DS LAB PROGRAM 8 WRITE UP:-

mithil Raj

lab program (8) part I

STACK using linked list program

```
# include <stdio.h>
# include <conio.h>
struct node
 {
 int info;
 struct node * link;
 };
typedef struct node * NODE ;
 NODE getnode ()
 {
 NODE x ;
 x = (NODE) malloc (sizeof (struct node));
 if (x == NULL)
 {
 printf ("mem full \n");
 exit (0);
 }
 return x ;
 }
Void freenode (NODE x)
{
 free (x);
}
NODE temp;
temp = get node ();
temp -> info = item;
temp -> link = NULL;
if ( first == NULL)
 return first ;
 }
NODE delete - front (NODE first)
```

```
{
NODE temp;
if(first==NULL)
{
printf("stack is empty cannot delete\n");
return first;
}
temp=first;
temp=temp->link
printf("item deleted at front-end is=%d\n"
  first->info);
free(first);
return temp;
}
Void display(NODE first)
{
NODE temp;
}
Void display(NODE first)
{
NODE temp;
if(first==NULL)
printf("stack empty cannote display items\n")
for(temp=first;temp!=NULL;temp->link)
{
printf("%d\n", temp->info);
  }
}
Void main()
{
int item, choice, pos;
for(;;)
{
printf("\n1: Insert-front\n 2: Delete-fro
\n 3: Display-list \n 4: Exit\n");
printf("enter the choice \n");
scanf ("%d", &choice);
Switch (choice)
{
case 1: printf("enter the item at fron
-end \n");
Scanf("%d", &item);
first = insert_front(first, item);
 break;
case 2: first=delete_front(first);
 break;
case 3: display(first);
 break;
default: exit(0);
        break;
}
}
}
```

mithul Ray

lab program (8) part (2)

QUEUES using linked list

```
#include <stdio.h>
#include <conio.h>
struct node
{
 int info;
 struct node * link;
 };
typedef struct node * NODE;
NODE getnode()
 {
 NODE x;
x = (NODE) malloc(size of(struct node));
 if (x == NULL)
 {
printf("mem full\n");
 exit(0);
}
return x;
}
Void freenode(NODE x)
{
free(x);
}
NODE insert_rear(NODE first, int item)
{
NODE temp, cur;
temp = getnode()
 temp -> info = item
temp -> link = NULL;
if (first == NULL)
```

```
cur= first;
while (cur->link!= NULL)
cur= cur ->link;
cur->link= temp;
return first;
}
NODE delete_front(NODE first)
{
NODE temp;
if(first == NULL)
{
printf(" list is empty cannote delete i
return first;
}
temp = first
temp= temp->link;
printf(" item deleted at front end
is \n", first->info);
free(first)
return temp;
~~if~~ }
void display(NODE first)
{
NODE temp;
if (first == NULL)
printf(" list empty cannote display ite
\n");
for (temp= first; temp!= NULL; temp=
temp->link)
{
printf("%d \n", temp->info);
}
}
```

```
Void main()
{
int item, choice, pos;
NODE first = NULL;
for(;;)
{
printf("\n1. Insert-rear\n2. Delete-front
\n3. Display-list \n4. Exit \n");
printf("enter the choice\n");
scanf("%d", & choice)
switch(choice)
{
case1: printf("enter the item at rear
end\n");
Scanf("%d", & item);
first = insert-rear(first, item);
break;
case 2: first = delete-front(first);
break;
case 3: display(first);
break;
default: exit(0);
break;
}
}
}
```