1)Problem 1

Makefile.mk

```
a.exe: client.o server.o
	gcc -o a.exe client.o server.o
client.o : client.c server.h
	gcc -c client.c
server.o : server.c server.h
	gcc -c server.c

```

Server.c

```c
#include<stdio.h>
#include<stdlib.h>
#include "server.h"
int sumall(struct node *head)
{
    int sum=0;
    struct node *temp;
    temp=head;
    while(temp!=NULL)
    {
        sum=sum+(temp->data);
        temp=temp->next;
    }
    return sum;
}
int sumeven(struct node *head)
{
    int sum=0;
    struct node *temp;
    temp=head;
    while(temp!=NULL)
    {
        if((temp->data)%2==0)
        sum=sum+(temp->data);
        temp=temp->next;
    }
    return sum;
}
```

```c
int sumodd(struct node *head)
{
    int sum=0;
    struct node *temp=head;
    while(temp!=NULL)
    {
        if((temp->data)%2!=0)
        sum=sum+(temp->data);
        temp=temp->next;
    }
    return sum;
}
int sumalternate(struct node *head)
{
    int sum=0,count=0;
    struct node *temp=head;
    while(temp!=NULL)
    {
        if(count%2==0)
        {
        sum=sum+(temp->data);
        }
        count++;
        temp=temp->next;
    }
    return sum;
}
```

```c
void display(struct node *head)
{
    struct node *temp=head;
    int d,i=1;
    while(temp!=NULL)
    {
        d=(temp->data);
        temp=temp->next;
        printf("node-%d:%d\n",i,d);
        i++;
    }
    if(i==1 && temp==NULL)
    {
        printf("Linked list is empty!\n");
    }
}
struct node* beg(struct node *head)
{
    struct node *p;
    int value;
    printf("Enter the data to be inserted:");
    scanf("%d",&value);
    p=malloc(sizeof(struct node));
    p->data=value;
    p->next=head;
    printf("Insertion successful!\n");
    return (p);
}
```

```c
struct node* end(struct node *head)
{
    int new_data;
    printf("Enter data to be inserted:");
    scanf("%d",&new_data);
    struct node* new_node = (struct node*) malloc(sizeof(struct node));

    struct node *last = head;

    new_node->data  = new_data;

    new_node->next = NULL;

    if (head == NULL)
    {
       head = new_node;
       return(head);
    }

    while (last->next != NULL)
        last = last->next;
       last->next = new_node;
    printf("Insertion successful!\n");
    return head;
}
```

Client.c

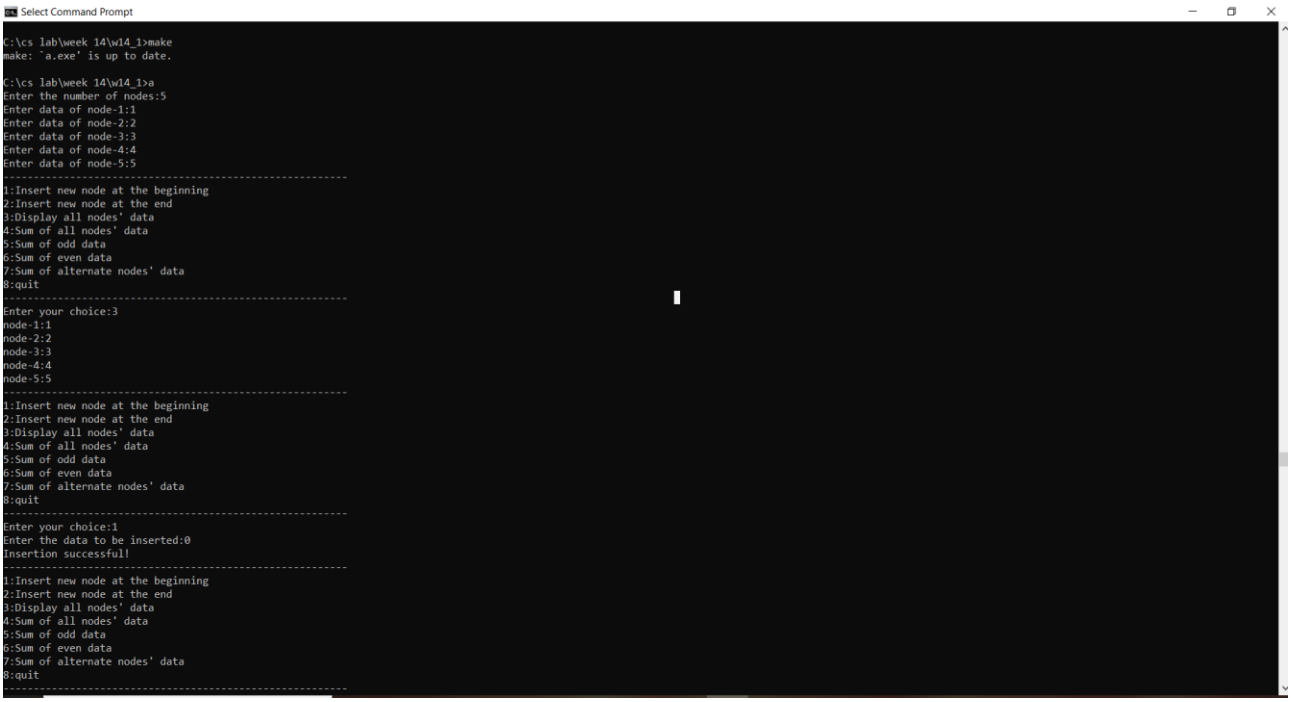
```c
#include<stdio.h>
#include<stdlib.h>
#include "server.h"
int main()
{
    int n;
    struct node *head=NULL,*newnode,*temp;
    printf("Enter the number of nodes:");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        newnode=(struct node *)malloc(sizeof(struct node));
        printf("Enter data of node-%d:",i+1);
        scanf("%d",&newnode->data);
        newnode->next=NULL;
        if(head==NULL)
        {
            head=newnode;
            temp=newnode;

        }
        else
        {
            (temp->next)=newnode;
            temp=newnode;
        }

    }
    int choice,sum=0,status=1;
    while(status)
    {
    printf("------------------------------------------------------------\n1:Insert new node at the beginning\n2:Insert new node at the end\n3:Display all nodes' data\n
    4:Sum of all nodes' data\n5:Sum of odd data\n6:Sum of even data\n7:Sum of alternate nodes' data\n8:quit
    \n------------------------------------------------------------\n");
    printf("Enter your choice:");
    scanf("%d",&choice);
```

