

# DATASTRUCTURE LAB

NAME: RIYA VINCENT

REGISTER NUMBER: TKM20MCA-2031

ROLL NO :29(OLD)

NEW ROLLNO: MCA231

## ALGORITHM:

Data Structure - LAB  
Riya Vincent  
Roll no: MCA231  
Reg no: TKM20MCA-2031

Algorithm  
Question No:1  
consider the directed acyclic graph is

```
graph LR; A((A)) --> B((B)); B --> C((C)); B --> D((D)); C --> E((E)); D --> E; D --> G((G)); E --> F((F)); F --> G;
```

Implement topological sorting

Algorithm  
step 1: for each vertex  $U \in V$   
step 2: do  $\text{indegree}[U] \leftarrow 0$   
step 3: for each vertex  $U \in V$   
step 4: do for each  $v \in \text{Adj}[U]$   
step 5: do  $\text{indegree}[v] \leftarrow \text{indegree}[v] + 1$

Adjacency Matrix

	A	B	C	D	E	F	G
A	0	1	0	0	0	0	0
B	0	0	1	1	0	0	0
C	0	0	0	0	1	0	0
D	0	0	0	0	1	0	1
E	0	0	0	0	0	1	0
F	0	0	0	0	0	0	1
G	0	0	0	1	0	0	0

Step 6:  $Q \leftarrow \emptyset$   
 Step 7: for each vertex  $U \in V$   
 Step 8: Do if  $\text{indegree}[U] = 0$   
 Step 9: Then Enqueue( $Q, U$ )  
 Step 10: while  $Q \neq \emptyset$   
 Step 11: Do  $U \leftarrow \text{Dequeue}(Q)$   
 Step 12: Output  $U$   
 Step 13: for each  $V \in \text{Adj}[U]$   
 Step 14: Do  $\text{indegree}[V] \leftarrow \text{indegree}[V] - 1$   
 Step 15: If  $\text{indegree}[V] = 0$   
 Step 16: Then Enqueue( $Q, V$ )  
 Step 17: Do if  $\text{indegree}[U] \neq 0$   
 Step 18: Repeat there is a cycle.  
 Step 19: stop

## Source code:

```

#include <stdio.h>

int main(){
    int i,j,k,n,a[10][10],indeg[10],flag[10],count=0;
    printf("topological sorting\n");
    printf("_____ \n");
    printf("\n");
    printf("Enter the no of vertices in the given graph:\n");
    scanf("%d",&n);
  
