ASSIGNMENT # 1
CODE:

```cpp
#include<iostream>
using namespace std;
class node{
public:
int info;
node *next;
node(int val){
info=val;
next=NULL;
}
};
class linkedlist{
private:
node* head;
int length;
public:
linkedlist(){
head=NULL;
length=0;
}
void insertion(int val,int pos){
node* n=new node(val);
if(head==NULL){
n->next=head;
head=n;
length++;
return;
}
else{
node* temp=head;
for(int i=1;i<pos-1;i++){
temp=temp->next;
}
n->next=temp->next;
temp->next=n;
length++;
}
}
void insertafterevenvalue(int val){
if(head==NULL){
return;
}
node* temp=head;
while(temp!=NULL){
if(temp->info%2==0){
node* n=new node(val);
n->next=temp->next;
temp->next=n;
```

```cpp
length++;
break;
}
temp=temp->next;
}
}
void swapPairs() {
node* n = new node(0);
n->next = head;
node* curr = n;

while (curr->next != NULL && curr->next->next != NULL) {
node* first = curr->next;
node* second = curr->next->next;
first->next = second->next;
curr->next = second;
curr->next->next = first;
curr = curr->next->next;
}
}

void palindrome(){
node* n=head;
int x=0;
int rev=0;
while(n!=NULL){
x=(x*10)+n->info;
n=n->next;
}
int a=x;
do{
int v=x%10;
rev=(rev*10)+v;
x=x/10;
}while(x!=0);
if(a==rev){
cout<<"LINKED LIST IS PALINDROME:"<<endl;
}
else{
cout<<"LINKED LIST IS NOT PALINDROME:"<<endl;
}

}

void removeDuplicates() {
if (head == NULL) {
return ;
}

node* curr = head;
while (curr->next != NULL) {
```

```cpp
            if (curr->info == curr->next->info) {
            node* temp = curr->next;
            curr->next = curr->next->next;
            delete temp;
            }
            else {
            curr = curr->next;
            }
            }
            }
            void printing(){
            node* n=head;
            while(n!=NULL){
            cout<<n->info<<"\t";
            n=n->next;
            }
            cout<<endl;

            }
            };
            int main(){
            linkedlist l1;
            cout<<"-------inserting nodes:-------"<<endl;
            l1.insertion(1,1);
            l1.insertion(1,2);
            l1.insertion(3,3);
            l1.insertion(4,4);
            l1.insertion(5,5);
            l1.insertion(6,6);

            l1.printing();



            // l1.insertafterevenvalue(7);
            // l1.printing();



            // l1.palindrome();



            // l1.removeDuplicates();
            // l1.printing();

            // l1.swapPairs();
            // l1.printing();




            }
```

OUTPUTS:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

cd "/home/captain/Documents/semester 4/Data lab/" && g++ lab
(base) captain@captain-AB:~$ cd "/home/captain/Documents/sem
/Data lab/"labassignment
-------inserting nodes:-------
1        1        3        4        5        6
removing duplicates:
1        3        4        5        6
(base) captain@captain-AB:~/Documents/semester 4/Data lab$
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

cd "/home/captain/Documents/semester 4/Data lab/" && g++ lab
(base) captain@captain-AB:~$ cd "/home/captain/Documents/sem
/Data lab/"labassignment
-------inserting nodes:-------
1        1        3        4        5        6
LINKED LIST IS NOT PALINDROME:
(base) captain@captain-AB:~/Documents/semester 4/Data lab$
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

cd "/home/captain/Documents/semester 4/Data lab/" && g++ lab
(base) captain@captain-AB:~$ cd "/home/captain/Documents/sem
/Data lab/"labassignment
-------inserting nodes:-------
1        1        3        4        5        6
swaping pairs:
1        4        3        6        5
(base) captain@captain-AB:~/Documents/semester 4/Data lab$
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

cd "/home/captain/Documents/semester 4/Data lab/" && g++ lab
(base) captain@captain-AB:~$ cd "/home/captain/Documents/sem
/Data lab/"labassignment
-------inserting nodes:-------
1        1        3        4        5        6
inserting after even value:
1        1        3        4        7        5        6
(base) captain@captain-AB:~/Documents/semester 4/Data lab$
```