Lab Assignment 2

Name – Akshat Chandrapatle

Roll no. – 45

Div. – CS-A

PRN – 12111449

Write an Append () function that takes two lists, 'a' and 'b', appends 'b' onto the end of 'a', and then sets 'b' to NULL (since it is now trailing off the end of 'a').

Code:

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node *ptr;
} *temp , *new;


struct Node* insert(struct Node *first, int x){
   if(first==NULL){
      new = (struct Node*)malloc(sizeof(struct Node));
      new->data=x;
      new->ptr=NULL;
      return new;
   }
   else{
      new = (struct Node*)malloc(sizeof(struct Node));
      new->data=x;
      new->ptr=NULL;
      temp=first;
```

```
        while(temp->ptr!=NULL){
            temp=temp->ptr;
        }
        temp->ptr=new;
        return first;
    }
}

struct Node *append(struct Node *first, struct Node *second){
    temp=first;
    while(temp->ptr!=NULL){
        temp=temp->ptr;
    }
    temp->ptr=second;
    return first;
}

void printList(struct Node* first){

    temp=first;
    while(temp!=NULL){
        printf("[Current_node:%p data:%d Next_node%p]\n",temp,temp->data ,temp-
>ptr);
        temp=temp->ptr;
    }
    printf("\n");
}

int main()
{
    struct Node *a =NULL;
    int size,n;

    printf("Enter the size of a list:");

    scanf("%d",&size);
    printf("\nEnter the data for a list: ");
    for(int i = 0; i<size; i++){
        scanf("%d",&n);
        a = insert(a,n);
```

```
    }

    printList(a);

    struct Node *b = NULL;
    printf("\n Enter the size of b list:");

    scanf("%d",&size);
    printf("\nEnter the data for b list: ");
    for(int i = 0; i<size; i++){
        scanf("%d",&n);
        b = insert(b,n);
    }

    printList(b);

    a = append(a,b);

    printf("\nAfter appending \n");
    printList(a);
    b=NULL;
    printList(b);
    return 0;
}
```

Output:

```
Enter the size of a list:4

Enter the data for a list: 1
3
2
5
[Current_node:0x556058a9dac0 data:1 Next_node0x556058a9dae0]
[Current_node:0x556058a9dae0 data:3 Next_node0x556058a9db00]
[Current_node:0x556058a9db00 data:2 Next_node0x556058a9db20]
[Current_node:0x556058a9db20 data:5 Next_node(nil)]


 Enter the size of b list:3

Enter the data for b list: 1
2
3
[Current_node:0x556058a9db40 data:1 Next_node0x556058a9db60]
[Current_node:0x556058a9db60 data:2 Next_node0x556058a9db80]
[Current_node:0x556058a9db80 data:3 Next_node(nil)]


After appending
[Current_node:0x556058a9dac0 data:1 Next_node0x556058a9dae0]
[Current_node:0x556058a9dae0 data:3 Next_node0x556058a9db00]
[Current_node:0x556058a9db00 data:2 Next_node0x556058a9db20]
[Current_node:0x556058a9db20 data:5 Next_node0x556058a9db40]
[Current_node:0x556058a9db40 data:1 Next_node0x556058a9db60]
[Current_node:0x556058a9db60 data:2 Next_node0x556058a9db80]
[Current_node:0x556058a9db80 data:3 Next_node(nil)]
```