



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: SE Data Science

Subject: Data Structure Lab

Name of Instructor: Prof. Rajashri Chaudhari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 27/07/2022

Date Of Submission: 01/08/2022

---

### Experiment No. 11

**Aim:-** Write a C program to implement STACK using an Array.

**Program:**

```
#include<stdlib.h>
```

```
#include<stdio.h>
```

```
#define max 5
```

```
int st[max],i=0,top=-1,ch,ele,item,size;
```

```
void push();
```

```
void pop();
```

```
void peek();
```

```
int main()
```

```
{
```

```
printf("Enter the array size: ");
```

```
scanf("%d",&size);
```



---

```
while(i)

{

printf("\n 1.Push \n 2.Pop \n 3.Peek \n 4.Exit\n");

printf("ENter your choice");

scanf("%d",&ch);

switch(ch)

{

case 1:push();

break;

case 2:pop();

break;

case 3:peek();

break;

case 4:exit(0);

default:

printf("Please enter a valid number");

}

}

}

void push()
```



```
{  
  
if(top==max-1)  
  
[  
  
printf("Stack is overflow,cant insert");  
  
return;  
  
}  
  
else  
  
{  
  
printf("Enter element: ");  
  
scanf("%d",&ele);  
  
top++;  
  
st[top]=ele;  
  
}  
  
}  
  
  
void pop()  
  
{  
  
if(top==-1)  
  
{  
  
printf("\n Stack is underflow");  
  
return;
```



```
}  
  
else  
  
{  
  
item=st[top];  
  
top--;  
  
printf("\n %d is deleted",item);  
  
}  
  
}  
  
  
  
void peek()  
  
{  
  
if(top==-1)  
  
{  
  
printf("\n Stack is underflow");  
  
return;  
  
}  
  
else  
  
{  
  
for(i=top;i>=0;i--)  
  
{  
  
printf("\n %d",st[i]);
```



}

}

## Output:-

```
Enter the size of array:
5
```

- 1.Push Operation
- 2.Pop Operation
- 3.Peek Operation
- 4.Exit

```
Enter your choice:
1
```

```
Enter the element: 10
```

- 1.Push Operation
- 2.Pop Operation
- 3.Peek Operation
- 4.Exit

```
Enter your choice:
1
```

```
Enter the element: 20
```

- 1.Push Operation
- 2.Pop Operation
- 3.Peek Operation
- 4.Exit

```
Enter your choice:
1
```

```
Enter the element: 30
```

- 1.Push Operation
- 2.Pop Operation
- 3.Peek Operation
- 4.Exit

```
Enter your choice:
3
```

```
30
20
10
```

- 1.Push Operation
- 2.Pop Operation
- 3.Peek Operation
- 4.Exit



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

Enter your choice:

2

30 is deleted

1.Push Operation

2.Pop Operation

3.Peek Operation

4.Exit

Enter your choice:

2

20 is deleted

1.Push Operation

2.Pop Operation

3.Peek Operation

4.Exit

Enter your choice:

3

10

1.Push Operation

2.Pop Operation

3.Peek Operation

4.Exit

Enter your choice:

4

apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~\$ |



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: SE Data Science

Subject: Data Structure Lab

Name of Instructor: Prof. Rajashri Chaudhari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 10/08/2022

Date Of Submission: 17/08/2022

---

### Experiment No. 2

**Aim:-** Write a C program to implement linear search and binary search techniques.

#### Program

```
// Program for infix expression
```

```
#include <stdio.h>
```

```
int icp(char ch)
```

```
{
```

```
switch(ch)
```

```
{
```

```
case '%':
```

```
case '^':return 4;
```

```
case '*':
```

```
case '/':
```

```
case '-':
```

```
case '+':return 2;
```



---

```
case '(':return -1;

case ')':return 6;

}

return 0;

}

int isp(char ch)

{

switch(ch)

{

case '^':

case '%':return 5;

case '*':

case '/':return 4;

case '+':

case '-':return 3;

case '(':return 7;

case ')':return 0;

case '#':return -1;

}

return 0;

}
```





---

```
void main()

{

char expr[25],post[25],stack[25];

int i=0,j=0,top=-1;

char ch;

printf("\n Enter infix Expression:");

scanf("%s",expr);

top++;

stack[top]='#';

for(i=0;expr[i]!='\0';i++)

{

ch=expr[i];

if(ch=='%'||ch=='%'||ch=='^'||ch=='*'||ch=='/'||ch=='+'||ch=='-')

{

while(icp(ch)<=isp(stack[top]))

{

post[j]=stack[top];

top--;

j++;

}

top++;
```



---

```
stack[top]=ch;

}

else if(ch=='(')

{

while(stack[top]!='(')

{

post[j]=stack[top];

top--;

j++;

}

top--;

}

else

{

post[j]=ch;

j++;

}}

while(stack[top]!='#')

{

post[j]=stack[top];

top--;
```



---

```
j++;  
  
}  
  
post[j]='\0';  
  
printf("\n Postfix Expression is = %s\n",post);  
  
}
```

## Output:-

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ gcc -o exp2 exp2.c  
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ ./exp2  
  
Enter infix Expression:a+b*c  
  
Postfix Expression is = ab+c*  
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ |
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: SE Data Science

Subject: Data Structure Lab

Name of Instructor: Prof. Rajashri Chaudhari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 17/08/2022

Date Of Submission: 17/08/2022

---

### Experiment No. 3

**Aim:-** To evaluate postfix expression using stack ADT Assignment

**Program:**

```
#include<stdio.h>
```

```
#include<ctype.h>
```

```
int stack[20];
```

```
int top=-1;
```

```
void push(int x)
```

```
{
```

```
stack[++top]=x;
```

```
}
```

```
int pop()
```

```
{
```