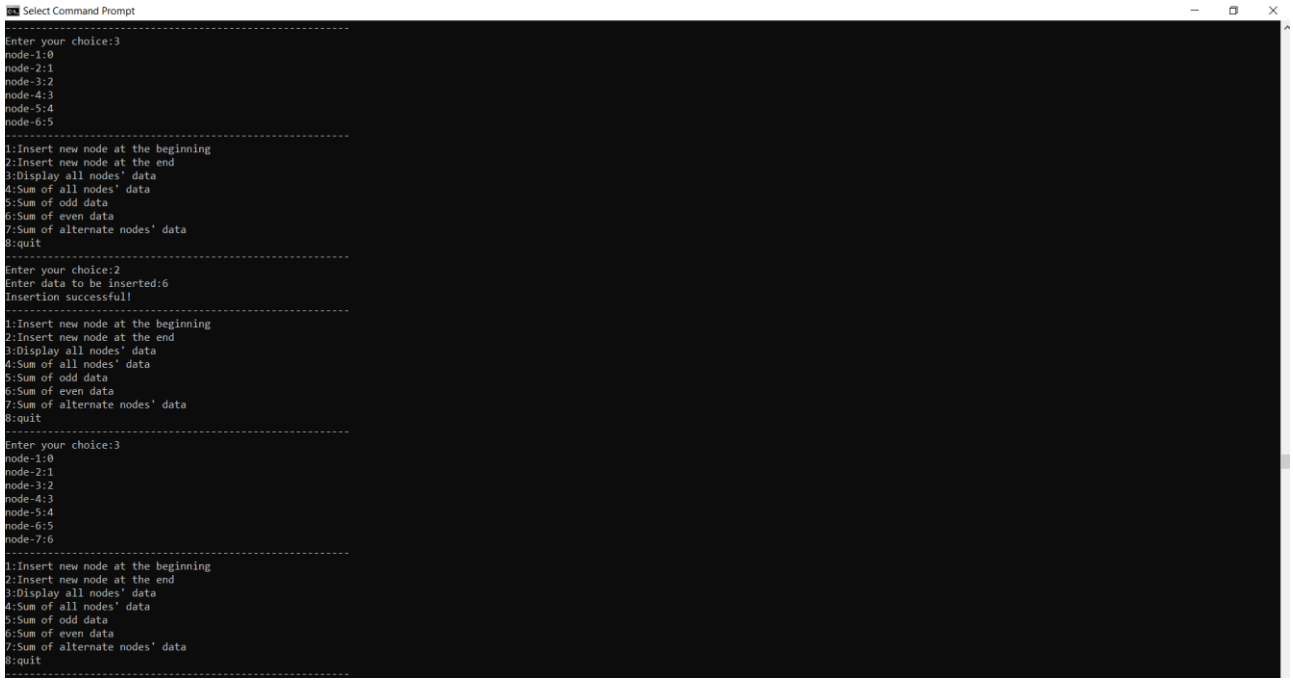
```c
    switch(choice)
    {
        case 1:head=beg(head);
        break;
        case 2:head=end(head);
        break;
        case 3:display(head);
        break;
        case 4:sum=sumall(head);
        printf("Sum of all nodes' data=%d\n",sum);
        sum=0;
        break;
        case 5:sum=sumodd(head);
        printf("Sum of odd data=%d\n",sum);
        sum=0;
        break;
        case 6:sum=sumeven(head);
        printf("Sum of even data=%d\n",sum);
        sum=0;
        break;
        case 7:sum=sumalternate(head);
        printf("Sum of alternate nodes' data=%d\n",sum);
        sum=0;
        break;
        case 8:status=0;
        break;
        default:printf("Enter a choice between 1 and 8 only\nTry again!\n");

    }
    }
    free(head);
    free(temp);
    free(newnode);
    return 0;
}
```

