Slip 1.	head==NULL;	newn-	
Que.1.	char ch[3];	>prev=temp;	while(temp!=
Implement a list library (doublylist.h) for a doubly linked list of integers with the create, display operations. Write a menu driven program to call these operations.	while(1){   struct node   *temp;   newn=(struct   node*)malloc   (sizeof(struct   node));	temp=temp- >next; }  printf("\nDo you want to enter more elements? - ");	<pre>NULL){   printf("%d\t",   temp-&gt;data);   temp=temp-   &gt;next;   } }</pre>
//doubly list header #include <stdi< td=""><td>printf("\nEnte r the data - "); scanf("%d",&amp;</td><td>scanf("%s",ch );</td><td>//doubly linked list #include<stdi o.h=""></stdi></td></stdi<>	printf("\nEnte r the data - "); scanf("%d",&	scanf("%s",ch );	//doubly linked list #include <stdi o.h=""></stdi>
o.h> #include <stdli b.h=""></stdli>	newn->data); newn- >next=newn-	if(strcmp(ch," n")==0) break;	#include <stdli b.h&gt; #include "1.1-</stdli 
<pre>#include<strin g.h=""> struct node{ int data;</strin></pre>	>prev=NULL;  if(head==NUL L){	} void display() {	<pre>doublylist.h" int main() { int ch;</pre>
struct node *next,*prev; }*head,*temp ,*newn; void create() {	head=temp=n ewn; } else{ temp- >next=newn;	struct node *temp; temp=head;  printf("\nDou bly linked list is - ");	while(1){  printf("\n\nD OUBLY LINEAR LINKED LIST");

	default:	}*head,*new	else{
printf("\n1.Cr	printf("\nEnte	n,*temp;	temp-
eate a linked	r correct	void create()	>next=newn;
list");	choice");	{	temp=temp-
	}	char ch[3];	>next;
printf("\n2.Di	}		}
splay the list");	}	head==NULL;	, printf("\nDo
1130 /,	Slip 1	while(1){	you want to
printf("\n3.Ex	Que.2	struct node	enter more
it");	•	*temp;	elements? ");
,,	Write a program that	newn=(struct	
printf("\nEnte	sorts the	node*)malloc	scanf("%s",ch
r the choice	elements of	(sizeof(struct node));	);
from above -	linked list using any of sorting	nodejj,	
");	technique	printf("\nEnte	if(strcmp(ch,"
	•	r the data - ");	n")==0)
scanf("%d",&	//sort linked list	,,	break;
ch);	#include <stdi< td=""><td>scanf("%d",&amp;</td><td>}</td></stdi<>	scanf("%d",&	}
switch(ch){	o.h>	newn->data);	}
case 1:	#include <stdli< td=""><td>newn-</td><td>void display()</td></stdli<>	newn-	void display()
create();	b.h>	>next=NULL;	ς
break;	#include <strin< td=""><td></td><td>l</td></strin<>		l
case 2:	g.h>	if(head==NUL	struct node
display();		L){	*temp;
break;	struct node{		temp=head;
case 3:	int data;	head=temp=n	
exit(0);	struct node	ewn;	printf("\nLink
	*next;	}	ed list is - ");

	j=i->next;		break;
while(temp!= NULL){		printf("\nOpe rations on	case 2 : sort();
<pre>printf("%d\t", temp-&gt;data); temp=temp- &gt;next; }  void sort() {   struct node *i=head,*j=N ULL; int curr;</pre>	<pre>while(j!=NULL ) {   if(i-&gt;data&gt;j-   &gt;data)   {   curr=i-&gt;data;   i-&gt;data=j-   &gt;data;   j-&gt;data=curr;   }   j=j-&gt;next; }</pre>	singly linked list - ");  printf("\n1.Cr eate a linked list");  printf("\n2.So rt the list");  printf("\n3.Di splay list");  printf("\n4.Ex it");	<pre>break; case 3 : display(); break; case 4 : exit(0); default: printf("\nEnte r correct choice - "); } }</pre>
<pre>if(head==NUL L){   return; } else { while(i!=NULL ) {</pre>	<pre>i=i-&gt;next; }  i=i-&gt;next; }  int main() {  int ch;  while(1) {</pre>	<pre>printf("\nEnte r your choice - ");  scanf("%d",&amp; ch); switch(ch) {   case 1:   create();</pre>	

Slip 2.	void create()		struct node
Que.1	{	head=temp=n	*temp;
-		<pre>head=temp=n ewn; } else { temp- &gt;next=newn; temp=newn; } printf("\nDo you want to enter more elements? - ");  scanf("%s",ch );  if(strcmp(ch," n")==0) break; } printf("\n"); }</pre>	_
}*head,*new n,*temp;		void display() {	{ int ch;
·			

1 11 (4) (	·	l	
while(1){	case 2:	preserve the order)	printf("Enter
	display();	,	the
printf("\n\nSI	break;	//evaluate	expression ::
NGLY LINEAR	case 3:	postflix	");
LINKED LIST");	exit(0);	#include <stdi< td=""><td></td></stdi<>	
	default:	o.h>	scanf("%s",ex
printf("\n1.Cr	printf("\nEnte	int stack[20];	p);
eate a linked	r correct	int top = -1;	e = exp;
list");	choice");		while(*e !=
	1	void push(int	'\0')
printf("\n2.Di	}	( x)	, ,
splay the	}	{	1
list");	}	stack[++top]	if(isdigit(*e))
	Que.2	= x;	{
printf("\n3.Ex		,	num = *e -
it");	. Write a	}	48;//num for
,,	program that copies the	int pop()	store value of
	contents of one	{	epression
printf("\nEnte	stack into	return	•
r the choice	another. Use	stack[top];	push(num);
from above -	stack library to		}
");	perform basic	}	else
	stack	int main()	ſ
scanf("%d",&	operations. The	{	l
ch);	order of two stacks		n1 =
switch(ch){		char exp[20];	pop();//if e is
case 1:	must be identical.(Hint:	char *e;	operator then
create();	Use a	int	pop the two
	temporary	n1,n2,n3,num	elemets from
break;	stack to	;//for storing	the
		pop element	

stack and	}	ascending	
sore it in var	case '/':	order by	array[j+1]=arr
n2 = pop();	1	using	ay[j];
		insertion sort	j;
switch(*e)//c	n3 = n2 / n1;	algorithm.	}
heck the	break;	// insertion	,
given	}	sort using	array[j+1]=te
operator and	}	function	mp;
perform the	nuch(n2):	#include <stdi< td=""><td>1</td></stdi<>	1
operations	push(n3);	o.h>	,
accordingly	}	int	}
{	e++;	insertion(int	int main()
case '+':	}	array[],int	{
s .	printf("\nThe	size)	int a[100],n;
l	result of	{	
n3 = n1 + n2;	expression %s	for(int	printf("\nEnte
break;	=	i=1;i <size;i++)< td=""><td>r size of array</td></size;i++)<>	r size of array
}	%d\n\n",exp,	{	- ");
case '-':	pop());	int	
1	return 0;	temp=array[i]	scanf("%d",&
	Slip 3.	;	n);
n3 = n2 - n1;	Que.1	int j=i-1;	
break;	Sort a	while(j>=0	printf("\nEnte
}	random array	&&	r %d
case '*':	of n integers	array[j]>temp	elements in
{	(accept the	)	array - ",n);
n3 = n1 * n2;	value of n	{	for(int
•	from user) in		i=0;i <n;i++)< td=""></n;i++)<>
break;			{
		I	I