---

```
return stack[top--];
```

```
}
```

```
int main()
```

```
{
```

```
char exp[20];
```

```
char *e; //ctpe.h function header file
```

```
int n1,n2,n3,num;
```

```
printf("Enter the expression: ");
```

```
scanf("%s",exp);
```

```
e=exp;
```

```
while(*e!='\0')
```

```
{
```

```
if(isdigit(*e))
```

```
{
```

```
num=*e-48;
```

```
push(num);
```

```
}
```

```
else
```

```
{
```



```
n2=pop();
```

```
n1=pop();
```

```
switch(*e)
```

```
{
```

```
case '+':
```

```
n3=n1+n2;
```

```
break;
```

```
case '-':
```

```
n3=n1-n2;
```

```
break;
```

```
case '*':
```

```
n3=n1*n2;
```

```
break;
```

```
case '/':
```

```
n3=n1/n2;
```

```
break;
```

```
}
```

```
push(n3);
```

```
}
```

```
e++;
```



```
}  
  
printf("\n The result of expression %s is = %d\n",exp,pop());  
  
return 0;  
  
}
```

## Output:-

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ gcc -o exp3 exp3.c  
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ ./exp3  
Enter the expression: 42+  
  
The result of expression 42+ is = 6  
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ ./exp3  
Enter the expression: 73-  
  
The result of expression 73- is = 4  
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ ./exp3  
Enter the expression: 32*  
  
The result of expression 32* is = 6  
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ ./exp3  
Enter the expression: 42/  
  
The result of expression 42/ is = 2  
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ |
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: SE Data Science

Subject: Data Structure Lab

Name of Instructor: Prof. Rajashri Chaudhari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 24/08/2022

Date Of Submission: 24/08/2022

---

### Experiment No. 4

**Aim :-** Write a program to implement linear queue ADT using array

**Program:**

```
#include #include #define MAX 50
```

```
int queue_arr[MAX];
```

```
int rear=-1;
```

```
int front=-1;
```

```
void enqueue( int item);
```

```
int dequeue();
```

```
int peek();
```

```
void display();
```

```
int isFull();
```

```
int isEmpty();
```

```
int main()
```





---

```
{  
  
int choice,item;  
  
while(1)  
  
{  
  
printf("\n1.Insert\n");  
  
printf("2.Delete\n");  
  
printf("3.Display element at the front\n");  
  
printf("4.Display all element at the front\n");  
  
printf("5.Quit\n");  
  
printf("\nEnter your choice :");  
  
scanf("%d",&choice);  
  
  
switch(choice)  
  
{  
  
case 1:  
  
printf("\nInput the elements for adding in queue");  
  
scanf("%d",&item);  
  
enqueue(item);  
  
break;  
  
  
case 2:
```



---

```
item=dequeue();

printf("\nDeleted elements is %d\n",item);

break;

case 3:

printf("\nElement at the front is %d\n",peek());

break;

case 4:

display();

break;

case 5:

exit(1);

break;

default:

printf("\nWrong choice\n");

} //End of switch

} //End of while
```



```
return 0;

} // End of main()

void enqueue(int item)
{
    if( isFull())
    {
        printf("\nQueue Overflow\n");
        return;
    }
    if( front == -1)
        front = 0;
    rear = rear + 1;
    queue_arr[rear] = item;
} // End of enqueue

int dequeue()
{
    int item;
    if( isEmpty())
    {
        printf("\nQueue Underflow\n");
```



---

```
exit(1);

}

item=queue_arr[front];

front=front+1;

return item;

} //End of dequeue


int peek()

{

if ( isEmpty())

{

printf("\nQueue Underflow\n");

exit(1);

}

return queue_arr[front];

} //End of peek


int isEmpty()

{

if( front==-1 || front==rear+1 )

return 1;
```



else

return 0;

}// End of isEmpty()

int isFull()

{

if( rear==MAX-1 )

return 1;

else

return 0;

}// End of isFull

void display()

{

int i;

if( isEmpty())

{

printf("\nQueue is empty\n");

return ;

}

printf("\nQueue is :\n");



```
for(i=front;i<=rear;i++)  
  
printf("%d",queue_arr[i]);  
  
printf("\n");  
  
} // End of display()
```

## Output:-

```
apsit@apsit-HP-280-G2-SFF:~$ gedit expt4.c  
apsit@apsit-HP-280-G2-SFF:~$ gcc -o expt4 expt4.c  
apsit@apsit-HP-280-G2-SFF:~$ ./expt4  
  
1.Insert  
2.Delete  
3.Display all the elements of the queue  
4.Quit  
  
Enter your choice:1  
  
Input the element for adding in queue:10  
  
1.Insert  
2.Delete  
3.Display all the elements of the queue  
4.Quit  
  
Enter your choice:1  
  
Input the element for adding in queue:20  
  
1.Insert  
2.Delete  
3.Display all the elements of the queue  
4.Quit  
  
Enter your choice:2  
  
Deleted element is 10
```



```
1.Insert
2.Delete
3.Display all the elements of the queue
4.Quit

Enter your choice:1

Inputthe element for adding in queue:30

1.Insert
2.Delete
3.Display all the elements of the queue
4.Quit

Enter your choice:3

Queue is:

2030

1.Insert
2.Delete
3.Display all the elements of the queue
4.Quit

Enter your choice:1

Inputthe element for adding in queue:40

Inputthe element for adding in queue:40

1.Insert
2.Delete
3.Display all the elements of the queue
4.Quit

Enter your choice:3

Queue is:

203040

1.Insert
2.Delete
3.Display all the elements of the queue
4.Quit

Enter your choice:4
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: CSE(DS)

Subject: DS

Name of Instructor: Prof. Rajashri Chaudhari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 07/09/22

Date Of Submission: 07/09/22

---

### Experiment No. 5

**Aim:-** To implement circular queue using array

#### Program:

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#define MAX 5
```

```
int cqueue_arr[MAX];
```

```
int front=-1;
```

```
int rear=-1;
```

```
void insert(int item)
```

```
{
```

```
    if(((front==0 && rear==MAX-1)|| (front==rear+1))
```

```
    {
```

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





```
return;

}

if(front== -1)

{

front=0;

rear=0;

}

else

{

if(rear==MAX-1)

rear=0;

else

rear=rear+1;

}

cqueue_arr[rear]=item;

}
```

```
void deletion()
```

```
{

    if(front== -1)

    {
```



---

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

return;

}

printf("Element deleted from queue is :%d\n",cqueue_arr[front]);

if(front==rear)

{

front=-1;

rear=-1;

}

}

void display()