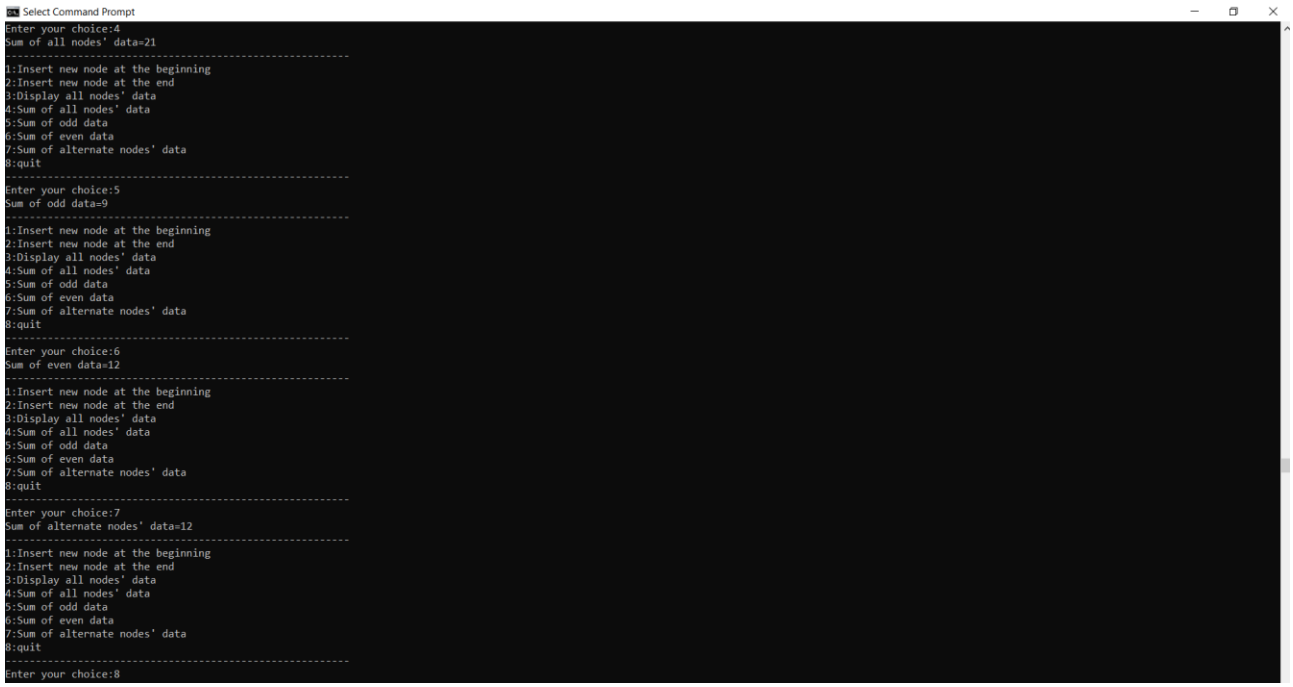
Server.h

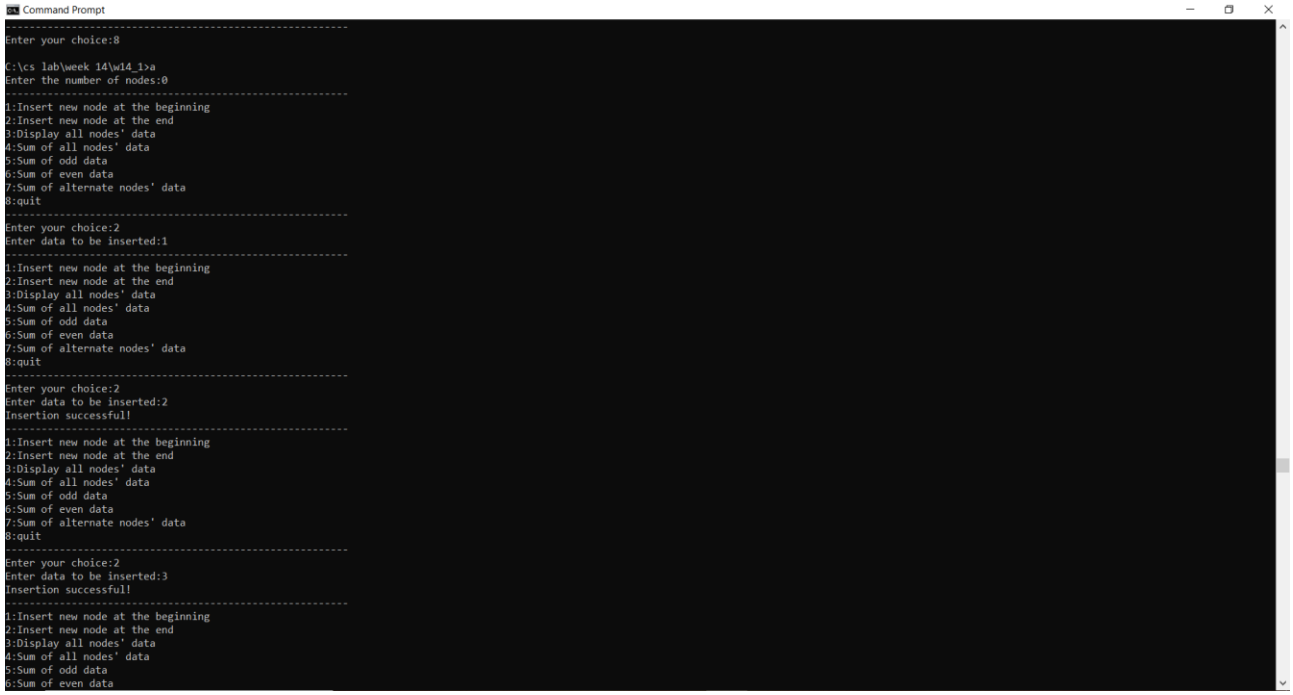
```c
struct node
{
    int data;
    struct Node *next;
};
int sumall(struct node *);
int sumalternate(struct node *);
int sumodd(struct node *);
int sumeven(struct node *);
void display(struct node *);
struct node* beg(struct node *);
struct node* end(struct node *);