```

```

printf("Enter Adjacency matrix:\n");
for(i=0;i<n;i++){
    printf("Enter elements of row  %d\n",i+1);
    for(j=0;j<n;j++)
        scanf("%d",&a[i][j]);

}

for(i=0;i<n;i++){
    indeg[i]=0;
    flag[i]=0;
}

for(i=0;i<n;i++)
    for(j=0;j<n;j++)
        indeg[i]=indeg[i]+a[j][i];

printf("\nTopological order is:");
printf("\n");

while(count<n){
    for(k=0;k<n;k++){
        if((indeg[k]==0) && (flag[k]==0)){
            printf("%d", (k+1));
            if(k+1==1)
            {
printf("(A) \t");
            }

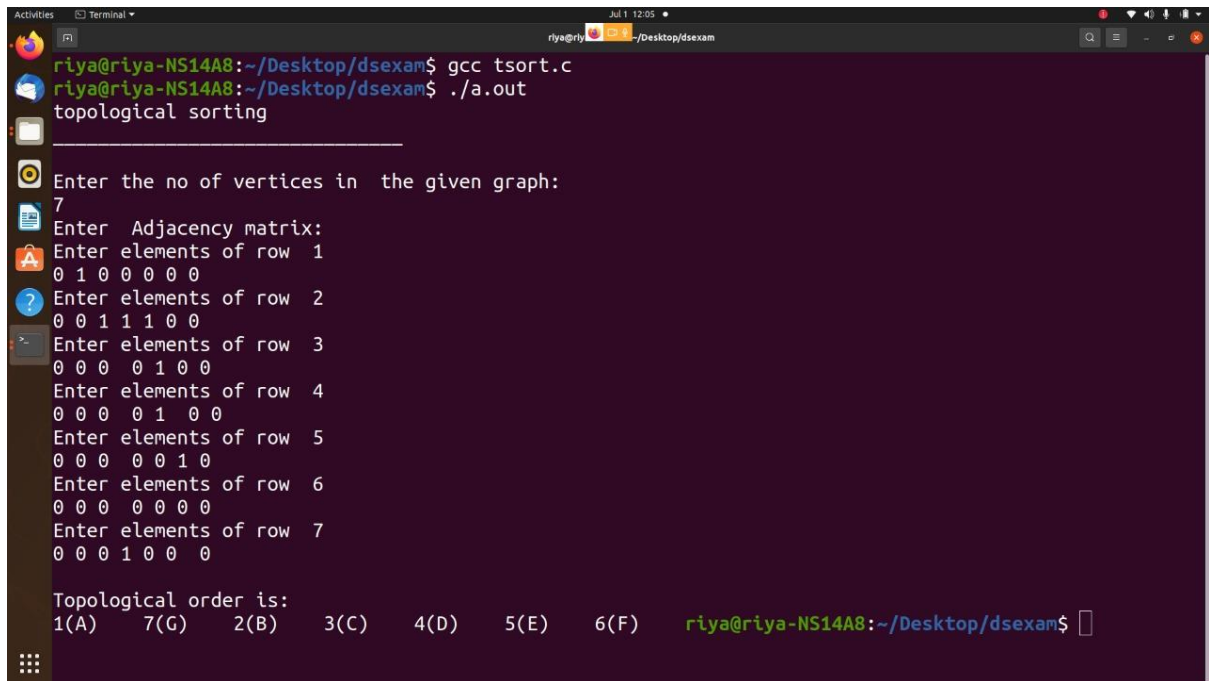
            else if(k+1==2)
            {
printf("(B) \t");

```

```
    }  
    else if(k+1==3)  
    {  
printf("(C) \t");  
    }  
    else if(k+1==4)  
    {  
printf("(D) \t");  
    }  
    else if(k+1==5)  
    {  
printf("(E) \t");  
    }  
    else if(k+1==6)  
    {  
printf("(F) \t");  
    }  
    else if(k+1==7)  
    {  
printf("(G) \t");  
  
    }  
    else  
printf("=");  
        flag [k]=1;  
    }  
    for(i=0;i<n;i++){  
        if(a[i][k]==1)  
            indeg[k]--;  
    }  
}
```

```
        count++;  
    }  
  
    return 0;  
}
```

## OUTPUT



A terminal window titled "Terminal" showing the execution of a C program for topological sorting. The user is at the prompt `riya@riya-NS14A8:~/Desktop/dsexam$`. They compile `gcc tsort.c` and run `./a.out`. The program prompts for the number of vertices (7) and then for the adjacency matrix row by row. The output shows the topological order: 1(A) 7(G) 2(B) 3(C) 4(D) 5(E) 6(F).

```
riya@riya-NS14A8:~/Desktop/dsexam$ gcc tsort.c  
riya@riya-NS14A8:~/Desktop/dsexam$ ./a.out  
topological sorting  
  
Enter the no of vertices in the given graph:  
7  
Enter Adjacency matrix:  
Enter elements of row 1  
0 1 0 0 0 0 0  
Enter elements of row 2  
0 0 1 1 1 0 0  
Enter elements of row 3  
0 0 0 0 1 0 0  
Enter elements of row 4  
0 0 0 0 1 0 0  
Enter elements of row 5  
0 0 0 0 0 1 0  
Enter elements of row 6  
0 0 0 0 0 0 0  
Enter elements of row 7  
0 0 0 1 0 0 0  
  
Topological order is:  
1(A) 7(G) 2(B) 3(C) 4(D) 5(E) 6(F) riya@riya-NS14A8:~/Desktop/dsexam$
```

## QUESTION NO:2

### ALGORITHM:

Question no:2 doubly linked list

Write a program for creating doubly linked list and perform the following operations:

- ① Insertion of an element at a particular position
- ② Search of an element
- ③ Delete an element at the end of the list

### Algorithm

#### Insertion

Begin (at the beginning)

1. if  $\text{start} = \text{NULL}$

$\text{start} = t$

\*) else

$t \rightarrow \text{next} = \text{NULL}$

$t \rightarrow \text{next} \rightarrow \text{prev} = t$

$\text{start} = t$

Return

at a particular position

Step 1: print "enter info/data of the node after which you want to insert."

