Name: - Atish Kumar

Roll No: - 120CS0173

Lab Sheet: - 05

Q2. Write a program to transpose a sparse matrix represented as an array of nx3 size, as mentioned in problem 1.A. The time complexity of the algorithm should be order of n and the space complexity should be order of 1.

Program:-

```
#include<stdio.h>
int main()
{
 int n,i;
 printf("How many values stored in your n*3 array\n"
);
 scanf("%d",&n);
 int A[3][n];
 for(i =0 ;i<n;i++)
 {
 printf("\nEnter row ,column and value : ");
 scanf("%d%d%d",&A[0][i],&A[1][i],&A[2][i]);
 }
 int Transpose[3][n];
 for(i=0;i<n;i++)
 {

 Transpose[0][i] = A[1][i];
 Transpose[1][i] = A[0][i];
 Transpose[2][i] = A[2][i];

 }
 printf(" 3 by n format of transposed matrix is :\n")
;
```

```
 printf("r : ");
 for(i=0;i<n;i++)
 {
 printf("%d ",Transpose[0][i]);
 }
 printf("\n");
 printf("c : ");
 for(i=0;i<n;i++)
 {
 printf("%d ",Transpose[1][i]);
 }
 printf("\n");
 printf("v : ");
 for(i=0;i< n;i++)
 {
 printf("%d ",Transpose[2][i]);
 }
 printf("\n");
}
```

Output;-

TDM-GCC 4.9.2 64

C:\Users\atish\Desktop\2.exe

```
How many values stored in your n*3 array
3

Enter row ,column and value : 58
63
25

Enter row ,column and value : 85
63
25

Enter row ,column and value : 35
85
32
 3 by n format of transposed matrix is :
r : 63 63 85
c : 58 85 35
v : 25 25 32

--------------------------------
Process exited after 16.68 seconds with return value 10
Press any key to continue . . .
```

Q3. Due to a rush at the end of the exam, Professor could not arrange the answer sheets in the sequence as desired. However, he mentioned the correct location of each answer sheet with it. Now, Professor wants all the answer sheets in the desired sequence.

Program:-

```
#include<stdio.h>

#include<stdlib.h>

struct sheet

{

 int item ;

 int location;

 struct sheet *next;

};

struct sheet *createNode(int ,int );

struct sheet * append(struct sheet *,struct sheet *);

struct sheet * sort(struct sheet *);

struct sheet * insert_sorted(struct sheet *, struct sheet *);

struct sheet * insertAtBeginning(struct sheet * s,struct sheet * );

void display(struct sheet * );

void freeNodes(struct sheet *);

int main()

{

 struct sheet *start,*newptr,*end,*sortedstart;

 int n;

 printf("How sheets you want to store/arrange \n");

 scanf("%d",&n);

 int i;
```

```
 int a,b;
 for(i=0;i<n;i++)
 {
 printf("Give the item data and location\n");
 scanf("%d%d",&a,&b);
 newptr = createNode(a,b);
 if(i==0)
 start = end = newptr;
 else
 end = append(end,newptr);
 }
 sortedstart = sort(start);
 printf("\n Unorded sheets : ");
 display(start);
 printf("\n");
 printf(" orded sheets : ");
 display(sortedstart);
 return 0;

}
struct sheet *createNode(int a,int b)
{
 struct sheet * ptr;
 ptr = (struct sheet * )malloc(sizeof(struct sheet));
 ptr->item = a;
 ptr->location = b;
 ptr->next = NULL;
 return ptr;
}
struct sheet * append(struct sheet * end,struct sheet *
newptr)
```

```
{
 end->next = newptr;
 return newptr;
}
struct sheet * sort(struct sheet * start)
{

 struct sheet * ptr ;
 ptr = start;
 struct sheet * newNode;
 struct sheet * sortedFirst;
 if(start!=NULL)
 {
 sortedFirst = createNode(ptr->item,ptr->location);
 ptr = ptr->next;
 }
 while(ptr!=NULL)
 {
 newNode = createNode(ptr->item,ptr->location);
 sortedFirst = insert_sorted(sortedFirst,newNode);
 ptr= ptr->next;
 }
 return (sortedFirst);
}
struct sheet * insert_sorted(struct sheet *start, struct sheet *newNode)
{
 struct sheet * ptr = start;
 struct sheet * prev =NULL;
 while((ptr!=NULL) && (ptr->location < newNode->location))
 {
 prev = ptr ;
```

```
 ptr = ptr->next;
 }
 if(prev == NULL)
 {
 start = insertAtBeginning(start,newNode);
 }
 else
 {
 prev->next = newNode;
 newNode->next = ptr;
 }

 return (start);
}
struct sheet * insertAtBeginning(struct sheet * start,struct
sheet * newNode)
{
 newNode->next = start;
 return newNode;
}
void freeNodes(struct sheet * start)
{
 struct sheet * ptr;
 ptr = start;
 while (ptr!=NULL)
 {
 start = ptr;
 ptr = ptr->next;
 free(ptr);
 }
}
```

```
void display(struct sheet * start)
{
 struct sheet * ptr;
 ptr = start;
 while(ptr!=NULL)
 {
 printf(" %d ",ptr->item);
 printf("%d ",ptr->location);
 ptr = ptr->next;
 }
}
```

Output:-

```
C:\Users\atish\Desktop\3.exe
How sheets you want to store/arrange
5
Give the item data and location
86
56
Give the item data and location
25
56
Give the item data and location
82
65
Give the item data and location
052
05
Give the item data and location
23
02

 Unorded sheets :  86 56  25 56  82 65  52 5  23 2
 orded sheets :  23 2  52 5  25 56  86 56  82 65
--------------------------------
Process exited after 17.6 seconds with return value 0
Press any key to continue . . .
```