```

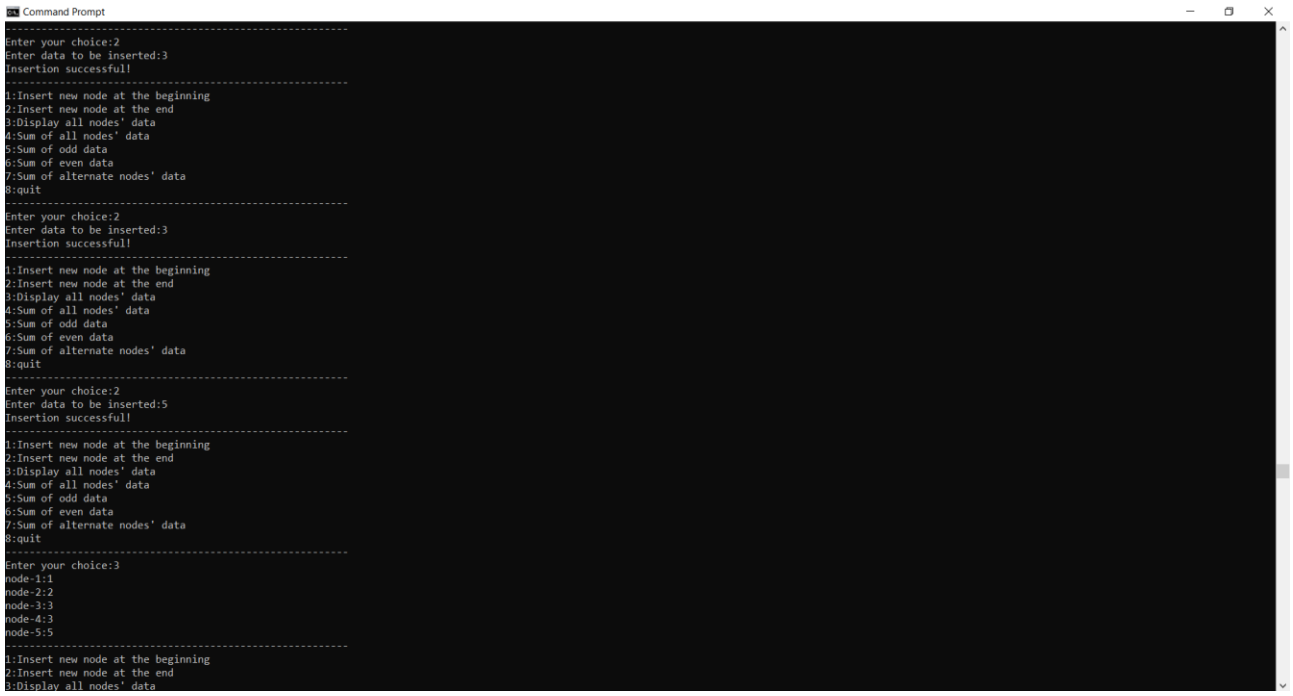
Output

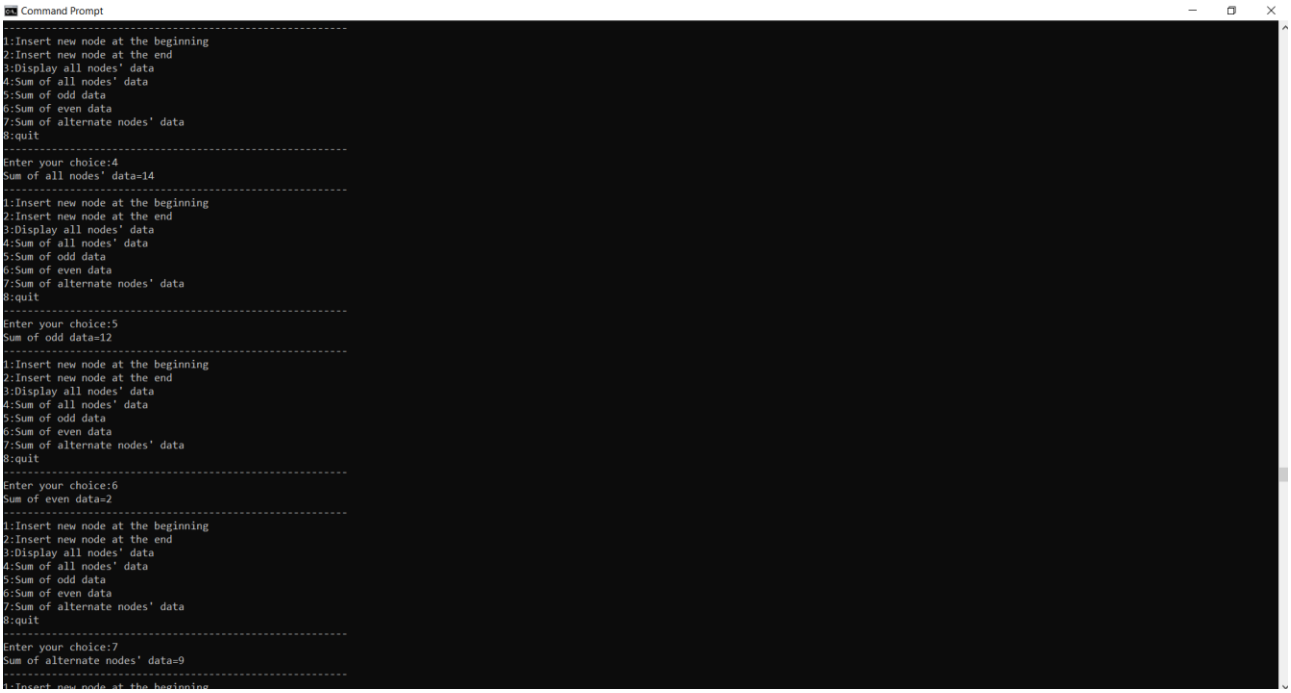

Select Command Prompt
C:\cs lab\week 14\w14_1>make
make: `a.exe' is up to date.
C:\cs lab\week 14\w14_1>a
Enter the number of nodes:5
Enter data of node-1:1
Enter data of node-2:2
Enter data of node-3:3
Enter data of node-4:4
Enter data of node-5:5
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit
Enter your choice:3
node-1:1
node-2:2
node-3:3
node-4:4
node-5:5
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit
Enter your choice:1