{

int front_pos=front,rear_pos=rear;

if(front==rear)

{

printf("Queue is Empty\n");

return;

}

printf("Queue elements :\n");

if(front_pos<=rear_pos)
```



---

```
while(front_pos<=rear_pos)

{

printf("%d\n",cqueue_arr[front_pos]);

front_pos++;

}

else

{

while(front_pos<=MAX-1)

{

printf("%d\n",cqueue_arr[front_pos]);

front_pos=0;

while(front_pos<=rear_pos)

{

printf("%d\n",cqueue_arr[front_pos]);

front_pos++;

}

}

printf("\n");

}

}

int main()
```



```
{  
  
int choice,item;  
  
do  
  
{  
  
printf("1.Insert\n");  
  
printf("2.Delete\n");  
  
printf("3.Display\n");  
  
printf("4.Quit\n");  
  
printf("Enter your choice :");  
  
scanf("%d",&choice);  
  
switch(choice)  
  
{  
  
case 1:  
  
printf("Input the elements for insertion in queue :");  
  
scanf("%d",&item);  
  
insert(item);  
  
break;  
  
  
case 2:  
  
deletion();  
  
break;
```



case 3:

display();

break;

case 4:

exit(0);

default:

printf("Wrong choice\n");

}

}while(choice!=4);

return 0;

}



## Output:-

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ gcc -o exp5 exp5.c
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ ./exp5
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice :1
Input the elements for insertion in queue :10
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice :1
Input the elements for insertion in queue :20
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice :1
Input the elements for insertion in queue :30
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice :1
Input the elements for insertion in queue :40
1.Insert
2.Delete
3.Display
4.Quit
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

```
Enter your choice :2
Element deleted from queue is :10
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice :2
Element deleted from queue is :10
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice :3
Queue elements :
10
20
30
40
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice :4
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$
```



---

**Department of Computer Science and Engineering**  
**Data Science**

**Academic Year: 2022-2023**

**Semester: 3**

**Class / Branch: SE/DS**

**Subject: Data structure**

**Name of Instructor: Prof. Rajashree Chaudhari**

**Name of Student: Arya Patil**

**Student ID: 21107009**

**Date Of Performance: 21/09/22**

**Date Of Submission: 21/09/22**

---

**Experiment No. 6**

**Aim:-** : Write a C program to implement singly linked list ADT.

PROGRAM:

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
struct node{
```

```
int info;
```

```
struct node* link;
```

```
};
```

```
struct node* start=NULL;
```

```
void createlist()
```

```
{
```

```
if(start==NULL){
```

```
int n;
```

```
printf("\nEnter the number of nodes:");
```





---

```
scanf("%d",&n);

if(n!=0){

int data;

struct node* newnode;

struct node* temp;

newnode=malloc(sizeof(struct node));

start=newnode;

temp=start;

printf("\nEnter number to be inserted:");

scanf("%d",&data);

start->info=data;

for(int i=2;i<=n;i++){

newnode=malloc(sizeof(struct node));

temp->link=newnode;

printf("\nEnter number to be inserted:");

scanf("%d",&data);

newnode->info=data;

temp=temp->link;

}

}

printf("\nThe list is created\n");
```



```
}  
  
else  
  
printf("\nThe list is already created\n");  
  
}  
  
void traverse()  
  
{  
  
struct node* temp;  
  
if(start==NULL)  
  
printf("\nList is empty\n");  
  
else{  
  
temp=start;  
  
while(temp!=NULL){  
  
printf("Data =%d\n",temp->info);  
  
temp=temp->link;  
  
}  
  
}  
  
}  
  
void insertAtFront()  
  
{  
  
int data;
```



---

```
struct node* temp;

temp=malloc(sizeof(struct node));

printf("\nEnter number to be inserted:");

scanf("%d",&data);

temp->info=data;

temp->link=start;

start=temp;

}

void insertAtEnd()

{

int data;

struct node *temp,*head;

temp=malloc(sizeof(struct node));

printf("\nEnter number to be inserted:");

scanf("%d",&data);

temp->link=0;

temp->info=data;

head=start;

while(head->link!=NULL){

head=head->link;

}

}
```



---

```
head->link=temp;

}

void insertAtPosition()

{

struct node *temp,*newnode;

int pos,data,i=1;

newnode=malloc(sizeof(struct node));

printf("\nEnter position and data:");

scanf("%d %d",&pos,&data);

temp=start;

newnode->info=data;

newnode->link=0;

while(i<pos-1){

temp=temp->link;

i++;

}

newnode->link=temp->link;

temp->link=newnode;

}

void deleteFirst(){

struct node* temp;
```



---

```
if (start==NULL)

printf("\nList is empty\n");

else{

temp=start;

start=start->link;

free(temp);

}

}

void deleteEnd(){

struct node *temp,*prevnode;

if(start==NULL)

printf("\nList is Empty\n");

else{

temp=start;

while(temp->link!=0){

prevnode=temp;

temp=temp->link;

}

free(temp);

prevnode->link=0;
```



---

```
}  
  
}  
  
void deletePosition(){  
  
    struct node *temp,*position;  
  
    int i=1,pos;  
  
    if(start==NULL)  
  
        printf("\nList is empty\n");  
  
    else{  
  
        printf("\nEnter index:");  
  
        scanf("%d",&pos);  
  
        position=malloc(sizeof(struct node));  
  
        temp=start;  
  
        while(i<pos-1){  
  
            temp=temp->link;  
  
            i++;  
  
        }  
  
        position=temp->link;  
  
        temp->link=position->link;  
  
        free(position);  
  
    }  
  
}
```



---

```
int main()

{

int choice;

while(1){

printf("\n\t 1.To display list\n");

printf("\n\t 2. For insertion at starting\n");

printf("\n\t 3. For insertion at end\n");

printf("\n\t 4. For insertion at any position\n");

printf("\n\t 5. For deletion of first element\n");

printf("\n\t 6. For deletion of last element\n");

printf("\n\t 7. For deletion of element at any position\n");

printf("\n\t 8. To exit\n");

printf("\nEnter choice:\n");

scanf("%d",&choice);

switch(choice){

case 1:

traverse();

break;

case 2:

insertAtFront();

break;
```



case 3:

```
insertAtEnd();
```

```
break;
```

case 4:

```
insertAtPosition();
```

```
break;
```

case 5:

```
deleteFirst();
```

```
break;
```

case 6:

```
deleteEnd();
```

```
break;
```

case 7:

```
deletePosition();
```

```
break;
```

case 8:

```
exit(1);
```

```
break;
```

default:

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





```
}  
  
}  
  
return 0;  
  
}
```

## Output:-

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ gcc -o exp6 exp6.c  
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ ./exp6  
  
1.To display list  
2. For insertion at starting  
3. For insertion at end  
4. For insertion at any position  
5. For deletion of first element  
6. For deletion of last element  
7. For deletion of element at any position  
8. To exit  
Enter choice:  
2  
Enter number to be inserted:10  
  
1.To display list  
2. For insertion at starting  
3. For insertion at end  
4. For insertion at any position  
5. For deletion of first element  
6. For deletion of last element  
7. For deletion of element at any position  
8. To exit
```



```
Enter choice:
20
INcorrect choice

      1.To display list
      2. For insertion at starting
      3. For insertion at end
      4. For insertion at any position
      5. For deletion of first element
      6. For deletion of last element
      7. For deletion of element at any position
      8. To exit
```

```
Enter choice:
3
Enter number to be inserted:30

      1.To display list
      2. For insertion at starting
      3. For insertion at end
      4. For insertion at any position
      5. For deletion of first element
      6. For deletion of last element
      7. For deletion of element at any position
      8. To exit
```

```
Enter choice:
3
Enter number to be inserted:40

      1.To display list
      2. For insertion at starting
      3. For insertion at end
      4. For insertion at any position
      5. For deletion of first element
      6. For deletion of last element
      7. For deletion of element at any position
      8. To exit

Enter choice:
4
Enter position and data:2 50

      1.To display list
      2. For insertion at starting
      3. For insertion at end
      4. For insertion at any position
      5. For deletion of first element
      6. For deletion of last element
      7. For deletion of element at any position
      8. To exit
```



Enter choice:

5

- 1.To display list
2. For insertion at starting
3. For insertion at end
4. For insertion at any position
5. For deletion of first element
6. For deletion of last element
7. For deletion of element at any position
8. To exit

Enter choice:

6

- 1.To display list
2. For insertion at starting
3. For insertion at end
4. For insertion at any position
5. For deletion of first element
6. For deletion of last element
7. For deletion of element at any position
8. To exit

Enter choice:

7

Enter index:1

- 1.To display list
2. For insertion at starting
3. For insertion at end
4. For insertion at any position
5. For deletion of first element
6. For deletion of last element
7. For deletion of element at any position
8. To exit

Enter choice:

1

Data =50

- 1.To display list
2. For insertion at starting
3. For insertion at end
4. For insertion at any position
5. For deletion of first element
6. For deletion of last element
7. For deletion of element at any position
8. To exit



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

```
Enter choice:
1
Data =50

1.To display list
2. For insertion at starting
3. For insertion at end
4. For insertion at any position
5. For deletion of first element
6. For deletion of last element
7. For deletion of element at any position
8. To exit

Enter choice:
3
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: 3

Class / Branch: SE/DS

Subject:DSL

Name of Instructor:Prof. Rajashri Chaudhari

Name of Student:Arya Patil

Student ID:21107006

Date Of Performance:28/09/22

Date Of Submission: 28/09/22

---

### Experiment No. 7

**Aim:-** To implement circular linked list ADT.

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
struct node{
```

```
int data;
```

```
struct node *next;
```

```
};
```

```
struct node *head;
```

```
void begininsert();
```

```
void lastinsert();
```

```
void begin_delete();
```

```
void last_delete();
```



```
void display();

void search();

void main()

{

    int choice=0;

    while(choice!=7)

    {

        printf("\n*****Main menu*****\n");

        printf("\nChoose one option from the following list...\n");

        printf("\n===== \n");

        printf("\n 1.Insert in begining\n 2.Insert at last\n 3.Delete from begining\n\n 4.Delete from last\n 5.Search for an element\n 6.Show\n 7.Exit\n");

        printf("\nEnter your choice\n");

        scanf("\n%d",&choice);

        switch(choice)

        {

            case 1:

                beginsert();

                break;

            case 2:
```



---

```
lastinsert();  
  
break;  
  
case 3:  
  
begin_delete();  
  
break;  
  
case 4:  
  
last_delete();  
  
break;  
  
case 5:  
  
search();  
  
break;  
  
case 6:  
  
display();  
  
break;  
  
case 7:  
  
exit(0);  
  
break;  
  
default:  
  
printf("Please enter valid choice...");  
  
}
```





}

}

void begininsert()

{

struct node \*ptr,\*temp;

int item;

ptr=(struct node \*)malloc(sizeof(struct node));

if(ptr==NULL)

{

printf("\nOverflow");

}

else

{

printf("\nEnter the node data:");

scanf("%d",&item);

ptr->data=item;

if(head==NULL)

{

head=ptr;

ptr->next=head;





```
}  
  
else  
  
{  
  
temp=head;  
while(temp->next!=head)  
    temp=temp->next;  
ptr->next=head;  
temp->next=ptr;  
head=ptr;  
}  
printf("\nNode inserted\n");  
}  
  
}  
  
void lastinsert()  
{  
  
    struct node *ptr,*temp;  
  
    int item;  
  
    ptr=(struct node *)malloc(sizeof(struct node));  
  
    if(ptr==NULL)  
    {
```



---

```
printf("\nOverflow\n");  
  
}  
  
else  
  
{  
  
printf("\nEnter data:");  
  
scanf("%d",&item);  
  
ptr->data=item;  
  
if(head==NULL)  
{  
  
head=ptr;  
  
ptr->next=head;  
  
}  
  
else  
  
{  
  
temp=head;  
  
while(temp->next!=head)  
  
{  
  
temp=temp->next;  
  
}  
  
temp->next=ptr;  
  
ptr->next=head;
```



```
}  
  
printf("\nNode inserted\n");  
  
