



**PARALA MAHARAJA ENGINEERING COLLEGE, BERHAMPUR**

**Dept. of Computer Sc. & Engg.**

**“DATA STRUCTURE”  
LABORATORY MANUAL**

**Kalyan Kumar Jena**

# CONTENTS

## ***Experiment No.1***

Write a C program to perform matrix multiplication using array.

## ***Experiment No.2***

Write a C program to create a stack using an array and perform  
(i) push operation (ii) pop operation

## ***Experiment No.3***

Write a C program to create a queue and perform  
i) insert ii) delete iii)traverse

## ***Experiment No. 4***

Write a C program that uses Stack operation for Converting infix expression into postfix expression

## ***Experiment No. 5***

Write a C program that uses functions to perform the following operations on Single linked list:  
i) Creation ii) Traversal

## ***Experiment No. 6***

Write a C program that uses functions to perform the following operations on Double linked list:  
i) Creation ii) Traversal

## ***Experiment No. 7***

Write C programs to perform the Linear search operation for a Key value in a given list of integers.

## ***Experiment No. 8***

Write C program to perform the Binary search operation for a Key value in a given list of integers.

## ***Experiment No.9***

Write a C program that implement Bubble Sort method to sort a given list of integers in descending order.

## ***Experiment No.10***

Write a C program that implement Quick Sort method to sort a given list of integers in ascending order:

### ***Experiment No.1***

Write a C program to perform matrix multiplication using array.

#### **PROGRAM**

```
#include<stdio.h>

#include<conio.h>

void mul(int [ ][ ],int [ ][ ]);

void main()

{

int x[3][3],y[3][3],i,j;

clrscr();

printf("enter elements to 1st matrix");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

scanf("%d",&x[i][j]);

}

}

printf("enter elements to 2nd matrix");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

scanf("%d",&y[i][j]);

}

}
```

```
}  
mul(x,y);  
getch();  
}  
void mul(int a[ ][ ],int b[ ][ ])  
{  
int i,j,k,c[3][3];  
for(i=0;i<3;i++)  
{  
for(j=0;j<3;j++)  
{  
c[i][j]=0;  
for(k=0;k<3;k++)  
{  
c[i][j]=c[i][j]+a[i][k]*b[k][j];  
}  
}  
}  
printf("the resultant matrix is \n \n");  
for(i=0;i<3;i++)  
{  
for(j=0;j<3;j++)  
{  
printf("%d\t",c[i][j]);
```

```
}
```

```
printf("\n");
```

```
}
```

### ***Experiment No.2***

**Write a C program to create a stack using an array and perform  
(i) push operation (ii) pop operation**

#### **PROGRAM**

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#define maxsize 20

void push();

int pop();

void display();

int stack[maxsize];

int top=-1;

void main()

{

    int x;

    char ch='1';

    while(ch!='4')

    {

        printf("\n 1.push");

        printf("\n 2.pop");

        printf("\n 3.display");

        printf("\n 4.quit");

        printf("\n enter your choice:");

        fflush(stdin);
```

```
ch=getchar();
switch(ch)
{
case '1':
push();
break;
case '2':
x=pop();
printf("\n the deleted element is %d",x);
break;
case '3':
display();
break;
case '4':
break;
default:
printf("\n wrong choice entered");
}
}
}
void push()
{
int v;
if(top==maxsize-1)
```

```
{  
printf("\n the stack is full");  
getch();  
exit(0);  
}  
else  
{  
printf("\n enter the element to be inserted");  
scanf("%d",&v);  
top++;  
stack[top]=v;  
}  
}  
int pop()  
{  
int v;  
if(top== -1)  
{  
printf("\n the stack is empty");  
getch();  
exit(0);  
}  
else  
{
```



```
v=stack[top];
top=top-1;
}
return(v);
}
void display()
{
int i;
if(top==-1)
{
printf("the stack is empty");
getch();
exit(0);
}
else
{
printf("\n elements are:");
for(i=top;i>=0;i--)
{
printf("%d\n",stack[i]);
}
}
}
```

### ***Experiment No.3***

Write a C program to create a queue and perform

i) insert ii) delete iii)traverse

#### **PROGRAM**

```
#include<stdio.h>
#include<conio.h>
#define maxsize 50
struct queue
{
int elements[maxsize];
int front,rear;
}q;
void add(int);
int delete();
void display();
void main()
{
int x;
char ch='1';
clrscr();
q.front=0;
q.rear=-1;
while(ch!='4')
{
printf("\n 1.insert");
printf("\n 2.delete");
printf("\n 3.traverse");
printf("\n 4.quit");
printf("\n enter your choice");
fflush(stdin);
ch=getchar();
switch(ch)
{
case '1';
printf("enter element to be added");
scanf("%d",&x);
add(x);
break;
case '2':
x=delete();
printf("\n deleted element is %d",x);
break;
case '3':
display();
break;
}
```