Enter the data to be inserted:0
Insertion successful!
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit

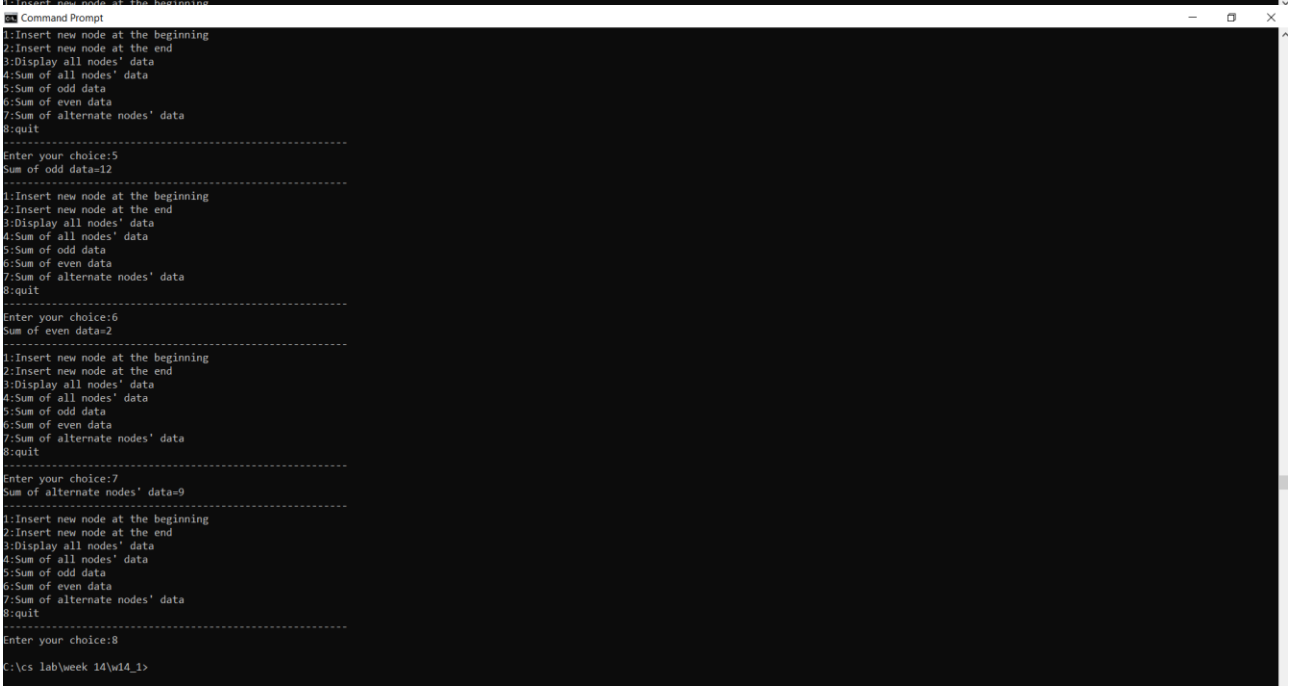

Select Command Prompt
Enter your choice:3
node-1:0
node-2:1
node-3:2
node-4:3
node-5:4
node-6:5
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit
Enter your choice:2
Enter data to be inserted:6
Insertion successful!
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit
Enter your choice:3
node-1:0
node-2:1
node-3:2
node-4:3
node-5:4
node-6:5
node-7:6
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit


