**Rohit Karunakaran Roll no 58**

**Program code:**

#include<stdio.h>

#include<stdlib.h>

typedef struct msll{

int arr[100];

struct msll \*link;

int size;

}msll;

void sortList(msll\*\* Header)

{

if((\*Header)==NULL)

{

printf("The List is empty]\n");

}

else

{

msll \*temp = (msll\*) malloc(sizeof(msll));

msll \*ptr = (\*Header)->link;

//temp ->link =(\*Header);

/\*

temp ->link =(\*Header);

msll \*ptr2 = (\*Header)->link;

msll \*temp2 = (msll\*)malloc(sizeof(msll));

while(temp->link!=NULL){

}

\*/

temp = (\*Header);

while(temp->link!=NULL){

ptr = temp->link;

temp->link=ptr->link;//increment the temp

if((\*Header)->link==NULL){

(\*Header)->link = ptr;

ptr->link=NULL;

}

else{

msll\* ptr1 = (\*Header)->link;

msll\* ptr2 = (\*Header);

while(ptr1!=NULL && ptr1->arr[0]<=ptr->arr[0]){

ptr1=ptr1->link;

ptr2 = ptr2->link;

}

ptr2->link=ptr;

ptr->link = ptr1;

}

}

free(temp);

}

}

void insert(msll \*\*Header, int data, int size){

msll \*ptr = (\*Header);

msll \*ptr1 =NULL;

int flag=0;

while(ptr!=NULL){

if(ptr->size<size){

flag=1;

break;

}

ptr1=ptr;

ptr = ptr->link;

}

if(flag){

ptr->arr[ptr->size] = data;

ptr->size=ptr->size+1;

}

else{

msll\* node = (msll\*)malloc(sizeof(msll));

node->arr[node->size]=data;

node->size = 1;

ptr1->link = node;

}

}

void display(msll \*Header,int size){

msll \*ptr = Header;

while(ptr!=NULL){

int i=0;

while(i<ptr->size){

printf("%d ",ptr->arr[i]);

i++;

}

printf("\n");

ptr = ptr->link;

}

}

int main(){

int n = 4;

int m = 5;

int elem;

printf("enter number of elements of the array: ");

scanf("%d",&n);

printf("Enter the value of m: ");

scanf("%d",&m);

msll \*Header = (msll\*)malloc(sizeof(msll));

int i = 0;

while(i<n){

scanf("%d",&elem);

insert(&Header,elem,m);

i++;

}

printf("The elmentes are: ");

display(Header,m);

printf("\n The sorted elements are: ");

sortList(&Header);

display(Header,m);

}