	}	{	int a[n],i;
scanf("%d",&	Slip 4		
a[i]);	Que.1	temp=array[i]	printf("\nEnte
}		;	r elements in
	Read the 'n'		the array:- ");
printf("\nUns	numbers from user and	array[i]=array	
orted array is	sort using	[i+1];	for(i=0;i <n;i++< td=""></n;i++<>
- ");	bubble sort.		)
for(int		array[i+1]=te	{
i=0;i <n;i++)< td=""><td>//bubble sort</td><td>mp;</td><td></td></n;i++)<>	//bubble sort	mp;	
{	using function	}	scanf("%d",&
•	#include <stdi< td=""><td>}</td><td>a[i]);</td></stdi<>	}	a[i]);
printf("%d\t",	o.h>	) )	}
a[i]);	int bubble(int	}	,
}	array[],int	return 0;	printf("\nUns
,	size)	}	orted array
insertion(a,n);	1	int main()	is:- ");
111301110111(4,11),		{	
nrintf("\nSort	int i,j,temp;	int n	for(i=0;i <n;i++< td=""></n;i++<>
printf("\nSort ed array is -	6 (1 - 1 - 1	int n;	)
");	for(i=0;i <size;i< td=""><td>:</td><td>{</td></size;i<>	:	{
for(int	++)	printf("\nHow	·
i=0;i <n;i++)< td=""><td>{</td><td>many elements you</td><td>printf("%d\t",</td></n;i++)<>	{	many elements you	printf("%d\t",
· •,· ·,· ,	for(j=0;j <size-< td=""><td>want to</td><td>a[i]);</td></size-<>	want to	a[i]);
l	i-1;j++)	enter? - ");	}
~~:~+f(!!0/ d\ +!!	{		) 
printf("%d\t", a[i]);		scanf("%d",&	bubble(a,n);
α[i] <i>]</i> ,	if(array[i]>arr	n);	
}	ay[i+1])		printf("\nSort
	ı		I

ed array is:- ");	<pre>int linear(int array[],int size,int key)</pre>	scanf("%d",& key);	implement the below two operations.  1) Add an
<pre>for(i=0;i<n;i++ )="" pre="" printf("%d\t",<="" {=""></n;i++></pre>	{ //int flag; for(int i=0;i <size;i++)< td=""><td><pre>int flag=linear(a, 5,key); if(flag) {</pre></td><td>element with its priority into the queue.  2) Delete an element from queue according to its priorit</td></size;i++)<>	<pre>int flag=linear(a, 5,key); if(flag) {</pre>	element with its priority into the queue.  2) Delete an element from queue according to its priorit
a[i]); }	if(array[i]==ke	printf("\n key is found");	/*  * C Program
Slip 5 Que.1	y) {	} else	to Implement Priority
Create a random array of n integers. Accept a value x from user and use linear search algorithm to check whether the number is present in the array or not and output the position if the number is present.	return 1; }  return 0; }  int main() {  int a[5]={23,56,7 8,93,90};	<pre>{   printf("\n key is not found"); }  Slip 5  Slip 5  Que.2 . Implement a priority rucus</pre>	Queue to Add and Delete Elements  */ #include <stdio.h> #include <stdlib.h> #define MAX 5  void</stdlib.h></stdio.h>
// linear search using functon #include <stdi o.h=""></stdi>	int key;  printf("\n  enter key to  the search:- ");	priority queue library (PriorityQ.h) of integers using a static implementation of the queue and	insert_by_pri ority(int);  void delete_by_pri ority(int);

<pre>void create(); void check(int); void display_pque ue(); int pri_que[MAX] ; int front, rear; void main() {   int n, ch;   printf("\n1 -   Insert an   element into   queue");   printf("\n2 -   Delete an   element from   queue");   printf("\n3 -   Display queue   elements");   printf("\n4 -   Exit");   create();</pre>	<pre>printf("\nEnte r your choice : "); scanf("%d", &amp;ch); switch (ch) {   case 1:    printf("\nEnte   r value to be   inserted : ");    scanf("%d",&amp;   n);    insert_by_pri   ority(n);   break;   case 2:    printf("\nEnte   r value to   delete : ");</pre>	scanf("%d",& n);  delete_by_pri ority(n); break; case 3:  display_pque ue(); break; case 4: exit(0); default:  printf("\nChoi ce is incorrect, Enter a correct choice"); } } /* Function to create an	<pre>priority queue */ void create() {   front = rear =   -1; }   /* Function to   insert value   into priority   queue */   void   insert_by_pri   ority(int data)   {   if (rear &gt;=     MAX - 1)     {     printf("\nQue     ue overflow     no more     elements can     be inserted");     return; }</pre>
while (1)		empty	

<pre>if ((front == - 1) &amp;&amp; (rear == -1)) {   front++;   rear++;   pri_que[rear] = data;   return; }   else   check(data);   rear++; } /* Function to   check priority   and place   element */   void check(int   data) {   int i,j;   for (i = 0; i &lt;= rear; i++)   {</pre>	<pre>if (data &gt;=   pri_que[i])  {   for (j = rear +   1; j &gt; i; j)   {     pri_que[j] =     pri_que[j - 1];   }   pri_que[i] =   data;   return;   }   }   pri_que[i] =   data;   rounction to   delete an   element from   queue */   void   delete_by_pri   ority(int data)   {   int i;   } </pre>	<pre>if ((front==-1)     &amp;&amp; (rear==- 1))     {         printf("\nQue         ue is empty         no elements         to delete");         return;         }         for (i = 0; i &lt;=             rear; i++)         {             if (data ==                   pri_que[i])             {                   pri_que[i] =</pre>	<pre>front = -1; return; }  printf("\n%d not found in queue to delete", data); }  /* Function to display queue elements */ void display_pque ue()  {   if ((front == - 1) &amp;&amp; (rear == -1))   {    printf("\nQue ue is empty");   return; }</pre>
	int i;		
		if (rear == -1)	

<pre>for(i=0;i<n;i++ )="" ++)="" 1;i++)="" \t",a[i]);="" begins="" for(i="0;i&lt;n-" for(j="i+1;j&lt;n;j" if(a[position]="" int="" j,position,swa="" p;="" position="i;" printf("%d="" printf("\n");="" selection="" sort="" {="" }=""> a[j])       {         position = j;       }     } }</n;i++></pre>	<pre>swap = a[i]; a[i] = a[position]; a[position] = swap; }  printf("sorted array is : \n"); //selection sorted array is  for(i=0;i<n;i++ (dyqueue.h)="" )="" 6="" \t",a[i]);="" a="" dynamic<="" end="" implement="" integers="" library="" of="" pre="" printf("%d="" que.2="" queue="" slip="" sorting="" using="" {="" }=""></n;i++></pre>
<pre>printf("enter size of array: \n");  scanf("%d",&amp; n);   //create array   int a[n],i;   printf("enter %d elements   in array:   \n",n);  for(i=0;i<n;i++ )="" <="" \n");="" a[i]);="" array="" display="" ed="" is:="" pre="" printf("unsort="" scanf("%d",&="" {="" }=""></n;i++></pre>	size of array :
	<pre>     printf("%d     \t",a[i]);     }     printf("\n");     //selection     sort begins     int     j,position,swa     p;     for(i=0;i<n- ++)="" 1;i++)="" for(j="i+1;j&lt;n;j" if(a[position]<="" position="i;" td="" {=""></n-></pre>