Select Command Prompt
Enter your choice:4
Sum of all nodes' data=21
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit
Enter your choice:5
Sum of odd data=9
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit
Enter your choice:6
Sum of even data=12
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit
Enter your choice:7
Sum of alternate nodes' data=12
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit
Enter your choice:8


Command Prompt
Enter your choice:8
C:\cs lab\week 14\w14_1>a
Enter the number of nodes:0
1:Insert new node at the beginning
2:Insert new node at the end
3:Display all nodes' data
4:Sum of all nodes' data
5:Sum of odd data
6:Sum of even data
7:Sum of alternate nodes' data
8:quit
Enter your choice:2
Enter data to be inserted:1
Enter your choice:2
Enter data to be inserted:2
Insertion successful!
Enter your choice:2
Enter data to be inserted:3
Insertion successful!


Command Prompt
Enter your choice:2
Enter data to be inserted:3
Insertion successful!
Enter your choice:2
Enter data to be inserted:3
Insertion successful!
Enter your choice:2
Enter data to be inserted:5
Insertion successful!
Enter your choice:3
node-1:1
node-2:2
node-3:3
node-4:3
node-5:5





2)Practice problem 1

Makefile.mk

```
a.exe: client.o server.o
    gcc -o a.exe client.o server.o
client.o : client.c server.h
    gcc -c client.c
server.o : server.c server.h
    gcc -c server.c

```

Server.c

```c
#include<stdio.h>
#include<stdlib.h>
#include "server.h"
int productall(struct node *head)
{
    int p=1;
    struct node *temp;
    temp=head;
    while(temp!=NULL)
    {
        p=p*(temp->data);
        temp=temp->next;
    }
    return p;
}
```

```c
void search(struct node *head)
{
    int status=-1,node=1,ele;
    printf("Enter the search element:");
    scanf("%d",&ele);
    struct node *temp;
    temp=head;
    while(temp!=NULL)
    {
        if((temp->data)==ele)
        {
            status=node;
            break;
        }
        temp=temp->next;
        node++;
    }
    if(status==-1)
    {
        printf("Element not found!\n");
    }
    else
    {
        printf("%d found in node-%d from beginning\n",ele,node);
    }

}
void display(struct node *head)
{
    struct node *temp=head;
    int d,i=1;
    while(temp!=NULL)
    {
        d=(temp->data);
        temp=temp->next;
        printf("node-%d:%d\n",i,d);
        i++;
    }
    if(i==1 && temp==NULL)
    {
        printf("Linked list is empty!\n");
    }
}
struct node* beg(struct node *head)
{
    if (head == NULL)
        return NULL;

    struct node* temp = head;
    head = head->next;

    free(temp);
    printf("Succesfully Deleted!\n");
    return head;
}
struct node* end(struct node *p_head)
{
    struct node *p_cur, *p_prev;

    if (p_head != NULL)
    {
        if (p_head->next == NULL)
        {
            free (p_head);
            p_head = NULL;
        }
        else
        {
            for (p_cur = p_head; p_cur->next != NULL; p_prev=p_cur,p_cur = p_cur->next)
                ;
            p_prev -> next = NULL;
            free (p_cur);
        }
    }
    printf("Succesfully Deleted!\n");
    return (p_head);
}
```

Client.c

```c
#include<stdio.h>
#include<stdlib.h>
#include "server.h"
int main()
{
    int n;
    struct node *head=NULL,*newnode,*temp;
    printf("Enter the number of nodes:");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        newnode=(struct node *)malloc(sizeof(struct node));
        printf("Enter data of node-%d:",i+1);
        scanf("%d",&newnode->data);
        newnode->next=NULL;
        if(head==NULL)
        {
            head=newnode;
            temp=newnode;

        }
        else
        {
            (temp->next)=newnode;
            temp=newnode;
        }

    }
    int choice,p=1,status=1;
    while(status)
    {
    printf("------------------------------------------------------------\n1:Delete node at the beginning\n
    2:Delete node at the end\n3:Display all nodes' data\n4:Product of all nodes' data\n5:Search data\n
    6:quit\n------------------------------------------------------------\n");
    printf("Enter your choice:");
    scanf("%d",&choice);
```