}  
  
}
```

```
void begin_delete()  
{  
    struct node *ptr;  
    if(head==NULL)  
    {  
        printf("\nUnderflow");  
    }  
    else if(head->next==head)  
    {  
        head=NULL;  
        free(head);  
        printf("\nNode deleted\n");  
    }  
    else  
    {  
        ptr=head;
```



---

```
while(ptr->next!=head)

    ptr=ptr->next;

ptr->next=head->next;

free(head);

head=ptr->next;

printf("\nNode deleted\n");

}

}
```

```
void last_delete()

{

    struct node *ptr,*preptr;

    if(head==NULL)

    {

        printf("\nunderflow");

    }

    else if(head->next==head)

    {

        head=NULL;

        free(head);

        printf("\nNode deleted\n");

    }

}
```



```
}  
  
else  
  
{  
  
    ptr=head;  
    while(ptr->next!=head)  
    {  
        preptr=ptr;  
        ptr=ptr->next;  
    }  
    preptr->next=ptr->next;  
    free(ptr);  
    printf("\nNode deleted\n");  
}  
}
```

```
void search()  
{  
    struct node *ptr;  
    int item,i=0,flag=1;  
    ptr=head;  
    if(ptr==NULL)
```



---

```
{  
  
    printf("\nEmpty list\n");  
  
}  
  
else  
  
{  
  
    printf("\nEnter item which you want to search\n");  
  
    scanf("%d",&item);  
  
    if(head->data==item)  
    {  
  
        printf("item found at location %d",i+1);  
  
        flag=0;  
  
    }  
  
    else  
  
    {  
  
        while(ptr->next!=head)  
        {  
  
            if(ptr->data==item)  
            {  
  
                printf("item found at location %d",i+1);  
  
                break;  
  
            }  
  
        }  
  
    }  
  
}
```



---

else

{

flag=1;

}

i++;

ptr=ptr->next;

}

}

if(flag!=0)

{

printf("Item not found\n");

}

}

}

void display()

{

struct node \*ptr;

ptr=head;

if(head==NULL)

{



---

```
printf("\nNothing to print");  
  
}  
  
else  
  
{  
  
printf("\n Printing values...\n");  
  
while(ptr->next!=head)  
  
{  
  
printf("%d\n",ptr->data);  
  
ptr=ptr->next;  
  
}  
  
printf("%d\n",ptr->data);  
  
}  
  
}
```

**Output:-**





```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ gcc -o exp7 exp7.c
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ ./exp7
```

```
*****Main menu*****
```

```
Choose one option from the following list...
```

```
=====
```

- 1.Insert in begining
- 2.Insert at last
- 3.Delete from begining
- 4.Delete from last
- 5.Search for an element
- 6.Show
- 7.Exit

```
Enter your choice
```

```
1
```

```
Enter the node data:10
```

```
Node inserted
```

```
*****Main menu*****
```

```
Choose one option from the following list...
```

```
=====
```

- 1.Insert in begining
- 2.Insert at last
- 3.Delete from begining
- 4.Delete from last
- 5.Search for an element
- 6.Show
- 7.Exit

```
Enter your choice
```

```
1
```

```
Enter the node data:20
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

```
Enter the node data:20
Node inserted
*****Main menu*****
Choose one option from the following list...
=====
1.Insert in begining
2.Insert at last
3.Delete from begining
4.Delete from last
5.Search for an element
6.Show
7.Exit
Enter your choice
1
Enter the node data:30
Node inserted
*****Main menu*****
Choose one option from the following list...
=====
1.Insert in begining
2.Insert at last
3.Delete from begining
4.Delete from last
5.Search for an element
6.Show
7.Exit
Enter your choice
2
Enter data:40
```



```
Node inserted

*****Main menu*****

Choose one option from the following list...

=====

1.Insert in begining
2.Insert at last
3.Delete from begining
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice
6

Printing values...
30
20
10
40

*****Main menu*****

Choose one option from the following list...

=====

1.Insert in begining
2.Insert at last
3.Delete from begining
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice
1

Enter the node data:50
```



```
Enter the node data:50
Node inserted
*****Main menu*****
Choose one option from the following list...
=====
1.Insert in begining
2.Insert at last
3.Delete from begining
4.Delete from last
5.Search for an element
6.Show
7.Exit
Enter your choice
3
Node deleted
*****Main menu*****
Choose one option from the following list...
=====
1.Insert in begining
2.Insert at last
3.Delete from begining
4.Delete from last
5.Search for an element
6.Show
7.Exit
Enter your choice
4
Node deleted
```



```
*****Main menu*****
```

```
Choose one option from the following list...
```

```
=====
```

- 1.Insert in begining
- 2.Insert at last
- 3.Delete from begining
- 4.Delete from last
- 5.Search for an element
- 6.Show
- 7.Exit

```
Enter your choice
```

```
6
```

```
Printing values...
```

```
30
```

```
20
```

```
10
```

```
*****Main menu*****
```

```
Choose one option from the following list...
```

```
=====
```

- 1.Insert in begining
- 2.Insert at last
- 3.Delete from begining
- 4.Delete from last
- 5.Search for an element
- 6.Show
- 7.Exit

```
Enter your choice
```

```
5
```

```
Enter item which you want to search
```

```
20
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

```
Enter item which you want to search
20
item found at location 2Item not found

*****Main menu*****

Choose one option from the following list...