Step 2: Read

### Algorithm

1.  $t = \text{new node}$

2. Enter "the data to be inserted"

3. Read  $n$

4.  $\&t \rightarrow \text{info} = n$

5.  $t \rightarrow \text{next} = \text{NULL}$

6.  $t \rightarrow \text{prev} = \text{NULL}$

Step 3:  $P \leftarrow \text{start}$

Step 4: ~~if  $P \rightarrow \text{next} = \text{NULL}$~~

Repeat while  $P \neq \text{NULL}$

if  $(P \rightarrow \text{info} = t)$

$t \rightarrow \text{next} = P \rightarrow \text{next}$

$P \rightarrow \text{next} = t$

$t \rightarrow \text{prev} = P$

$P \rightarrow \text{next} \rightarrow \text{prev} = t$

Return

else

$P \leftarrow P \rightarrow \text{next}$

Step 5: print  $x$  and  $P$  and

$t \rightarrow \text{next} = \text{NULL}$

$P \rightarrow \text{next} = t$

Algorithm for deletion at end

Step 1:  $P \leftarrow \text{start}$

Step 2: Repeat while  $P \neq \text{NULL}$

if  $(P \rightarrow \text{next} = \text{NULL})$

DeleNode( $P$ )

Step 3: Return

### Algorithm for display

Step 1:  $p = \text{start}$   
Step 2: Repeat step 3 <sup>while</sup>  $p \neq \text{NULL}$   
Step 3: print  $p \rightarrow \text{info}$

### Algorithm for deletion

Step 1: If  $\text{Head} == \text{NULL}$   
    write "underflow"  
    go to step 8  
    [End of IF]  
Step 2: Set  $\text{PTR} = \text{HEAD}$   
Step 3: Set  $i = 0$   
Step 4: Repeat step 5 to 7 while  $\text{PTR} \neq \text{NULL}$   
Step 5: If  $\text{PTR} \rightarrow \text{data} = \text{data}$   
    return  $i$   
    [End of IF]  
Step 6:  $i = i + 1$   
Step 7:  $\text{PTR} = \text{PTR} \rightarrow \text{next}$   
Step 8: Exit

### SOURCE CODE:

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
struct node
```

```
{
```

```
    struct node *prev;
```



```

    struct node *next;

    int data;
};

struct node *head;

void insertion_beginning();
void insertion_specified();
void deletion_last();
void display();
void search();
void main ()
{
    int choice =0;

    while(choice != 9)
    {
        printf("\n****Main Menu****\n");
        printf("\nChoose one option from the following list ...\n");
        printf("\n===== \n");
        printf("\n1.Insert in beginning\n2.Insert at any random location\n3.Delete from
last\n4.Search\n5.display elements\n6.exit\n");
        printf("\nEnter your choice?\n");
        scanf("\n%d",&choice);
        switch(choice)
        {
            case 1:
                insertion_beginning();
                break;
            case 2:
                insertion_specified();
                break;
            case 3:
                deletion_last();

```

```
        break;
    case 4:
        search();
        break;
    case 5:
        display();
        break;
    case 6:
        exit(0);
        break;
    default:
        printf("Please enter valid choice..");
    }
}
```

```
void insertion_beginning()
{
    struct node *ptr;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter Item value");
        scanf("%d",&item);
```

```

if(head==NULL)
{
    ptr->next = NULL;
    ptr->prev=NULL;
    ptr->data=item;
    head=ptr;
}
else
{
    ptr->data=item;
    ptr->prev=NULL;
    ptr->next = head;
    head->prev=ptr;
    head=ptr;
}
printf("\nNode inserted\n");
}