(linked list) implementatio n of the queue and implement init, enqueue, dequeue, isempty, peek operations.  //dynamic insertion of queue #include <stdi o.h="">  #include"6-2- dyqueue.h" int main() {   int n,ch;   //clrscr();   initqueue();   do   {    printf("\n\n1.   ADD\n2.DELE   TE\n3.DISPLA   Y\n4.PEEK\n5   .EXIT");</stdi>	<pre>printf("\nEnte r your choice - ");  scanf("%d",&amp; ch); switch(ch) {   case 1:   printf("\nEnte   r the data - ");  scanf("%d",&amp;   n);   enqueue(n);   break;   case 2:   dequeue();   break;   case 3:   display();   break;   case 4:   peek();   break;</pre>	<pre>case 5 :   exit(0); }</pre>	<pre>void initqueue() {     q.front=0;     q.rear=-1; }     int     enqueue(int     num)     {         if(q.rear!=MA         X-1)         {             ++q.rear;             q.data[q.rear]             =num;             printf("%d is             added at %d             position",nu             m,q.rear);         }         else          printf("\nQue</pre>
--	---	----------------------------------	--

<pre> } int dequeue() {   if(q.front==-1   &amp;&amp; q.rear==- 1)</pre>	<pre>printf("\nDele ted data is %d",x); } yoid display()</pre>	<pre>&amp;&amp; q.rear==- 1)  printf("\nQue ue is empty!!!"); }</pre>	int partition(int *a,int lb,int ub) //function defination { //Entered in
printf("\nQue ue is empty!!!");	{ int i;	void peek() {	int up,down,pivo
else if(q.front>q.re ar){	printf("\n Front -> %d",q.front);	<pre>printf("\nPee k element is %d",q.data[q. front]);}</pre>	t,swap; down = lb; up = ub;
q.rear=q.front =-1;	for(i=q.front;i <=q.rear;i++)	Slip 7. Que.1	pivot = a[lb]; do{
<pre>printf("\nQue ue is empty!!!"); } else if(q.rear!=-1){ int x=q.data[q.fr ont]; q.front++;</pre>	<pre>printf("\nDat a -&gt; %d\t",q.data[i ]); printf("\n Rear -&gt; %d",q.rear);  if(q.front&gt;q.re ar    q.front==-1</pre>	Sort a random array of n integers (accept the value of n from user) in ascending order by using quick sort algorithm.  //quick sort #include <stdio.h></stdio.h>	while((a[dow n]<=pivot) && (down < ub)) {    down ++;   }   while(a[up]>p ivot && up> lb)

{	int j;	//create	printf("%d
up;	if(lb < ub)	array	\t",a[i]);
}	{	printf("enter	}
	//calling	%d elements	printf("\n");
if(down < up)	function	in array \n",n);	}
{	j =		Slip 7
swap =	partition(a,lb,	for(i=0;i <n;i++< td=""><td>Que.2</td></n;i++<>	Que.2
a[down];	ub);	)	. Write a
a[down] =		{	program that
a[up];	quicksort(a,lb ,j-1);		checks whether a string of
a[up] = swap;	,, ±/,	scanf("%d",&	characters is
}	quicksort(a,j+	a[i]);	palindrome
}while(down	1, ub);	}	or not. The
< up);	}	//calling	function should use a stack
a[lb] = a[up];	}	function	library
a[up] = pivot;	int main()		(cststack.h) of stack of
return up;	<i>s</i>	quicksort(a,0, n-1);	characters
i	int n		using a static
}	int n;		implementatio
void quicksort(int	printf("enter the size of	//sorted array	n of the stack
*a,int lb,int	array : \n");	printf("Sortd	//palindrome( static stack
ub) //function		array is : \n");	implimentatio
defination	scanf("%d",&		n)
{	n);	for(i=0;i <n;i++< td=""><td>#include<stdi< td=""></stdi<></td></n;i++<>	#include <stdi< td=""></stdi<>
//Entered in	int a[n],i;	)	o.h>
quick sort		{	
		I	

#include <stdli b.h&gt;</stdli 	} for (i = 0; i	while(1) {	case 1:init(); break;
#include "7-2- palindrome.h "  void palindrome()	<pre><len; (string[i]="=" count++;<="" i++)="" if="" pop())="" pre="" {=""></len;></pre>	printf("\nCHE CK PALINDROME ");	case 2: palindrome(); break; case 3:isempty(); break;
char string[15];	}	printf("\n1.Ini t\n");	case 4:exit(0);
int i,count=0,len;	if (count == len)	printf("2.Chec k Palindrome	} }
printf("\nEnte r a string:");	<pre>printf("%s is a Palindrome string\n", string);</pre>	String\n"); printf("3.IsEm	} //palindrome #include <stdi< td=""></stdi<>
scanf("%s",str	else	pty\n");	o.h>
ing);	printf("%s is not a	printf("4.Exit\	#include <strin g.h&gt;</strin 
len=strlen(stri ng);	<pre>palindrome string\n", string);</pre>	n"); printf("Enter your choice:	#define MAX 15 char
for(i=0;i <len;i ++)</len;i 	} int main()	");	name[MAX],t op;
{	{	scanf("%d",& ch);	void init()
push(string[i]) ;	int ch,i; char string[15];	switch(ch) {	{ top=-1;

<pre>printf("\nStati c Stack Initiaized\n"); } void push(char c) {   top++;   name[top]=c; } char pop() {   return   name[top]; } isempty() {   if(top==-1)</pre>	printf("\nStati c Stack is not empty\n"); } Slip 8. Que.1 . Implement a list library (singlylist.h) for a singly linked list of integer With the operations create, delete specific element and display. Write a menu driven program to call these operation	<pre>int data; struct node *next; }*head,*new n,*temp; void create() {   char ch[3];   head==NULL;    while(1)   {   struct node   *temp;   newn=(struct   node*)malloc   (sizeof(struct   node));</pre>	<pre>if(head==NUL L) {   head=temp=n   ewn; }   else   {   temp-   &gt;next=newn;   temp=newn; }   printf("\nDo   you want to   enter more   elements? -   ");</pre>
{	//singly list #include <stdi o.h&gt;</stdi 	printf("\nEnte r the data - ");	scanf("%s",ch );
<pre>printf("\nStati c Stack is empty\n"); } else</pre>	#include <strin g.h&gt; #include<stdli b.h&gt;</stdli </strin 	scanf("%d",& newn->data); newn- >next=NULL;	if(strcmp(ch," n")==0) break; }
	struct node{		

printf("\n");	while((i <pos-< td=""><td>temp=head;</td><td>while(1)</td></pos-<>	temp=head;	while(1)
}	1)&&(temp-		{
void delete()	>next!=NULL)	printf("\nEle	
{	{	ments of list are - ");	printf("\nOpe
temp=head;	i++;	,,	rations on singly linked
int pos,i;	temp=temp-	while(temp!=	list - ");
struct node	>next;	NULL)	
*curr;	}	{	printf("\n1.Cr eate a linked
printf("\nEnte	if(temp-	printf("%d\t",	list");
r the position	>next==NULL)	temp->data);	
to delete - ");	printf("\nNod	temp=temp-	printf("\n2.De lete element
scanf("%d",&	es not	>next;	from the
pos);	present");	}	list");
if(pos==1)	else	printf("\n");	
{	{	}	printf("\n3.Di splay
head=head-	curr=temp- >next;	//singly list	elements of
>next;	temp-	#include <stdi o.h&gt;</stdi 	list");
free(temp);	>next=curr-	#include <stdli< td=""><td></td></stdli<>	
}	>next;	b.h>	printf("\n4Exi t");
else	free(curr);	#include"7-1-	<i>C J</i> ,
{	}	singly.h"	printf("\nEnte
i=1;	}	int main()	r your choice
temp=head;	void display()	{	- ");
}	{	int ch;	
		I	I

