PROGRAM 1
REVERSAL OF LINKED LIST

ALGORITHM:

Step 1: Start

Step 2: Declare structure
```
typedef struct node {
        int data;
        struct node *link;
}listnode;
```

Step 3: listnode *front = NULL;

Step 4: Define a create() to create a node

```
listnode *create(int value){
   listnode *node;
   node = (listnode*)malloc(sizeof(listnode));
   node->data = value;
   node->link = NULL;
   return node;
}
```

Step 5: Print Menu

Step6: Declare choice (int) and input values from user

Step7: Check if(choice!=9)
if true proceed to step 8 else go to step 9

Step 8: switch(choice)
```
case 1: call insert_begin(); break;
case 2: call insert_rand(); break;
case 3: call insert_end(); break;
case 4: call delete_begin(); break;
case 5: call delete_rand(); break;
case 6: call delete_end(); break;
case 7: call display(); break;
case 8: call reverse(); break;
case 3: exit
default : Print “Enter a valid choice”
```

Step 9: Stop

Reverse( ):
Algo

Step 1: Define reverse()
Step 2: Declare variables (int) count and initialize it to 1, i and ct;
Step 3: listnode *temp = front
        Declaring temp
Step 4: listnode *ptr;
        Declaring ptr
Step 5: Start While loop check for temp->link!=NULL
        Step 5.1: count++;
        Step 5.2: temp=temp->link;
Step 6: ptr = temp;
Step 7: Exit While
Step 8: Start For loop
        For i=0 till i<(count-1) increment i++;
        Step 8.1: temp = front->link;
                   front->link = ptr->link;
                   ptr->link = front;
                   front = temp;
Loop 8.1

Step 9: Exit For loop
Step 10: front = ptr;
Step 11: Print The reversed list
Step 12: call display();