)
        {
            new_node->next=NULL;
            start=new_node;
        }
        else
        {
            ptr=start;
            while(ptr->next!=NULL)
                ptr=ptr->next;
            ptr->next=new_node;
            new_node->next=NULL;
        }
    }
}
```

```

        fflush(stdin);
        printf("Enter the data : ");
        scanf("%c",&alpha);
    }
    return start;
}

struct alpha *create_u(struct alpha *start)
{
    struct alpha *new_node, *ptr;
    char alpha;
    fflush(stdin);
    printf("Enter -1 to end\n");
    printf("Enter the data(upper case) : ");
    scanf("%c",&alpha);
    fflush(stdin);
    while(alpha>='A' && alpha<='Z')
    {
        new_node=(struct alpha*)malloc(sizeof(struct alpha));
        new_node->data;
        if(start==NULL)
        {
            new_node->next=NULL;
            start=new_node;
        }
        else
        {
            ptr=start;
            while(ptr->next!=NULL)
                ptr=ptr->next;
            ptr->next=new_node;
            new_node->next=NULL;
        }
        fflush(stdin);
        printf("Enter the data : ");
        scanf("%c",&alpha);
    }
    return start;
}

```



```
C:\WINDOWS\system32\cmd.exe
Enter -1 to end
Enter the data(lower case) : a
Enter the data : b
Enter the data : c
Enter the data : d
Enter the data : e
Enter the data : f
Enter the data : -1
Enter -1 to end
Enter the data(upper case) : A
Enter the data : B
Enter the data : C
Enter the data : D
Enter the data : F
Enter the data : -1
계속하려면 아무 키나 누르십시오 . . .
```

11.

```
#include <stdio.h>
#include <malloc.h>
#include <string.h>
struct node
{
    char name[10];
    struct node *next;
};
struct node *start=NULL;
struct node *create(struct node *start);
struct node *display(struct node *start);
struct node *sort(struct node *start);
int main()
{
    start=create(start);
    start=sort(start);
    start=display(start);
}
struct node *create(struct node *start)
{
    struct node *new_node, *ptr;
    char name[10];
```

```

int num;
printf("Enter -1 to end\n");
printf("did you want to enter the name (yes=1) : ");
scanf("%d",&num);
printf("Enter the name : ");
scanf("%s",name);
while(num!=-1)
{
    new_node=(struct node*)malloc(sizeof(struct node));
    strcpy(new_node->name,name);
    if(start==NULL)
    {
        new_node->next=NULL;
        start=new_node;
    }
    else
    {
        ptr=start;
        while(ptr->next!=NULL)
            ptr=ptr->next;
        ptr->next=new_node;
        new_node->next=NULL;
    }
    printf("did you want to enter the name (yes=1) : ");
    scanf("%d",&num);
    if(num===-1)
        break;
    printf("Enter the name : ");
    scanf("%s",name);
}
return start;
}

```

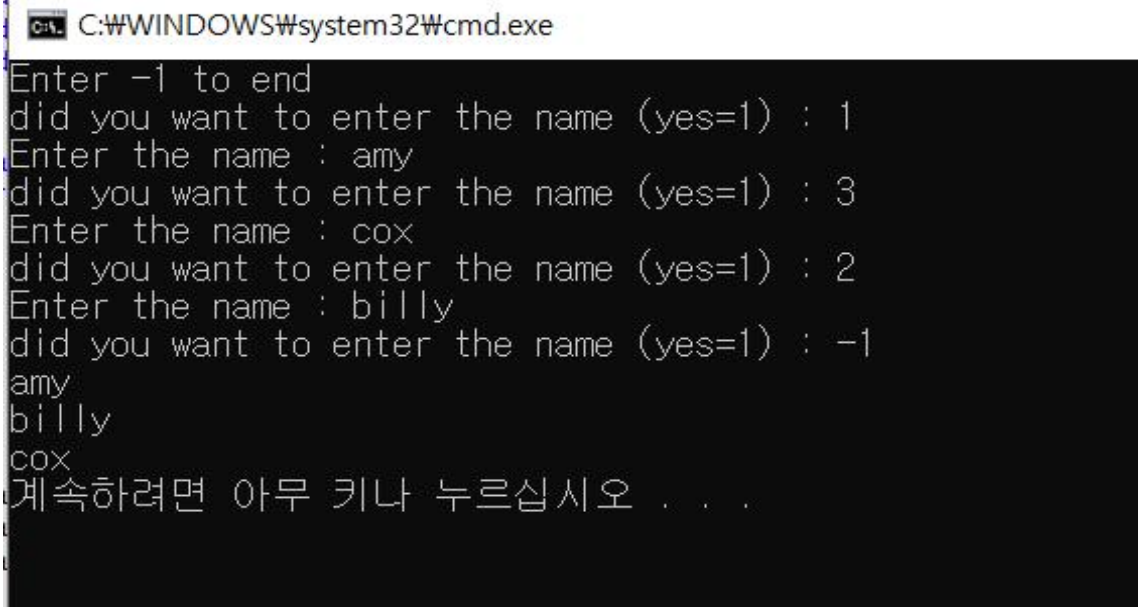
```

struct node *display(struct node *start)
{
    struct node *ptr;
    ptr=start;
    while(ptr!=NULL)
    {
        printf("%s\n",ptr->name);
        ptr=ptr->next;
    }
    return start;
}

```

```
}
```

```
struct node *sort(struct node *start)
{
    struct node *ptr1, *ptr2;
    char temp[10];
    ptr1=start;
    while(ptr1->next!=NULL)
    {
        ptr2=ptr1->next;
        while(ptr2!=NULL)
        {
            if(strcmp(ptr1->name,ptr2->name)==1)
            {
                strcpy(temp,ptr1->name);
                strcpy(ptr1->name,ptr2->name);
                strcpy(ptr2->name,temp);
            }
            ptr2=ptr2->next;
        }
        ptr1=ptr1->next;
    }
    return start;
}
```



```
C:\WINDOWS\system32\cmd.exe
Enter -1 to end
did you want to enter the name (yes=1) : 1
Enter the name : amy
did you want to enter the name (yes=1) : 3
Enter the name : cox
did you want to enter the name (yes=1) : 2
Enter the name : billy
did you want to enter the name (yes=1) : -1
amy
billy
cox
계속하려면 아무 키나 누르십시오 . . .
```

Multiple-choice Questions

1. (b)
2. (a)
3. (d)
4. (c)
5. (b)
6. (d)
7. (b)

True or False

1. T
2. T
3. F
4. F
5. F
6. F
7. T
8. F
9. T
10. T

Fill in the Blanks

1. A pointer variable AVAIL
2. $O(1)$
3. $O(1)$
4. two
5. two
6. two
7. four
8. one
9. two
10. one
11. node
12. head
13. storing the information part
14. no free memory cell is present in the system
15. first