<pre>printf("%c",x); } else {  while(priority (stack[top])&gt;= priority(*e))  printf("%c",p op()); push(*e); } e++; } while(top!=- 1) {  printf("%c ",pop()); } return 0;</pre>	Read the data from the 'employee.txt' file and sort on age using Counting sort or Quick sort and write the sorted data to another file 'sortedemponage. txt'  //read the data from emp_quick_a ge.txt and sort bt age #include <stdi o.h="">  #include<strin g.h="">  typedef struct {     char name[30];     int age;     int salary;</strin></stdi>	RECORD emp[100]; int readfile(RECO RD[]); void writefile(REC ORD[],int); int quicksort(REC ORD *emp,int low,int high); int partition(REC ORD *emp, int low, int high); void sort(RECORD *emp,int n); int main() { int n;	<pre>writefile(emp, n); } int readfile(RECO RD *a) {   int i=0;   FILE *fp;   if   ((fp=fopen("e   mp_quick_ag   e.txt","r"))!=N   ULL)   while(!   feof(fp))   {   fscanf(fp,"%s   %d%d",a[i].na   me,&amp;a[i].age,   &amp;a[i].salary);   i++; }</pre>
} Slip 9 Que.2	RECORD;	p); sort(emp,n);	} return i; }

<pre>void writefile(REC ORD *a,int n) {   int i=0;   FILE * fp;  if((fp=fopen("   sorted_emp_   quick_age.txt ","w"))!=NULL )  for(i=0;i<n;i++ )="" );="" *emp,int="" \t%d\t%d\n",="" a[i].name,a[i].="" age,a[i].salary="" fprintf(fp,"%s="" high)="" int="" low,int="" ord="" pre="" quicksort(rec="" {<="" }=""></n;i++></pre>	<pre>{   int p;   p =   partition(emp   ,low, high);    quicksort(em   p, low, p-1);    quicksort(em   p, p+1, high);   }   void   sort(RECORD   *emp, int n)   {     quicksort(em   p,0,n-1);   }   int   partition(REC   ORD *emp,   int low, int   high)</pre>	RECORD pivot = emp[low]; int start, end; start = low; end = high; while( start < end ) {   while( start <= high && emp[start].ag e <= pivot.age )   start++;   while( emp[end].age > pivot.age )   end;   if( start < end )   {     RECORD     swap =     emp[start];	<pre>emp[end] = swap; } emp[low] = emp[end]; emp[end] = pivot; return end; } //make a file emp_quick_a ge.txt and list some name age and salary of emp //wiev the orderd age wise result in sorted_ emp_quick_a ge.txt file</pre>
{ if( low < high )	high) {	emp[start], emp[start] = emp[end];	

Slip.10	int salary;	int	int i=0;
Que.2	}	readfile(RECO RD *a)	FILE * fp;
Read the data from the file "employee.tx t" and sort on names in alphabetical order (use strcmp) using bubble sort or selection sort //read the data from txt file and sort the names using bubble sort #include <stdi o.h=""> #include<strin g.h=""> typedef struct { char name[30]; int age;</strin></stdi>	RECORD; RECORD emp[100]; int readfile(RECO RD[]); void writefile(REC ORD[],int); int sort(RECORD *emp,int n); int main() {   int n;  n=readfile(em p);   sort(emp,n);  writefile(emp, n); }	<pre>fru fa) {   int i=0;   FILE *fp;   if   ((fp=fopen("1     O-     2emp_bubble     .txt","r"))!=N   ULL)   while(!   feof(fp))   {   fscanf(fp,"%s   %d%d",a[i].na   me,&amp;a[i].age,   &amp;a[i].salary);   i++;   }   return i; }   void   writefile(REC   ORD *a,int n)   { </pre>	<pre>if((fp=fopen(" 10- 2sortedemp_ bubble.txt"," w"))!=NULL)  for(i=0;i<n;i++ )="" );="" *a="*b;" *a,="" *b="tmp;&lt;/pre" *b)="" \t%d\t%d\n",="" a[i].name,a[i].="" age,a[i].salary="" d="" fprintf(fp,"%s="" record="" swap(recor="" tmp="*a;" void="" {="" }=""></n;i++></pre>

}	//make text	printf("\n2 -	
	file name as	Delete an	insert_by_pri
int	10-	element from	ority(n);
sort(RECORD	2emp_bubble	queue");	break;
* emp,int n)	.txt save it in vs code	printf("\n3 -	case 2:
{		Display queue	
intii naas	//and output	elements");	printf("\nEnte
int i,j,pass;	is displayed in the 10-	printf("\n4 -	r value to
5 / 4	2sortedemp	Exit");	delete : ");
for(pass=1;pa	bubble.txt file	create();	
ss <n;pass++)< td=""><td>Slip.11</td><td>while (1)</td><td>scanf("%d",&amp;</td></n;pass++)<>	Slip.11	while (1)	scanf("%d",&
{	•	{	n);
for(i=0;i <n-< td=""><td>Que.2</td><td></td><td></td></n-<>	Que.2		
pass;i++)	//priority	printf("\nEnte	delete_by_pri
{	queue	r your choice :	ority(n);
if (strcmp	#include	");	break;
(emp[i].name,	<stdio.h></stdio.h>	scanf("%d",	case 3:
emp[i+1].nam	#include	&ch);	
e)>0)	<stdlib.h></stdlib.h>	switch (ch)	display_pque
{	#include "11-	, ,	ue();
	2priorityq.h"	{	break;
swap(&emp[i]	void main()	case 1:	·
,&emp[i+1]);	{		case 4:
}	int n, ch;	printf("\nEnte	exit(0);
}		r value to be	default:
}	printf("\n1 - Insert an	inserted : ");	
J	element into		printf("\nChoi
}	queue");	scanf("%d",&	ce is
		n);	incorrect,
		'	

Enter a	}	return;	pri_que[i] =
correct	/* Function to	}	data;
choice");	insert value	else	return;
}	into priority		}
}	queue */	check(data);	,
,	void	rear++;	}
}	insert_by_pri	}	pri_que[i] =
//priority	ority(int data)	/* Function to	data;
queue	{	check priority	}
#include	:£ /	and place	/* Function to
<stdio.h></stdio.h>	if (rear >= MAX - 1)	element */	delete an
#include	iviAX - 1)	void check(int	element from
<stdlib.h></stdlib.h>	{	data)	queue */
#define MAX		r	void
5	printf("\nQue	1	delete_by_pri
int	ue overflow	int i,j;	ority(int data)
pri_que[MAX]	no more	for (i = 0; i <=	{
;	elements can	rear; i++)	int it
int front, rear;	be inserted");	{	int i;
	return;	if (data >=	if ((front==-1)
/* Function to	}	pri_que[i])	&& (rear==-
create an	if ((front == -	(qo.e[.])	1))
empty priority	1) && (rear	1	{
queue */	== -1))	for (j = rear +	
•	{	1; j > i; j)	printf("\nQue
void create()	frontili	{	ue is empty
{	front++;	pri_que[j] =	no elements
front = rear =	rear++;	pri_que[j - 1];	to delete");
-1;	pri_que[rear]	}	return;
	= data;	,	