=====

1.Insert in begining
2.Insert at last
3.Delete from begining
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice
7
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$
```





PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

**Academic Year:** 2022-2023

**Semester:** III

**Class / Branch:** SE Data Science

**Subject:** Data Structure Lab

**Name of Instructor:** Prof. Rajashri Chaudhari

**Name of Student:** Arya Patil

**Student ID:** 21107009

**Date Of Performance:** 12/10/2022

**Date Of Submission:** 12/10/2022

---

### Experiment No. 8

**Aim:-** Write a C program to implement stack ADT using linked list.

**Program:**

```
#include<stdio.h>

#include<stdlib.h>

#include<malloc.h>

#define size 5

typedef struct node

{

int data;

struct node*next;

}

node;

int main()

{
```



---

```
node*start=NULL,*temp;

int top=0,ch,n;

for(;;)

{

printf("Stack using linked list\n");

printf("1. Push\n");

printf("2. Pop\n");

printf("3. Display\n");

printf("4. Exit\n");

printf("Enter choice: ");

scanf("%d",&ch);

switch(ch)

{

case 1:

if(top==size)

{

printf("Stack is full");

}

else

{
```





---

```
printf("Enter a number: ");

scanf("%d",&n);

temp=(node*)malloc(sizeof(node));

temp->data=n;

temp->next=NULL;

if(start==NULL)

{

start=temp;

}

else

{

temp->next=start;

start=temp;

}

top++;

}

break;

case 2:

if(start==NULL)

{
```



---

```
printf("Stack is empty");  
  
}  
  
else  
  
{  
  
printf("Number Popped = %d",start->data);  
  
temp=start;  
  
start=start->next;  
  
free(temp);  
  
top--;  
  
}  
  
break;  
  
  
case 3:  
  
if(start==NULL)  
  
{  
  
printf("Stack is empty");  
  
}  
  
else  
  
{  
  
temp=start;  
  
while(temp!=NULL)
```



```
{  
  
printf("%d\n",temp->data);  
  
temp=temp->next;  
  
}  
  
}  
  
break;  
  
  
case 4:  
  
exit(0);  
  
break;  
  
default:  
  
printf("Wrong Choice");  
  
}  
  
}  
  
return 0;  
  
}
```



## Output:-

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ gcc -o exp8 exp8.c
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ ./exp8
Stack using linked list
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 1
Enter a number: 23
Stack using linked list
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 1
Enter a number: 78
Stack using linked list
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 1
Enter a number: 34
Stack using linked list
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 3
34
78
23
```



```
Stack using linked list
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 2
Number Popped = 34
Stack using linked list
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 3
78
23
Stack using linked list
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 4
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: SE Data Science

Subject: Data Structure Lab

Name of Instructor: Prof. Rajashri Chaudhari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 12/10/2022

Date Of Submission: 12/10/2022

---

### Experiment No. 9

**Aim:-** To implement binary search tree ADT using linked list

**Program:**

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct btnode
```

```
{
```

```
    int value;
```

```
    struct btnode *l;
```

```
    struct btnode *r;
```

```
}*root = NULL, *temp = NULL, *t2, *t1;
```

```
void delete1();
```

```
void insert();
```



---

```
void delete();

void inorder(struct btnode *t);

void create();

void search(struct btnode *t);

void preorder(struct btnode *t);

void postorder(struct btnode *t);

void search1(struct btnode *t,int data);

int smallest(struct btnode *t);

int largest(struct btnode *t);


int flag = 1;


void main()
{
    int ch;


    printf("\nOPERATIONS ---");

    printf("\n1 - Insert an element into tree\n");

    printf("\n2 - Delete an element from the tree\n");

    printf("\n3 - Inorder Traversal\n");

    printf("\n4 - Preorder Traversal\n");
```



---

```
printf("5 - Postorder Traversal\n");
```

```
printf("6 - Exit\n");
```

```
while(1)
```

```
{
```

```
    printf("\nEnter your choice : ");
```

```
    scanf("%d", &ch);
```

```
    switch (ch)
```

```
    {
```

```
        case 1:
```

```
            insert();
```

```
            break;
```

```
        case 2:
```

```
            delete();
```

```
            break;
```

```
        case 3:
```

```
            inorder(root);
```

```
            break;
```

```
        case 4:
```

```
            preorder(root);
```

```
            break;
```

```
        case 5:
```





---

```
    postorder(root);

    break;

case 6:

    exit(0);

default :

    printf("Wrong choice, Please enter correct choice ");

    break;

}

}

}

/* To insert a node in the tree */

void insert()

{

    create();

    if (root == NULL)

        root = temp;

    else

        search(root);

}
```



---

/\* To create a node \*/

void create()

{

int data;

printf("Enter data of node to be inserted : ");

scanf("%d", &data);

temp = (struct btnode \*)malloc(1\*sizeof(struct btnode));

temp->value = data;

temp->l = temp->r = NULL;

}

/\* Function to search the appropriate position to insert the new node \*/

void search(struct btnode \*t)

{

if ((temp->value > t->value) && (t->r != NULL)) /\* value more than root node  
value insert at right \*/

search(t->r);

else if ((temp->value > t->value) && (t->r == NULL))

t->r = temp;



---

```
else if ((temp->value < t->value) && (t->l != NULL)) /* value less than root node
value insert at left */
```

```
search(t->l);
```

```
else if ((temp->value < t->value) && (t->l == NULL))
```

```
t->l = temp;
```

```
}
```

```
/* recursive function to perform inorder traversal of tree */
```

```
void inorder(struct btnode *t)
```

```
{
```

```
if (root == NULL)
```

```
{
```

```
printf("No elements in a tree to display");
```

```
return;
```

```
}
```

```
if (t->l != NULL)
```

```
inorder(t->l);
```

```
printf("%d -> ", t->value);
```

```
if (t->r != NULL)
```

```
inorder(t->r);
```

```
}
```



```
/* To check for the deleted node */
```

```
void delete()
```

```
{
```

```
    int data;
```

```
    if (root == NULL)
```

```
    {
```

```
        printf("No elements in a tree to delete");
```

```
        return;
```

```
    }
```

```
    printf("Enter the data to be deleted : ");
```

```
    scanf("%d", &data);
```

```
    t1 = root;
```

```
    t2 = root;
```

```
    search1(root, data);
```

```
}
```

```
/* To find the preorder traversal */
```

```
void preorder(struct btnode *t)
```

```
{
```



---

```
if (root == NULL)

{

    printf("No elements in a tree to display");

    return;

}

printf("%d -> ", t->value);

if (t->l != NULL)

    preorder(t->l);

if (t->r != NULL)

    preorder(t->r);

}
```

/\* To find the postorder traversal \*/

```
void postorder(struct btnode *t)

{

    if (root == NULL)

    {

        printf("No elements in a tree to display ");

        return;

    }

    if (t->l != NULL)
```



