DS PROGRAM

SINGLY LINKED LIST PROGRAM

```
#include<stdio.h>
#include<stdlib.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 insert_front(NODE first,int item)
{
NODE temp;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
return temp;
temp->link=first;
first=temp;
return first;
}
NODE delete_front(NODE first)
{
NODE temp;
if(first==NULL)
{
```

```
printf("list 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;
}
NODE insert_rear(NODE first,int item)
{
NODE temp,cur;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
 return temp;
cur=first;
while(cur->link!=NULL)
 cur=cur->link;
cur->link=temp;
return first;
```

```
}
NODE delete_rear(NODE first)
{
NODE cur,prev;
if(first==NULL)
{
printf("list is empty cannot delete\n");
return first;
}
if(first->link==NULL)
{
printf("item deleted is %d\n",first->info);
free(first);
return NULL;
}
prev=NULL;
cur=first;
while(cur->link!=NULL)
{
prev=cur;
cur=cur->link;
}
```

```
printf("item deleted at rear-end is %d",cur->info);
free(cur);
prev->link=NULL;
return first;
}

void display(NODE first)
{
 NODE temp;
 if(first==NULL)
 printf("list empty cannot display items\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("\n 1:Insert_front\n 2:Delete_front\n 3:Insert_rear\n 4:Delete_rear\n
5:Display_list\n6:Exit\n");

printf("ENTER THE CHOICE\n");

scanf("%d",&choice);

switch(choice)

 {

  case 1:printf("Enter the item at front-end\n");

         scanf("%d",&item);

         first=insert_front(first,item);

         break;

  case 2:first=delete_front(first);

         break;

  case 3:printf("Enter the item at rear-end\n");

         scanf("%d",&item);

         first=insert_rear(first,item);

         break;

  case 4:first=delete_rear(first);

         break;

  case 5:display(first);

         break;

 default:exit(0);

         break;
```

```
}

}


}
```

OUTPUT-

```
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
1
Enter the item at front-end
6

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
1
Enter the item at front-end
7

 1:Insert_front
 2:Delete_front
```

```
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
1
Enter the item at front-end
8

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
5
8
7
6

 1:Insert_front
```

```
6

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
2
item deleted at front-end is=8

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
2
item deleted at front-end is=7

 1:Insert_front
 2:Delete_front
```

```
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
3
Enter the item at rear-end
4

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
3
Enter the item at rear-end
7

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
```

```
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
3
Enter the item at rear-end
9

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
5
4
7
9

 1:Insert_front
 2:Delete_front
 3:Insert_rear
```

```
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
4
item deleted at rear-end is 9
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
4
item deleted at rear-end is 7
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
```

```
ENTER THE CHOICE
4
item deleted at rear-end is 7
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
4
item deleted is 4

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
6:Exit
ENTER THE CHOICE
6


...Program finished with exit code 0
Press ENTER to exit console.
```