}	}	Slip.12	
for (i = 0; i <=	/* Function to	Que.1	fp=fopen("12-
<pre>rear; i++) {   if (data ==   pri_que[i])   {   for (; i &lt; rear;   i++)    {     pri_que[i] =     pri_que[i + 1]; }</pre>	display queue elements */  void display_pque ue() {  if ((front == - 1) && (rear == -1)) {	//read data from 12-2- cities.txt file and apply linear search to find cities  #include <stdi o.h="">  #include<strin g.h="">  typedef struct</strin></stdi>	cities.txt","r") ; if(fp==NULL) printf("File Not Exist"); else { while(!feof(fp ))
} pri_que[i] = -	printf("\nQue ue is empty");	city {	{
99;	return;	char	fscanf(fp,"%s
rear;	}	name[20];	%d",
if (rear == -1)	for (; front <=	int code;	a[i].name, &a[i].code);
front = -1;	rear; front++)	} city;	i++;
return;	{	//Fileread	}
} }	printf(" %d ", pri_que[front ]);	int readfile(city a[20])	fclose(fp); }
printf("\n%d not found in	}	{	return i;
queue to	front = 0;	FILE *fp;	}
delete",	}	int i=0;	int main()
data);			{

int i, n;	int	Slip.13	(sizeof(struct
char key[20];	index,flag=0;	Que.2	node));
city a[20];	<pre>printf("Enter city:");</pre>	//sort list	printf("\nEnte
n = fileread(a);	City. J,	using bubble sort	r the data - ");
for(int i=0; i <n; i++)<="" td=""><td>scanf("%s",str );</td><td>#include<stdi o.h&gt;</stdi </td><td>scanf("%d",&amp; newn-&gt;data);</td></n;>	scanf("%s",str );	#include <stdi o.h&gt;</stdi 	scanf("%d",& newn->data);
printf("%s %d\n",	for(int i=0;i <n;i++)< td=""><td>#include<stdli b.h&gt;</stdli </td><td>newn-</td></n;i++)<>	#include <stdli b.h&gt;</stdli 	newn-
a[i].name, a[i].code);	{	#include <strin g.h&gt;</strin 	>next=NULL;
/* displaying	if(strcmp(str,a	struct node{	if(head==NUL L){
records*/	[i].name)==0)	int data;	,,
linearsearch(	flag=1;	struct node *next;	head=temp=n ewn;
n); เ	index=i;	}*head,*new	}
//Linear	}	n,*temp;	else{
Search	}	void create()	temp-
int	if(flag==1)	{	>next=newn;
linearsearch(i	printf("City Code:	char ch[3];	temp=newn;
nt n)	%d",a[index].	head=NULL;	} printf("\ nDo
\ =:t===[20]-	code);	while(1){	printf("\nDo you want to
city a[20];	else	struct node *temp;	enter more
n=fileread(a);	printf("City	newn=(struct	elements? ");
char str[20];	Not in list");	node*)malloc	
	}		

C/IIO/ II I	{	GLY LINKED	break;
scanf("%s",ch );	struct node	LIST");	case 3:
1,	*i,*j;	nrin+f("\n1 Cr	display();
if(strcmp(ch,"	int curr;	printf("\n1.Cr eate the list");	break;
n")==0)	for(i=head;i-	,,,	case 4:
break;	>next!=NULL;i =i->next){	printf("\n2.So	exit(0);
}	,,	rt the list");	default:
}	for(j=i- >next;j!=NULL		printf("\nEnte r correct
void display()	;j=j->next){	printf("\n3.Di	choice");
s	if(i->data>j-	splay the list");	}
struct node	>data){	1130 /,	}
struct node *temp;	curr=i->data;	printf("\n4.Ex	}
temp=head;	i->data=j-	it");	•
printf("\nList	>data;		
is - ");	j->data=curr;	printf("\nEnte	
	}	r the choice from the	
while(temp!=	}	above - ");	
NULL){	}	,,	
	}	scanf("%d",&	
<pre>printf("%d\t", temp-&gt;data);</pre>	int main()	ch);	
	{	switch(ch){	
temp=temp- >next;	int ch;	case 1:	
}	while(1){	create();	
, }	willic(±/[	break;	
J	printf("\nSIN	case 2 :	
void sort()	, , ,	sort();	

Slip.14		printf("\n");	Slip.15
Que.1	printf("\nEnte		Que.1
//linear search	r the size of array - ");	printf("\nEnte r key to the	//selection sort
#include <stdi o.h&gt;</stdi 	scanf("%d",& n);	search - "); scanf("%d",&	#include <stdi o.h&gt;</stdi 
int linear(int	a[n];	key);	int main()
array[], int size, int key)	printf("Enter	int	{
{	elements in	found=linear(	int n;
int i;	array - ");	a,n,key);	<pre>printf("enter size of array :</pre>
for(i=0;i <size;i< td=""><td>for(i=0;i<n;i++ ){</n;i++ </td><td>if(key==found ){</td><td>\n");</td></size;i<>	for(i=0;i <n;i++ ){</n;i++ 	if(key==found ){	\n");
++){  if(key==array[ i]){	scanf("%d",& a[i]); }	printf("\nkey is found at %d location",i);	scanf("%d",& n); //create array
return 1;	,	else{	int a[n],i;
} } return 0;	printf("\nGive n array is - ");	printf("\nKey is not found!!!");	printf("enter %d elements in array:
}	for(i=0;i <n;i++ ){</n;i++ 	}	\n",n);
int main()			for(i=0;i <n;i++< td=""></n;i++<>
{ int	printf("%d\t", a[i]);		) {
a[100],n,i,key;	}		•
	·	·	

		printf("%d	int main()
scanf("%d",&	for(j=i+1;j <n;j< td=""><td>\t",a[i]);</td><td>{</td></n;j<>	\t",a[i]);	{
a[i]);	++)	}	int ch;
}	{	//sorting end	·
	if(a[position]	1	while(1)
printf("Unsort	> a[j])	}	{
ed array is :	<b>!</b>	Slip.17	
\n");		Que.1	printf("\nOpe
//display	position = j;	//singly	rations on
array	}	reverse	singly linked
	}	#include <stdi< td=""><td>list - ");</td></stdi<>	list - ");
for(i=0;i <n;i++< td=""><td>swap = a[i];</td><td>o.h&gt;</td><td></td></n;i++<>	swap = a[i];	o.h>	
)		#include <stdli< td=""><td>printf("\n1.Cr</td></stdli<>	printf("\n1.Cr
ĺ	a[i] = a[position];	b.h>	eate a linked
l			list");
printf("%d	a[position] =	#include <strin< td=""><td></td></strin<>	
\t",a[i]);	swap;	g.h>	printf("\n2.Di
}	}	struct node	splay
printf("\n");		{	elements of
//selection	printf("sorted	int data;	list");
sort begins	array is : \n");		
int	//selection	struct node *next;	printf("\n3.Re
j,position,swa	sorted array	,	verse a list");
p;	is	}*head,*new	
		n,*temp;	printf("\n4.Ex
for(i=0;i <n- 1;i++)</n- 	for(i=0;i <n;i++< td=""><td><pre>void create();</pre></td><td>it");</td></n;i++<>	<pre>void create();</pre>	it");
±,ı · ⊤)	)	void display();	
{	[	void	printf("\nEnte
position = i;		reverse();	r your choice
			- ");

	head==NIIII:	t	
<pre>scanf("%d",&amp; ch); switch(ch) {   case 1 :   create();   break;   case 2 :   display();</pre>	head==NULL;  while(1) {   struct node   *temp;   newn=(struct   node*)malloc   (sizeof(struct   node));	temp- >next=newn; temp=newn; } printf("\nDo you want to enter more elements? - ");	<pre>while(temp!= NULL) {  printf("%d\t", temp-&gt;data); temp=temp- &gt;next; }</pre>
break; case 3: reverse();	printf("\nEnte r the data - ");	scanf("%s",ch );	<pre>printf("\n"); } void reverse()</pre>
<pre>break; case 4 : exit(0); default: printf("\nEnte r correct choice - "); } }</pre>	scanf("%d",& newn->data); newn- >next=NULL;  if(head==NUL L) {	<pre>if(strcmp(ch," n")==0) break; } printf("\n"); } void display() {</pre>	<pre>struct node *t1,*t2; t1=t2=NULL;  while(head!= NULL) { t2=head-</pre>
<pre> } void create() {  char ch[3];</pre>	head=temp=n ewn; } else	temp=head;  printf("\nEle ments of list are - ");	>next; head- >next=t1; t1=head; head=t2;

