**EVALUTION-1**

#include <iostream>

using namespace std;

class node

{

public:

int data;

node \*nextadd;

node \*prev;

node \*head;

void create(int A[],int x)

{

node \*last;

node \*temp;

head=new node;

head->data=A[0];

head->nextadd=NULL;

head->prev=NULL;

last=head;

for (int i=1;i<7;i++)

{

temp=new node;

temp->data=A[i];

temp->nextadd=NULL;

temp->prev=head;

last->nextadd=temp;

temp->prev=temp->nextadd;

last=temp;

}

}

void addnode(node \*p,int index,int x)

{

node \*t;

int i;

if (index<0||index>7)

return;

t=new node;

t->data=x;

if (index==0)

{

p->nextadd=head;

head=t;

}

else{

for (i=1;i<index-1;i++)

{

p=p->nextadd;

p=p->prev;

}

t->nextadd=p->nextadd;

p->nextadd=t;

p->nextadd=p->prev;

}

}

void display(node \*p)

{

while(p!=NULL)

{

cout<<p->data<<"<->";

p=p->nextadd;

p=p->prev;

}

}

int sum(Node \*p)

{

int s=0;

while(p!=NULL)

{

s+=p->data;

p=p->next;

}

return s;

}

void Rsum(Node \*p)

{

if(p==NULL)

return 0;

else

return Rsum(p->next)+p->data;

}

};

int main()

{

int A[]={1,2,4,5,6,8,9};

node obj;

obj.create(A,7);

obj.addnode(obj.head,1,23);

obj.display(obj.head);

obj.sum(obj.head);

obj.Rsum(obj.head);

return 0;

}

output

