```
#include<stdio.h>
#include<stdlib.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 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 insert_pos(int item,int pos,NODE first)
{
	NODE temp,cur,prev;
	int count;
	temp=getnode();
	temp->info=item;
	temp->link=NULL;
	if(first==NULL&&pos==1)
	{
		return temp;
	}
	if(first==NULL)
	{
		printf("invalid position\n");
		return first;
	}
	if(pos==1)
	{
		temp->link=first;
		first=temp;
		return temp;
	}
	count=1;
	prev=NULL;
	cur=first;
	while(cur!=NULL&&count!=pos)
	{
		prev=cur;
		cur=cur->link;
		count++;
	}
	if(count==pos)
	{
		prev->link=temp;
		temp->link=cur;
		return first;
	}
	printf("invalid position\n");
	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);
	}
}
int main()
{
	int item,choice,pos;
	NODE first=NULL;
	for(;;)
	{
		printf("\n 1:Insert_front\n 2:Insert_rear\n 3:Insert pos\n 4:Display_list\n 5: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:printf("enter the item at rear-end\n");
			scanf("%d",&item);
			first=insert_rear(first,item);
			break;
```

```
            case 3:printf("enter the item to be inserted at given position\n");
            scanf("%d",&item);
            printf("enter the position\n");
            scanf("%d",&pos);
            first=insert_pos(item,pos,first);
            break;
            case 4:display(first);
            break;
            default:exit(0);
            break;
        }
    }
}
```