}	printf("Enter	",peek(temp))	int peek(stack
head=t1;	%d	;	a)
}	elements\n", n);	for(int	{
Slip 17	, ,	i=0;i <n;i++)< td=""><td>return</td></n;i++)<>	return
Que.2	for(int i=0;i <n;i++){< td=""><td></td><td>a.st[a.top];</td></n;i++){<>		a.st[a.top];
	1-0,1 (11,11 1)[	push(&c_or,p	}
//copy stack	scanf("%d",&	op(&temp));/ /to copy temp	void init(stack
#include <stdli b.h&gt;</stdli 	x);	stack into	* a)
#include"2-2-	//accpeting	c_stack;	{
copystack.h"	orignal		a->top=-1;
void main()	stack	printf("\npee	}
r	nuch/8 orginal	k element of	void
1	push(&orginal,x);	temp stack is %d	push(stack *
stack orginal,temp,	\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \	",peek(c_or));	a,int x)
c or;	forlint	}	{
int n,x;	for(int i=0;i <n;i++)< td=""><td>//copy stack</td><td>if(a-</td></n;i++)<>	//copy stack	if(a-
	, com (1.1)	#include <stdi< td=""><td>&gt;top==MAX-</td></stdi<>	>top==MAX-
init(&orginal);	push(&temp,	o.h>	1)
init(&temp);	pop(&orginal)	#define MAX	printf("\nstac k is full ");
init(&c_or);	);	20	else{
· <del>-</del> ·	orignal stack	typedef struct	
printf("Enter no of	into temp stack	{	a->st[++a- >top]=x;
nodes\n");	Stack		ι
, ,,	printf("\npee	int top;	}
scanf("%d",&	k element of	int st[MAX];	}
n);	temp stack is	}stack;	int pop(stack
	%d		*a)

```
{
                     printf("enter
                                                               //selection
                                          int
                     %d elements
                                         i,position,swa
                                                              sorted array
if(a-
                     in array:
                                         p;
                                                              is
>top>=MAX-
                     \n",n);
1) return 0;
                                          for(i=0;i<n-
                                          1;i++)
                                                              for(i=0;i<n;i++
else
                    for(i=0;i<n;i++
                                          {
return a-
                                          position = i;
>st[a->top--];
                     {
                                                               printf("%d
                                                              \t",a[i]);
                                         for(j=i+1;j<n;j
Slip.19
                     scanf("%d",&
                                         ++)
                     a[i]);
Que.1
                                                               //sorting end
//selection
                     }
                                          if(a[position]
                                                              }
sort
                                         > a[j]
#include<stdi
                     printf("Unsort
                                          {
                     ed array is:
o.h>
                     \n");
                                          position = j;
int main()
                     //display
                                          }
{
                     array
int n;
                                          swap = a[i];
printf("enter
                    for(i=0;i<n;i++
size of array:
                                          a[i] =
                                          a[position];
\n");
                                          a[position] =
                     printf("%d
scanf("%d",&
                                          swap;
                     \t",a[i]);
n);
                                          }
//create
                     printf("\n");
array
                                          printf("sorted
int a[n],i;
                     //selection
                                          array is : \n");
                     sort begins
```

Slip.20		case 4:	
Que.1	printf("\n3.Di	printf("\nTop	printf("\nEnte
//Array as a	splay");	most element	r the data to
stack(static)	2.00	of stack is %d",peek());	push - ");
#include <stdi< td=""><td>printf("\n4.Pe</td><td></td><td>f/!!0/ d!! Q</td></stdi<>	printf("\n4.Pe		f/!!0/ d!! Q
o.h>	ek");	break; _	scanf("%d",& data);
#include <stdli< td=""><td>~~:~+f/"\ ~</td><td>case 5:</td><td>•</td></stdli<>	~~:~+f/"\ ~	case 5:	•
b.h>	printf("\n5.Ex it");	exit(0);	if(isfull())
#include "20-	ις γ,	default:	{
1static_stack.	printf("\nEnte	printf("\nEnte r the correct	
h"	r your choice :	choice : ");	printf("\nStac
int main()	");	}	k Overflow");
{		1	}
int ch;	scanf("%d",&	,	top+=1;
,	ch);	}	
while(1)		#include <stdi< td=""><td>stack[top]=da</td></stdi<>	stack[top]=da
{	switch(ch)	o.h>	ta;
	<i>.</i>	#include <stdli< td=""><td>}</td></stdli<>	}
printf("\nEnte r a number to	case 1 :	b.h>	int pop()
perform	push();	#define MAX	{
operations on	break;	3	int p;
stack : ");	ŕ	int top=-	if(isempty())
	case 2 :	1,stack[MAX];	s
printf("\n1.Pu	pop();	void push()	١
sh");	break;	{	printf("\nStac
	case 3:	int data;	k
printf("\n2.Po	display();		 Underflow");
p");	break;		

1		l	
}	printf("\nStac	return 1;	printf("\n stack is full
	k	else	\n");
p=stack[top];	Underflow");	return 0;	
top=top-1;	else	Slip.21	return;
	(	Que.1	}
printf("\nPop	1	//reverse	top+=1;
ed element is	f/: O: 1 h	string	
%d",p);	for(i=0;i<=top ;i++)	#include <stdi< td=""><td>stack_string[t</td></stdi<>	stack_string[t
}	,177)	o.h>	op]=item;
int peek(){	nrintf("ctack[	#include <strin< td=""><td>}</td></strin<>	}
	printf("stack[ %d]=%d\n",i,s	g.h>	char
if(ton== 1)	tack[i]);	#define MAX	popchar()
if(top==-1)	}	100	{
{	1	//maximum	if(isempty())
	}	no of	{
printf("\nStac	int isfull()	character	n nin+f/!!\ n
k Underflow");	{	int top=-1;	printf("\n stack is empty
,	if(top==MAX-	int item;	\n");
exit(1);	1)	char	return 0;
}	return 1;	stack_string[	return o,
return	else	MAX];	}
stack[top];	return 0;	void	item =
}	1	pushchar(cha	stack_string[t
void display()	· · · · · · · · · · · · · · · · · · ·	r item)	op];
{	int isempty()	{	top-=1;
int i;	{	if(isfull())	return item;
if(top==-1)	if(top==-1)	{	}
Π(top== 1)		(	

