**PROGRAM: 29**

**8-10-2019** **INSERT A NEW ELEMENT IN SORTED ORDER**

**OBJECTIVES:**

Let List be a single linked list with N elements in the sorted order. Write a C++ program using function INSERTNODE () to insert a new element which ensures that the sorted order of the remaining elements is not affected. Do not append and sort the list again. You can assume the linked list is containing numbers.

**SOURCE CODE:**

#include<iostream.h>

#include<conio.h>

struct Node

{

float value;

Node \*next;

};

class List

{

private:

Node \*head;

public:

List(void)

{ head = NULL; }

void appendNode(float);

void insertNode(float);

void displayList(void);

};

void List::appendNode(float num)

{

Node \*nptr, \*ptr;

nptr = new Node;

nptr->value = num;

nptr->next = NULL;

if (!head)

head = nptr;

else

{

ptr = head;

while (ptr->next)

ptr = ptr->next;

ptr->next = nptr;

}

};

void List::displayList(void)

{

Node \*ptr;

ptr = head;

while (ptr!=NULL)

{

cout << ptr->value <<" ";

ptr = ptr->next;

}

};

void List::insertNode(float num)

{

Node \*nptr, \*ptr, \*previousNode= NULL;

nptr = new Node;

nptr->value = num;

if (!head)

{

head = nptr;

nptr->next = NULL;

}

else

{ ptr = head;

while (ptr != NULL && ptr->value < num)

{

previousNode = ptr;

ptr = ptr->next;

}

if (previousNode == NULL)

{

head = nptr;

nptr->next = ptr;

}

else

{

previousNode->next = nptr;

nptr->next = ptr;

}

}

};

void main()

{cout<<"------------------------------------------------------------------------"<<endl;

cout<<"\t\tPROGRAM 29\n";

cout<<"-------------------------------------------------------------------------"<<endl;

List list;

list.appendNode(2.5);

list.appendNode(7.9);

list.appendNode(12.6);

cout<<"\nTHE LINKED LIST PRESENT BY DEFAULT IN SORTED ORDER: "<<endl;

list.displayList();

int cho=1;

float val;

while(cho==1)

{cout<<"\nENTER THE VALUE: ";

cin>>val;

list.insertNode(val);

cout<<"\npress 1 to continue, else press 0"<<endl;

cin>>cho;

}

cout<<"\nTHE NEW LIST IS: "<<endl;

list.displayList();

getch();

}

**SAMPLE OUTPUT:** 