---

```
    postorder(t->l);

    if (t->r != NULL)

        postorder(t->r);

    printf("%d -> ", t->value);

}

/* Search for the appropriate position to insert the new node */

void search1(struct bnode *t, int data)

{

    if ((data>t->value))

    {

        t1 = t;

        search1(t->r, data);

    }

    else if ((data < t->value))

    {

        t1 = t;

        search1(t->l, data);

    }

    else if ((data==t->value))

    {
```



---

```
delete1(t);

}

}

/* To delete a node */

void delete1(struct btnode *t)
{
    int k;

    /* To delete leaf node */
    if ((t->l == NULL) && (t->r == NULL))
    {
        if (t1->l == t)
        {
            t1->l = NULL;
        }
        else
        {
            t1->r = NULL;
        }
        t = NULL;
    }
}
```



---

```
free(t);

return;

}

/* To delete node having one left hand child */

else if ((t->r == NULL))

{

    if (t1 == t)

    {

        root = t->l;

        t1 = root;

    }

    else if (t1->l == t)

    {

        t1->l = t->l;

    }

    else

    {

        t1->r = t->l;

    }

}
```





---

```
t = NULL;  
  
free(t);  
  
return;  
  
}  
  
/* To delete node having right hand child */  
  
else if (t->l == NULL)  
{  
    if (t1 == t)  
    {  
        root = t->r;  
        t1 = root;  
    }  
    else if (t1->r == t)  
        t1->r = t->r;  
    else  
        t1->l = t->r;  
    t == NULL;  
    free(t);  
    return;  
}
```



```
/* To delete node having two child */  
else if ((t->l != NULL) && (t->r != NULL))  
{  
    t2 = root;  
    if (t->r != NULL)  
    {  
        k = smallest(t->r);  
        flag = 1;  
    }  
    else  
    {  
        k = largest(t->l);  
        flag = 2;  
    }  
    search1(root, k);  
    t->value = k;  
}  
  
}  
  
/* To find the smallest element in the right sub tree */
```



---

```
int smallest(struct bnode *t)
```

```
{
    t2 = t;
    if (t->l != NULL)
    {
        t2 = t;
        return(smallest(t->l));
    }
    else
        return (t->value);
}
```

```
/* To find the largest element in the left sub tree */
```

```
int largest(struct bnode *t)
{
    if (t->r != NULL)
    {
        t2 = t;
        return(largest(t->r));
    }
    else
```



```
return(t->value);
```

```
}
```

**Output:-**

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ gcc -o exp9 exp9.c
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ ./exp9

OPERATIONS ---
1 - Insert an element into tree
2 - Delete an element from the tree
3 - Inorder Traversal
4 - Preorder Traversal
5 - Postorder Traversal
6 - Exit

Enter your choice : 1
Enter data of node to be inserted : 35

Enter your choice : 1
Enter data of node to be inserted : 24

Enter your choice : 1
Enter data of node to be inserted : 63

Enter your choice : 1
Enter data of node to be inserted : 15

Enter your choice : 1
Enter data of node to be inserted : 30

Enter your choice : 1
Enter data of node to be inserted : 55

Enter your choice : 3
15 -> 24 -> 30 -> 35 -> 55 -> 63 ->
Enter your choice : 4
35 -> 24 -> 15 -> 30 -> 63 -> 55 ->
Enter your choice : 5
15 -> 30 -> 24 -> 55 -> 63 -> 35 ->
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

```
Enter your choice : 2
Enter the data to be deleted : 30

Enter your choice : 3
15 -> 24 -> 35 -> 55 -> 63 ->
Enter your choice : 4
35 -> 24 -> 15 -> 63 -> 55 ->
Enter your choice : 5
15 -> 24 -> 55 -> 63 -> 35 ->
Enter your choice : 6
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: SE Data Science

Subject: Data Structure Lab

Name of Instructor: Prof. Rajashri Chaudhari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 12/10/2022

Date Of Submission: 12/10/2022

---

### Experiment No. 10

**Aim:-** Write a C program to implement graph traversal techniques Depth First Search and Breadth First Search.

**Program:**

```
#include<stdio.h>
```

```
int q[20],top=-1,front=-1,rear=-1,a[20][20],vis[20],stack[20];
```

```
int delete();
```

```
void add(int item);
```

```
void bfs(int s,int n);
```

```
void dfs(int s,int n);
```

```
void push(int item);
```

```
int pop();
```

```
void main()
```

```
{
```



---

```
int n,i,s,ch,j;

char c,dummy;

printf("ENTER THE NUMBER VERTICES ");

scanf("%d",&n);

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

{

printf("ENTER 1 IF %d HAS A NODE WITH %d ELSE 0 ",i,j);

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

}

}

printf("THE ADJACENCY MATRIX IS\n");

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

{

printf(" %d",a[i][j]);

}

printf("\n");

}
```



```
do
{
for(i=1;i<=n;i++)
vis[i]=0;
printf("\nMENU");
printf("\n1.B.F.S");
printf("\n2.D.F.S");
printf("\nENTER YOUR CHOICE");
scanf("%d",&ch);
printf("ENTER THE SOURCE VERTEX :");
scanf("%d",&s);

switch(ch)
{
case 1: bfs(s,n);
break;
case 2:
dfs(s,n);
break;
}
```





---

```
printf("DO U WANT TO CONTINUE(Y/N) ? ");
```

```
scanf("%c",&dummy);
```

```
scanf("%c",&c);
```

```
}while((c=='y')||(c=='Y'));
```

```
}
```

```
//*****BFS(breadth-first search) code*****//
```

```
void bfs(int s,int n)
```

```
{
```

```
int p,i;
```

```
add(s);
```

```
vis[s]=1;
```

```
p=delete();
```

```
if(p!=0)
```

```
printf(" %d",p);
```

```
while(p!=0)
```

```
{
```

```
for(i=1;i<=n;i++)
```

```
if((a[p][i]!=0)&&(vis[i]==0))
```

```
{
```



```
add(i);

vis[i]=1;

}

p=delete();

if(p!=0)

printf(" %d ",p);

}

for(i=1;i<=n;i++)

if(vis[i]==0)

bfs(i,n);

}


void add(int item)

{

if(rear==19)

printf("QUEUE FULL");

else

{

if(rear==-1)

{
```



---

```
q[++rear]=item;

front++;

}

else

q[++rear]=item;