int isempty()	int main()	return 0;	void
{	{	}	writefile(REC
if(top==-1)	char	Slip 21	ORD[],int);
return 1;	str[MAX];	Que.2	int sort(RECORD
else	int i;	//read data	*emp,int n);
return 0;	printf("input	from 21-2-	int main()
}	a string :- ");	emp_insertio	{
int isfull()	<del>-</del> - (     0 /   0 ) - 1	n_name.txt file and sort	int n;
{	scanf("%[^\n] s",str);	the names	,
if(top==MAX-	string with	#include <stdi< td=""><td>n=readfile(em</td></stdi<>	n=readfile(em
1)	spaces	o.h>	p);
return 1;	5 (1 2 1 1 1	#include <strin< td=""><td>sort(emp,n);</td></strin<>	sort(emp,n);
else	for(i=0;i <strle n(str);i++)</strle 	g.h>	
return 0;	11(301),111)	typedef struct	writefile(emp, n);
}	pushchar(str[i	{	11 <i>),</i> 1
//reverse	1);	char	; int
string using		name[50];	int readfile(RECO
array(static	for(i=0;i <strle< td=""><td>int age;</td><td>RD *a)</td></strle<>	int age;	RD *a)
implimentatio n)	n(str);i++)	int salary;	{
#include <stdi< td=""><td>str[i]=popchar</td><td>}</td><td>int i=0;</td></stdi<>	str[i]=popchar	}	int i=0;
o.h>	();	RECORD;	FILE *fp;
#include "21-		RECORD emp[200];	if
1cstacks.h"	printf("revers	int	((fp=fopen("2
#define MAX	ed string is :- %s \n",str);	readfile(RECO	1-2-
100	/03 \II ,3U J,	RD[]);	emp_insertio
	'		

n_name.txt","			Slip.23
r"))!=NULL)	for(i=0;i <n;i++< td=""><td>for(i=1;i<n;i++< td=""><td>Que.1</td></n;i++<></td></n;i++<>	for(i=1;i <n;i++< td=""><td>Que.1</td></n;i++<>	Que.1
while(! feof(fp)) {	fprintf(fp,"%s	<b>)</b> {	//priority queue
<pre>fscanf(fp,"%s %d%d",a[i].na me,&amp;a[i].age, &amp;a[i].salary); i++; } return i; } void writefile(REC ORD *a,int n) { int i=0; FILE * fp;  if((fp=fopen(" 21-2- sorted_emp_i nsertion_nam</pre>	<pre>fprintf(fp,"%s \t%d\t%d\n", a[i].name,a[i]. age,a[i].salary ); }  void swap(RECOR D *a, RECORD *b) {    RECORD tmp=*a;    *a=*b;    *b=tmp; }    int    sort(RECORD    *emp,int n) {    RECORD temp;</pre>	<pre>temp=emp[i];   for(j=i-1;j&gt;=0     &amp;&amp;     strcmp(emp[j   ].name,temp.     name)&gt;0;j)     {     emp[j+1]=em     p[j];     }     emp[j+1]=te     mp;     } }</pre>	#include <stdio.h> #include <stdlib.h> #include "23- 1-priorityq.h"  void main() {   int n, ch;   printf("\n1 -   lnsert an   element into   queue");   printf("\n3 -   Display queue   elements");   printf("\n4 -   Exit");   create();   while (1)</stdlib.h></stdio.h>
e.txt","w"))!= NULL)	int i,j;		{

		•	I
	default:	void create()	front++;
printf("\nEnte		{	rear++;
r your choice :	printf("\nChoi	front = rear =	pri_que[rear]
");	ce is	-1;	= data;
scanf("%d",	incorrect,	}	return;
&ch);	Enter a correct	/* Function to	,
switch (ch)	choice");	insert value	J
{	1	into priority	else
case 1:	J	queue */	check(data);
	}	void	rear++;
printf("\nEnte	}	insert_by_pri	}
r value to be	//priority	ority(int data)	/* Function to
inserted : ");	queue	{	check priority
	#include	if (rear >=	and place
scanf("%d",&	<stdio.h></stdio.h>	MAX - 1)	element */
n);	#include	{	void check(int
	<stdlib.h></stdlib.h>	•	data)
insert_by_pri	#define MAX	printf("\nQue	{
ority(n);	5	ue overflow	int i,j;
break;	int	no more	for (i = 0; i <=
case 3:	pri_que[MAX]	elements can	rear; i++)
	;	be inserted");	{
display_pque	int front, rear;	return;	if (data >=
ue();	/* Function to	}	pri_que[i])
break;	create an	if ((front == -	\[ \frac{1}{2} = \frac{1}{2} = \frac{1}{2} \frac{1}{2} \]
case 4:	empty	1) && (rear	
exit(0);	priority	== -1))	for (j = rear +
	queue */	{	1; j > i; j)

<pre>{   pri_que[j] =   pri_que[j - 1];   }   pri_que[i] =   data;   return;   }   pri_que[i] =   data; }  /* Function to   display queue   elements */   void   display_pque   ue()   {   if ((front == -   1) &amp;&amp; (rear   == -1))    {   printf("\nQue</pre>	<pre>for (; front &lt;= rear; front++) {   printf(" %d ",   pri_que[front ]); } front = 0; } Slip.24 Que.2 //read data from 24-2- emp_insertio   n_name.txt file and sort the names #include<stdi o.h=""> #include<strin g.h=""> typedef struct {   char</strin></stdi></pre>	<pre>int salary; } RECORD; RECORD emp[200]; int readfile(RECO RD[]); void writefile(REC ORD[],int); int sort(RECORD *emp,int n); int main() {   int n;    n=readfile(em   p);   sort(emp,n);    writefile(emp,   n); }</pre>	<pre>int readfile(RECO RD *a) {   int i=0;   FILE *fp;   if   ((fp=fopen("2 4-2-   emp_insertio   n_name.txt","   r"))!=NULL)   while(!   feof(fp))   {   fscanf(fp,"%s   %d%d",a[i].na   me,&amp;a[i].age,   &amp;a[i].salary);   i++;   }   return i; }   void   writefile(REC</pre>
printf("\nQue ue is empty"); return;	{ char name[50]; int age;		

}	}	
int	Slip.26	printf("\n2.Po
sort(RECORD	Que.1	p");
<pre>*emp,int n) {    RECORD    temp;    int i,j;  for(i=1;i<n;i++< pre=""></n;i++<></pre>	//Array as a stack(static) #include <stdi o.h=""> #include<stdli b.h=""> #include "20-</stdli></stdi>	<pre>printf("\n3.Di splay");  printf("\n4.Pe ek");  printf("\n5.Ex</pre>
) {	1static_stack. h"	it");
temp=emp[i]; for(j=i-1;j>=0 && strcmp(emp[j ].name,temp. name)>0;j)	<pre>int main() {  int ch;  while(1) {</pre>	<pre>printf("\nEnte r your choice : "); scanf("%d",&amp; ch);</pre>
<pre>{ emp[j+1]=em p[j]; } emp[j+1]=te mp; }</pre>	<pre>printf("\nEnte r a number to perform operations on stack : ");  printf("\n1.Pu sh");</pre>	switch(ch) {   case 1 :   push();   break;   case 2 :   pop();   break;
	<pre>int sort(RECORD *emp,int n) {    RECORD temp; int i,j;  for(i=1;i<n;i++ )="" for(j="i-1;j" temp="emp[i];" {="">=0     &amp;&amp;    strcmp(emp[j ].name,temp.    name)&gt;0;j)    {    emp[j+1]=em    p[j];    }    emp[j+1]=te    mp;</n;i++></pre>	<pre>int sort(RECORD *emp,int n) {     RECORD temp; int i,j;     int i,j;     #include<stdi o.h="">     int main()  temp=emp[i]; for(j=i-1;j&gt;=0     &amp;&amp; strcmp(emp[j ].name,temp. name)&gt;0;j) {     printf("\nEnte     r a number to     perform     operations on     stack: "); }  emp[j+1]=te     mp;</stdi></pre>

case 3:	{	k	int i;
display();	int data;	Underflow");	if(top==-1)
break;		}	
case 4: printf("\nTop most element of stack is %d",peek());	printf("\nEnte r the data to push - ");	p=stack[top]; top=top-1;	<pre>printf("\nStac k Underflow"); else</pre>
break;	scanf("%d",& data);	printf("\nPop ed element is	{
case 5 : exit(0);	if(isfull()) {	%d",p);	for(i=0;i<=top ;i++)
<pre>default: printf("\nEnte r the correct choice : "); } }</pre>	<pre>printf("\nStac k Overflow"); } top+=1;</pre>	int peek(){  if(top==-1)  {	<pre>printf("stack[ %d]=%d\n",i,s tack[i]); }</pre>
#include <stdi o.h=""> #include<stdli b.h=""> #define MAX</stdli></stdi>	stack[top]=da ta; } int pop() {	<pre>printf("\nStac k Underflow"); exit(1); } return</pre>	<pre>int isfull() {   if(top==MAX- 1)   return 1;</pre>
3	int p; if(isempty())	stack[top];	else return 0;
int top=- 1,stack[MAX];	{	} void display()	}
void push()	printf("\nStac	{	int isempty()