```
case '4':
break;
default:
printf("\n wrong choice");
}
}
}
void add(int y)
{
++q.rear;
if(q.rear<maxsize)
{
q.elements[q.rear]=y;
}
else
{
printf("queue overflow");
}
}
void delete()
{
if(q.front>q.rear)
{
printf("queue empty");
}
else
{
x=q.elements[q.front];
q.front++;
}
return x;
}
void display()
{
int i;
printf("elements in the queue are \n");
for(i=q.front;i<=q.rear;i++)
{
printf("%d\n",q.elements[i]);
}
}
```

#### ***Experiment No. 4***

Write a C program that uses Stack operation for Converting infix expression into postfix expression

#### **PROGRAM**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
char str[100];
int top=-1;
void push(char s)
{
top=top+1;
str[top]=s;
}
char pop()
{
char i;
if(top==-1)
{
printf("\n the stack is empty");
getch();
return 0;
}
else
{
i=str[top];
top=top-1;
}
return i;
}
int precedence(char c)
{
if(c=='$' || c=='^')
return 4;
if(c=='/' || c=='*')
return 3;
if(c=='+' || c=='-')
return 2;
return 1;
}
void infix2pofx(char in[ ])
{
int l;
static int i=0,px=0;
char s,t;
char pofx[80];
```

```

l=strlen(in);
while(i<1)
{
s=in[i];
switch(s)
{
case '(':
push(s);
break;
case ')':
t=pop();
while(t!='(')
{
pofx[px]=t;
px=px+1;
t=pop();
}
break;
case '+':
case '-':
case '*':
case '/':
case '^':
while(precedence(str[top])>=precedence(s))
{
t=pop();
pofx[px]=t;
px++;
}
push(s);
break;
default:
pofx[px++]=s;
break;
}
i=i+1;
}
while(top>-1)
{
t=pop();
pofx[px++]=t;
}
pofx[px++]='\0';
puts(pofx);
return;
}
int main()
{
char ifx[50];

```

```
printf("\n enter the infix expression");  
gets(ifx);  
infx2pofx(ifx);  
getch();  
return 0;  
}
```

### ***Experiment No. 5***

Write a C program that uses functions to perform the following operations on Single linked list:

i) Creation ii) Traversal

### **PROGRAM**

```
#include<stdio.h>
#include<conio.h>
#include<alloc.h>
struct slink
{
int info;
struct slink *next;
};
void create(struct slink *);
void display(struct slink *);
void main()
{
struct slink *node;
node=(struct slink *)malloc(sizeof(struct slink));
create(node);
display(node);
getch();
}
void create(struct slink *node)
{
char ch;
printf("\n enter the information");
scanf("%d",&node->info);
node->next=NULL;
printf("\n do you want to create more nodes(y/n)");
fflush(stdin);
ch=getchar();
while(ch=='y' || ch=='Y')
{
node->next=(struct slink *)malloc(sizeof(struct slink));
node=node->next;
printf("\n enter the value");
scanf("%d",&node->info);
node->next=NULL;
printf("\n do you want to create more nodes(y/n)");
fflush(stdin);
ch=getchar();
}
}
void display(struct slink *node)
```

```
{  
printf("\n the nodes are:");  
while(node!=NULL)  
{  
printf("%d->",node->info);  
node=node->next;  
}  
}
```



### ***Experiment No. 6***

Write a C program that uses functions to perform the following operations on Double linked list:

i) Creation ii) Traversal

#### **PROGRAM**

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct dlink *prev;
int info;
struct dlink *next;
}*node;
void create(struct dlink*);
void show(struct dlink*);
void main()
{
node=(struct dlink*)malloc(sizeof(struct dlink*));
clrscr();
create(node);
show(node);
getch();
}
void create(struct dlink *node)
{
struct dlink *ptr;
int i=1;
char ch;
printf("\n enter the information for the node %d:",i);
scanf("%d",&node->info);
node->next=NULL;
node->prev=NULL;
i=i+1;
printf("\n do you want to create more nodes(y/n):");
fflush(stdin);
ch=getchar();
while(ch=='y' || ch=='Y')
{
node->next=(struct dlink*)malloc(sizeof(struct dlink));
ptr=node;
node=node->next;
printf("\n enter the information for the node %d:",i);
scanf("%d",&node->info);
node->next=NULL;
node->prev=ptr;
i=i+1;
printf("\n do you want to create more nodes(y/n)");
fflush(stdin);
```