}

```

```

void insertion_specified()
{
    struct node *ptr,*temp;
    int item,loc,i;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\n OVERFLOW");
    }
    else

```

```

{
    temp=head;
    printf("Enter the location");
    scanf("%d",&loc);
    for(i=0;i<loc;i++)
    {
        temp = temp->next;
        if(temp == NULL)
        {
            printf("\n There are less than %d elements", loc);
            return;
        }
    }
    printf("Enter value");
    scanf("%d",&item);
    ptr->data = item;
    ptr->next = temp->next;
    ptr -> prev = temp;
    temp->next = ptr;
    temp->next->prev=ptr;
    printf("\nnode inserted\n");
}
}

```

```

void deletion_last()
{
    struct node *ptr;
    if(head == NULL)
    {
        printf("\n UNDERFLOW");
    }
}

```

```

    }
    else if(head->next == NULL)
    {
        head = NULL;
        free(head);
        printf("\nnode deleted\n");
    }
    else
    {
        ptr = head;
        if(ptr->next != NULL)
        {
            ptr = ptr -> next;
        }
        ptr -> prev -> next = NULL;
        free(ptr);
        printf("\nnode deleted\n");
    }
}

```

```

void display()
{
    struct node *ptr;
    printf("\n printing values...\n");
    ptr = head;
    while(ptr != NULL)
    {
        printf("%d\n",ptr->data);
        ptr=ptr->next;
    }
}

```

```

}

void search()
{
    struct node *ptr;
    int item,i=0,flag;
    ptr = head;
    if(ptr == NULL)
    {
        printf("\nEmpty List\n");
    }
    else
    {
        printf("\nEnter item which you want to search?\n");
        scanf("%d",&item);
        while (ptr!=NULL)
        {
            if(ptr->data == item)
            {
                printf("\nitem found at location %d ",i+1);
                flag=0;
                break;
            }
            else
            {
                flag=1;
            }
            i++;
            ptr = ptr -> next;
        }
        if(flag==1)

```

```

    {

        printf("\nItem not found\n");

    }

}

}

```

## OUTPUT

```

r@rtya-NS1448:~/Desktop/dsexam$ gcc dll.c
r@rtya-NS1448:~/Desktop/dsexam$ ./a.out
*****Main Menu*****
Choose one option from the following list ...
*****
1.Insert in beginning
2.Insert at any random location
3.Delete from last
4.Search
5.display elements
6.exit
Enter your choice?
1
Enter Item value:101
Node Inserted
*****Main Menu*****
Choose one option from the following list ...
*****
1.Insert in beginning
2.Insert at any random location
3.Delete from last
4.Search
5.display elements
6.exit
Enter your choice?
1
Enter Item value:102
Node Inserted
*****Main Menu*****
Choose one option from the following list ...

```

```
Activities Terminal Jul 1 12:12 rlye@rlye ~/Desktop/dsexam

*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at any random location
3.Delete from last
4.Search
5.display elements
6.exit
Enter your choice?
2
Enter the location1
Enter value211
node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at any random location
3.Delete from last
4.Search
5.display elements
6.exit
Enter your choice?
4
Enter item which you want to search?
211
item found at location 3
*****Main Menu*****
Choose one option from the following list ...
=====
```

```
Activities Terminal Jul 1 12:12 rlye@rlye ~/Desktop/dsexam
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at any random location
3.Delete from last
4.Search
5.display elements
6.exit
Enter your choice?
3
node deleted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at any random location
3.Delete from last
4.Search
5.display elements
6.exit
Enter your choice?
5
printing values...
102
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at any random location
3.Delete from last
4.Search
5.display elements
```



```
Activities Terminal Jul 1 12:12
riya@riya-NS14AB: ~/Desktop/dsexam

3.Delete from last
4.Search
5.display elements
6.exit
Enter your choice?
3
node deleted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at any random location
3.Delete from last
4.Search
5.display elements
6.exit
Enter your choice?
5
printing values...
102
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at any random location
3.Delete from last
4.Search
5.display elements
6.exit
Enter your choice?
6
riya@riya-NS14AB:~/Desktop/dsexam$ ./a.out
```