{	Slip.26	int	FILE *fp;
if(top==-1)	Que.2	quicksort(REC	if
return 1;	//read the	ORD *emp,int low,int high);	((fp=fopen("e
else	data from	int	mp_quick_ag e.txt","r"))!=N
return 0;	emp_quick_a ge.txt and	partition(REC	ULL)
}	sort bt age	ORD *emp,	while(!
Que.2	#include <stdi< td=""><td>int low, int high);</td><td>feof(fp))</td></stdi<>	int low, int high);	feof(fp))
	o.h>	void	{
	#include <strin< td=""><td>sort(RECORD</td><td></td></strin<>	sort(RECORD	
SF026-	g.h>	*emp,int n);	fscanf(fp,"%s %d%d",a[i].na
	typedef struct	int main()	me,&a[i].age,
	{	{	&a[i].salary);
	char	int n;	i++;
Clin 27	name[30];		}
Slip.27	int age;	n=readfile(em	return i;
Que.1	int salary;	p);	}
Que.2	}	sort(emp,n);	void
	RECORD;	writefile(emp,	writefile(REC
	RECORD emp[100];	n);	ORD *a,int n)
SF026-	int	}	{
	readfile(RECO	int	int i=0;
	RD[]);	readfile(RECO	FILE * fp;
	void	RD *a)	;f//fw_fc/!!
Slip.28	writefile(REC	{	if((fp=fopen(" sorted emp
Que.1	ORD[],int);	int i=0;	quick_age.txt

","w"))!=NULL )	quicksort(em	{ while( start	return end;
<pre>for(i=0;i<n;i++ )="" );="" *emp,int="" <="" \t%d\t%d\n",="" a[i].name,a[i].="" age,a[i].salary="" fprintf(fp,"%s="" high="" high)="" if(="" int="" low="" low,int="" ord="" p="&lt;/pre" p;="" quicksort(rec="" {="" }=""></n;i++></pre>	<pre>p, p+1, high); }  void sort(RECORD *emp, int n) {  quicksort(em p,0,n-1); }  int partition(REC ORD *emp, int low, int high) {  RECORD pivot = emp[low];</pre>	<pre>{   while( start   &lt;= high &amp;&amp;   emp[start].ag   e &lt;= pivot.age )   start++;   while(   emp[end].age   &gt; pivot.age )   end;   if( start &lt; end )   {     RECORD     swap =     emp[start];     emp[end];     emp[end] =     swap;   } }</pre>	Slip.29 Que.1  //Array as a stack(static)  #include <stdi o.h="">  #include "20-1static_stack.h"  int main()  {  int ch;  while(1)  {  printf("\nEnter a number to</stdi>
p = partition(emp , low, high);	emp[low]; int start, end; start = low;	} emp[low] =	•
partition(emp	int start, end;	} emp[low] = emp[end];	perform operations on
p, low, p-1);	while( start < end )	emp[end] = pivot;	printf("\n1.Pu sh");

((11) - 2 - 5	case 3:	{	k
printf("\n2.Po p");	display();	int data;	Underflow");
Ρ /,	break;		}
printf("\n3.Di splay");	case 4: printf("\nTop most element of stack is	printf("\nEnte r the data to push - ");	p=stack[top]; top=top-1;
printf("\n4.Pe ek");	%d",peek()); break;	scanf("%d",& data);	printf("\nPop ed element is
printf("\n5.Ex	case 5 :	if(isfull())	%d",p);
it");	exit(0);	{	}
<pre>printf("\nEnte r your choice : ");</pre>	<pre>default:   printf("\nEnte   r the correct   choice : "); }</pre>	<pre>printf("\nStac k Overflow"); } top+=1;</pre>	int peek(){  if(top==-1)  {
scanf("%d",& ch); switch(ch)	} #include <stdi o.h&gt;</stdi 	stack[top]=da ta; }	printf("\nStac k Underflow");
{	#include <stdli b.h&gt;</stdli 	int pop()	exit(1);
case 1 : push();	#define MAX	{ int p;	return stack[top];
break;	int top=-	if(isempty())	}
case 2 : pop();	1,stack[MAX];	{	void display()
break;	void push()	printf("\nStac	{

int i;	{	RECORD	FILE *fp;
if(top==-1)	if(top==-1)	emp[100];	if
<pre>printf("\nStac k Underflow"); else { for(i=0;i&lt;=top ;i++)</pre>	return 1; else return 0; } Slip.30 Que.2 //read the data from txt file and sort	<pre>int readfile(RECO RD[]); void writefile(REC ORD[],int); int sort(RECORD *emp,int n); int main()</pre>	<pre>((fp=fopen("3     O-     2emp_bubble     .txt","r"))!=N     ULL)     while(!     feof(fp))     {     fscanf(fp,"%s</pre>
printf("stack[ %d]=%d\n",i,s tack[i]);	the names using bubble sort	{ int n;	%d%d",a[i].na me,&a[i].age, &a[i].salary);
<pre>int isfull() {   if(top==MAX- 1)   return 1;   else   return 0; } int isempty()</pre>	<pre>#include<stdi o.h=""> #include<strin g.h=""> typedef struct {    char    name[30];    int age;    int salary; } RECORD;</strin></stdi></pre>	<pre>n=readfile(em p); sort(emp,n); writefile(emp, n); } int readfile(RECO RD *a) { int i=0;</pre>	<pre>i++; } return i-1; } void writefile(REC ORD *a,int n) { int i=0; FILE * fp;  if((fp=fopen(" 20</pre>
			30-

```
2sortedemp_
                   {
bubble.txt","
                   int i,j,pass;
w"))!=NULL)
                   for(pass=1;pa
for(i=0;i<n;i++
                   ss<n;pass++)
                   {
                   for(i=0;i<n-
fprintf(fp,"%s
                   pass;i++)
t%d\t%d\n'',
a[i].name,a[i].
                   {
age,a[i].salary
                   if (strcmp
);
                   (emp[i].name,
                   emp[i+1].nam
}
                   e)>0)
                   {
void
swap(RECOR
D *a, RECORD
                   swap(&emp[i]
*b)
                   ,&emp[i+1]);
                   }}}
{
                   //make text
RECORD
tmp=*a;
                   file name as
                   30-
*a=*b;
                   2emp_bubble
*b=tmp;
                   .txt save it in
}
                   vs code
                   //and output
                   is displayed in
int
                   the 30-
sort(RECORD
* emp,int n)
```

2sortedemp\_ bubble.txt file