```
ch=getchar();
}
}
void show(struct dlink *node)
{
struct dlink *temp;
printf("\n the link list in forward direction:\n \n ");
while (node!=NULL)
{
temp=node;
printf("%d->",node->info);
node=node->next;
}
printf("\n the link list in forward direction:\n \n");
node=temp;
while(node!=NULL)
{
printf("%d->",node->info);
node=node->prev;
}
printf("\n the link list in backward direction:\n \n");
node temp;
while(node!=NULL)
{
printf("%d->",node->info);
node=node->prev;
}
}
```

### ***Experiment No. 7***

Write C programs to perform the Linear search operation for a Key value in a given list of integers.

#### **PROGRAM**

```
#include<stdio.h>
#include<conio.h>
int lsearch (int [ ],int,int);
void main()
{
int x[50],i,k,s,n;
clrscr();
printf("specify the number of elements to be entered");
scanf("%d",&n);
printf("\n enter numbers");
for(i=0;i<n;i++)
{
scanf("%d",&x[i]);
}
printf("\n the elements are\n");
for(i=0;i<n;i++)
{
printf("%d\t",x[i]);
}
printf("\n enter the element to be searched");
scanf("%d",&k);
s=lsearch(x,n,k);
if(s==-1)
{
printf("\n element is not found");
}
else
{
printf("\n element is found at %d position",s);
}
getch();
}
int lsearch(int a[ ],int no,int key)
{
int i;
for(i=0;i<no;i++)
{
if(key==a[i])
{
return(i);
}
}
return(-1);
}
```

### ***Experiment No. 8***

Write C program to perform the Binary search operation for a Key value in a given list of integers.

#### **PROGRAM**

```
#include<stdio.h>
#include<conio.h>
int bsearch (int [ ],int,int);
void main()
{
int x[50],i,k,s,n;
clrscr();
printf("specify the number of elements to be entered");
scanf("%d",&n);
printf("\n enter numbers");
for(i=0;i<n;i++)
{
scanf("%d",&x[i]);
}
printf("\n the elements are\n");
for(i=0;i<n;i++)
{
printf("%d\t",x[i]);
}
printf("\n enter the element to be searched");
scanf("%d",&k);
s=bsearch(x,n,k);
if(s==-1)
{
printf("\n element is not found");
}
else
{
printf("\n element is found at %d position",s);
}
getch();
}
int bsearch(int a[ ],int n,int key)
{
int l,h,m;
l=0;
h=n-1;
while(l<=h)
{
m=(l+h)/2;
if(key==a[m])
{
return(m);
}
}
```

```
if(key<a[m])
{
h=m-1;
}
else
{
l=m+1;
}
}
return(-1);
}
```

### ***Experiment No.9***

Write a C program that implement Bubble Sort method to sort a given list of integers in descending order.

#### **PROGRAM**

```
#include<stdio.h>
#include<conio.h>
void bsort (int [ ],int);
void main()
{
int x[50],i,n;
clrscr();
printf("specify the number of elements to be entered");
scanf("%d",&n);
printf("\n enter numbers");
for(i=0;i<n;i++)
{
scanf("%d",&x[i]);
}
printf("\n the elements before sorting are\n");
for(i=0;i<n;i++)
{
printf("%d\t",x[i]);
}
bsort(x,n);
getch();
}
void bsort(int a[],int no)
{
int i,j,temp;
for(i=0;i<no;i++)
{
for(j=0;j<no-i;j++)
{
if(a[j]<a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
printf("\n after sorting the elements are\n");
for(i=0;i<no;i++)
{
printf("%d\t",a[i]);
}
}
```

### ***Experiment No.10***

Write a C program that implement Quick Sort method to sort a given list of integers in ascending order:

#### **PROGRAM**

```
#include<stdio.h>
#include<conio.h>
void qsort(int *,int,int);
int partition(int *,int,int);
void main()
{
int x[50],i,n;
clrscr();
printf("specify the number of elements to be entered");
scanf("%d",&n);
printf("\n enter numbers");
for(i=0;i<n;i++)
{
scanf("%d",&x[i]);
}
printf("\n the elements before sorting are\n");
for(i=0;i<n;i++)
{
printf("%d\t",x[i]);
}
qsort(x,0,n-1);
printf("\n after sorting the elements are\n");
for(i=0;i<n;i++)
{
printf("%d\t",x[i]);
}
getch();
}
void qsort(int a[ ],int l,int u)
{
int i;
if(u>l)
{
i=partition(a,l,u);
qsort(a,l,i-1);
qsort(a,i+1,u);
}
}
int partition(int a[ ],int l,int u)
{
int i,p,q,t;
p=l+1;
q=u;
```

```
i=a[l];
while(q>=p)
{
while(a[p]<i)
{
p++;
}
while(a[q]>i)
{
q--;
}
if(q>p)
{
t=a[p];
a[p]=a[q];
a[q]=t;
}
}
t=a[l];
a[l]=a[q];
a[q]=t;
return q;
}
```