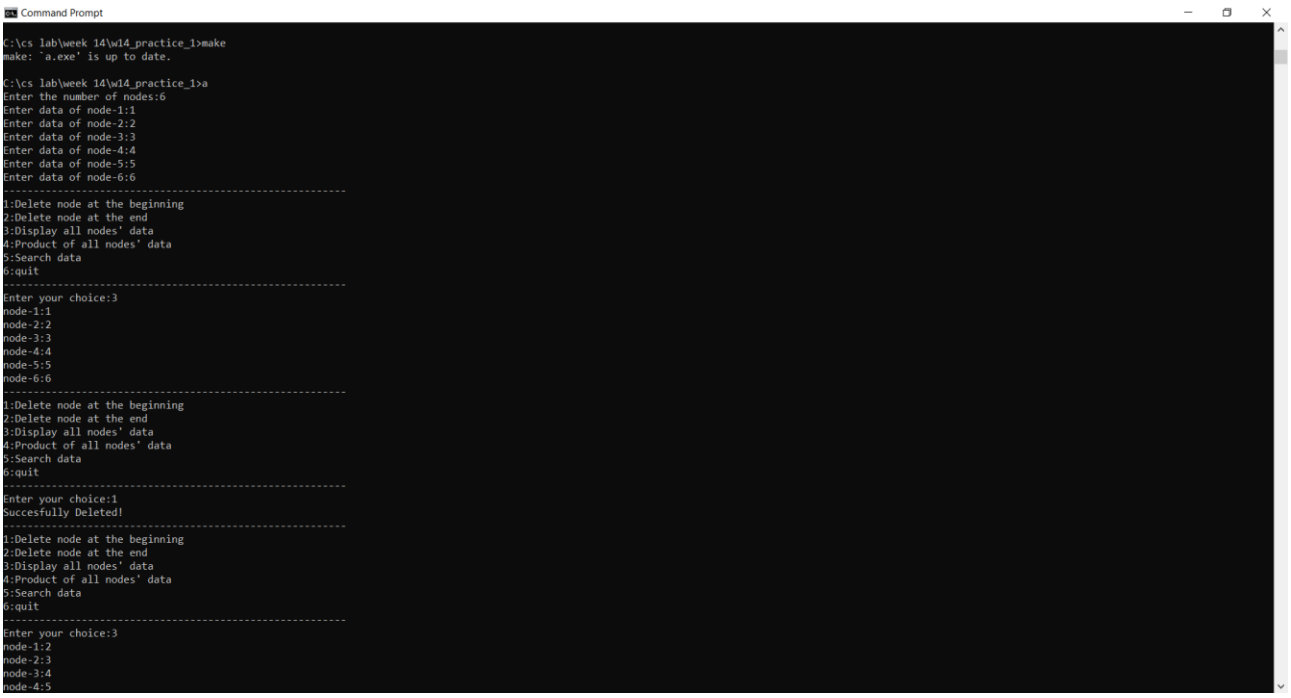
```c
    switch(choice)
    {
        case 1:head=beg(head);
        break;
        case 2:head=end(head);
        break;
        case 3:display(head);
        break;
        case 4:p=productall(head);
        printf("Product of all nodes' data=%d\n",p);
        p=1;
        break;
        case 5:search(head);
        break;
        case 6:status=0;
        break;
        default:printf("Enter a choice between 1 and 8 only\nTry again!\n");

    }
    }
    free(head);
    free(temp);
    free(newnode);
    return 0;
}

```

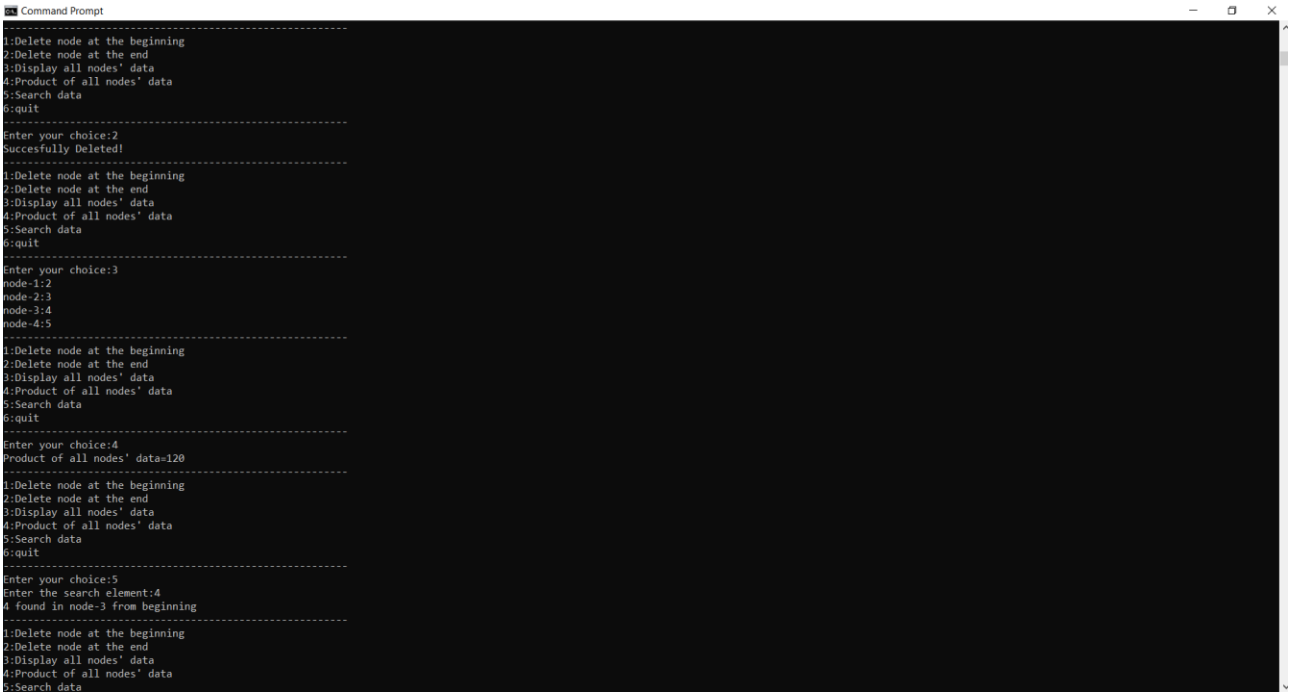
Server.h

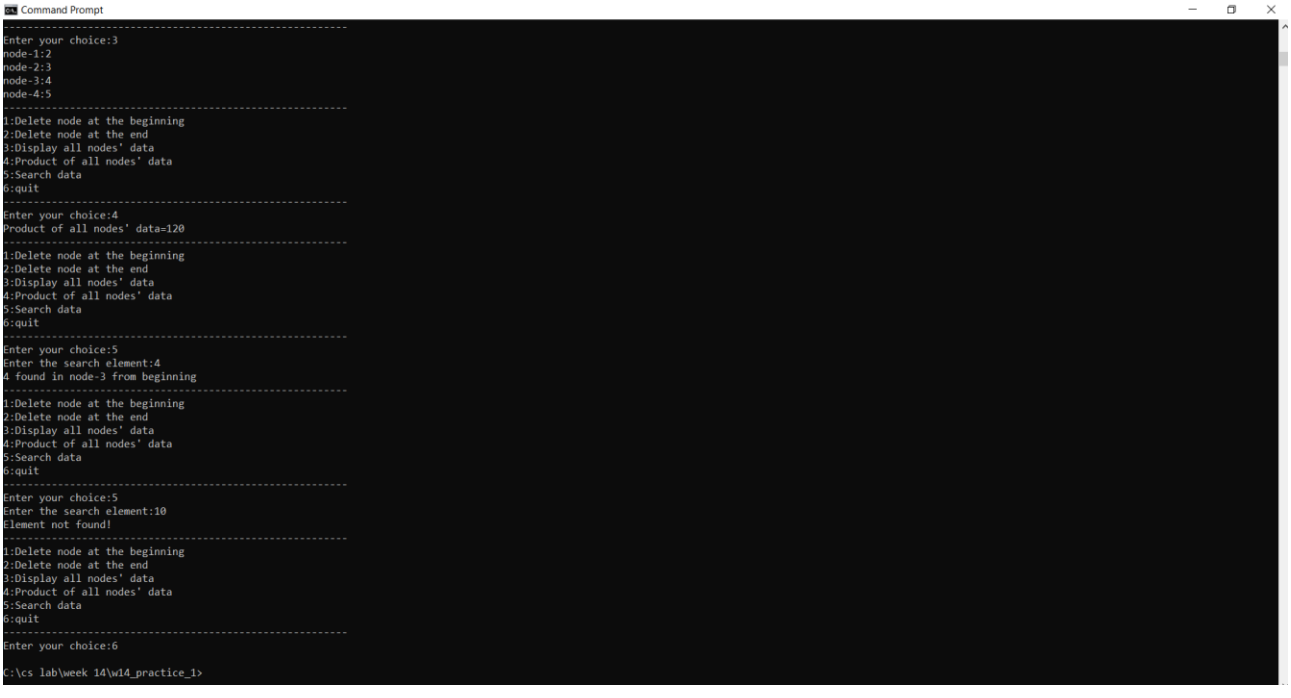
server.h > search(node *)

```c
struct node
{
    int data;
    struct Node *next;
};
int productall(struct node *);
void search(struct node *);
void display(struct node *);
struct node* beg(struct node *);
struct node* end(struct node *);

```

Output




Command Prompt
1:Delete node at the beginning
2:Delete node at the end
3:Display all nodes' data
4:Product of all nodes' data
5:Search data
6:quit
Enter your choice:2
Succesfully Deleted!
1:Delete node at the beginning
2:Delete node at the end
3:Display all nodes' data
4:Product of all nodes' data
5:Search data
6:quit
Enter your choice:3
node-1:2
node-2:3
node-3:4
node-4:5
1:Delete node at the beginning
2:Delete node at the end
3:Display all nodes' data
4:Product of all nodes' data
5:Search data
6:quit
Enter your choice:4
Product of all nodes' data=120
1:Delete node at the beginning
2:Delete node at the end
3:Display all nodes' data
4:Product of all nodes' data
5:Search data
6:quit
Enter your choice:5
Enter the search element:4
4 found in node-3 from beginning
1:Delete node at the beginning
2:Delete node at the end
3:Display all nodes' data
4:Product of all nodes' data
5:Search data


Command Prompt
Enter your choice:3
node-1:2
node-2:3
node-3:4
node-4:5
1:Delete node at the beginning
2:Delete node at the end
3:Display all nodes' data
4:Product of all nodes' data
5:Search data
6:quit
Enter your choice:4
Product of all nodes' data=120
1:Delete node at the beginning
2:Delete node at the end
3:Display all nodes' data
4:Product of all nodes' data
5:Search data
6:quit
Enter your choice:5
Enter the search element:4
4 found in node-3 from beginning
1:Delete node at the beginning
2:Delete node at the end
3:Display all nodes' data
4:Product of all nodes' data
5:Search data
6:quit
Enter your choice:5
Enter the search element:10
Element not found!
1:Delete node at the beginning
2:Delete node at the end
3:Display all nodes' data
4:Product of all nodes' data
5:Search data
6:quit
Enter your choice:6
C:\cs lab\week 14\w14_practice_1>