}

}

int delete()

{

int k;

if((front>rear)|| (front==-1))

return(0);

else

{

k=q[front++];

return(k);

}

}
```

```
//*****DFS(depth-first search) code*****//
```



---

```
void dfs(int s,int n)

{

int i,k;

push(s);

vis[s]=1;

k=pop();

if(k!=0)

printf(" %d ",k);

while(k!=0)

{

for(i=1;i<=n;i++)

if((a[k][i]!=0)&&(vis[i]==0))

{

push(i);

vis[i]=1;

}

k=pop();

if(k!=0)

printf(" %d ",k);

}

for(i=1;i<=n;i++)
```



---

```
if(vis[i]==0)

dfs(i,n);

}

void push(int item)

{

if(top==19)

printf("Stack overflow ");

else

stack[++top]=item;

}

int pop()

{

int k;

if(top== -1)

return(0);

else

{

k=stack[top--];

return(k);

}

}
```

**Output:-**

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ gcc -o exp10 exp10.c
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ ./exp10
ENTER THE NUMBER VERTICES 3
ENTER 1 IF 1 HAS A NODE WITH 1 ELSE 0 1
ENTER 1 IF 1 HAS A NODE WITH 2 ELSE 0 2
ENTER 1 IF 1 HAS A NODE WITH 3 ELSE 0 3
ENTER 1 IF 2 HAS A NODE WITH 1 ELSE 0 4
ENTER 1 IF 2 HAS A NODE WITH 2 ELSE 0 5
ENTER 1 IF 2 HAS A NODE WITH 3 ELSE 0 6
ENTER 1 IF 3 HAS A NODE WITH 1 ELSE 0 7
ENTER 1 IF 3 HAS A NODE WITH 2 ELSE 0 8
ENTER 1 IF 3 HAS A NODE WITH 3 ELSE 0 9
THE ADJACENCY MATRIX IS
 1 2 3
 4 5 6
 7 8 9

MENU
1.B.F.S
2.D.F.S
ENTER YOUR CHOICE1
ENTER THE SOURCE VERTEX :5
 5 1 2 3 DO U WANT TO CONTINUE(Y/N) ? y

MENU
1.B.F.S
2.D.F.S
ENTER YOUR CHOICE2
ENTER THE SOURCE VERTEX :5
 5 1 3 2 DO U WANT TO CONTINUE(Y/N) ? n
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: SE Data Science

Subject: Data Structure Lab

Name of Instructor: Prof. Rajashri Chaudhari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 27/07/2022

Date Of Submission: 01/08/2022

---

### Experiment No. 11

**Aim:-** Write a C program to implement linear search and binary search techniques.

#### Program 11 (a): Linear Search

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int flag=0;
```

```
int k,n,i;
```

```
int a[10];
```

```
printf("Enter the size of the array\n");
```

```
scanf("%d",&n);
```

```
printf("Enter the elements\n");
```

```
for(i=0;i<n;i++)
```

```
{
```





---

```
scanf("%d",&a[i]);
```

```
}
```

```
printf("Enter the number to be searched\n");
```

```
scanf("%d",&k);
```

```
for(i=0;i<n;i++)
```

```
{
```

```
if(k==a[i])
```

```
{
```

```
flag=1;
```

```
break;
```

```
}
```

```
}
```

```
if(flag==1)
```

```
{
```

```
printf("Number found at index %d",i);
```

```
}
```

```
else
```

```
{
```



---

```
printf("Number not found");
```

```
}
```

```
}
```

## Program 11 (b): Binary Search

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int i,l,u,mid,n,k,a[100];
```

```
printf("Enter the size of the array\n");
```

```
scanf("%d",&n);
```

```
printf("Enter %d integers in sorted array\n",n);
```

```
for(i=0;i<n;i++)
```

```
{
```

```
scanf("%d",&a[i]);
```

```
}
```

```
printf("Enter the number to be searched\n");
```

```
scanf("%d",&k);
```

```
l=0;
```



---

u=n-1;

mid=(l+u)/2;

while(l<=u)

{

if(a[mid]<k)

l=mid+1;

else if(a[mid]==k)

{

printf("%d found at location %d\n",k,mid+1);

break;

}

else

u=mid-1;

mid=(l+u)/2;

}

if(l>u)

{

printf(" Not found! %d is not present in the list\n",k);

}

}



## Output:-

### Program 11(a)

```
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ gcc -o test test.c
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ ./test
Enter the size of an array
5
enter the elements
1
2
3
4
5
Enter the number to be searched
4
Number found at index 3apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ |
```

### Program 11 (b)

```
Enter the number of elements
5
Enter 5 integers in sorted array
10
20
30
40
50
Enter the number to find
40
40 found at location 4
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ ./exp11
Enter the number of elements
5
Enter 5 integers in sorted array
10
20
30
40
50
Enter the number to find
11
Not found! 11 is not present in the list
apsit@apsit-HP-ProDesk-600-G4-PCI-MT:~$ |
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: SE-Data Science

Subject: DSL

Name of Instructor: Prof.Rajashri Chaudari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 20/10/2022

Date Of Submission: 21/10/2022

---

### Experiment No. 12

**Aim:-** To develop an application for shopping cart and display mostly viewed or purchased item and search for an item.

**Code:**

```
#include <stdio.h>
```

```
#include<string.h>
```

```
#include <stdlib.h>
```

```
#include <malloc.h>
```

```
#define size 5
```

```
// Define node structure
```

```
typedef struct node
```

```
{
```

```
    char data[10];
```

```
    struct node *next;
```

```
} node;
```

```
int main()
```



---

```
{  
  
node *start=NULL,*temp;  
  
int top=0,ch,i,c=0;  
  
char n[10];  
  
char ele[10];  
  
for(;;) // An infinite loop  
{  
  
printf("1. Insert item\n");  
  
printf("2. Recently added item\n");  
  
printf("3. View all items in cart\n");  
  
printf("4. Search an item\n");  
  
printf("5. Exit\n");  
  
printf("\nEnter Choice: ");  
  
scanf("%d",&ch);  
  
switch(ch)  
{  
  
case 1:  
  
if(top==size)  
{  
  
printf("Cart is full");  
  
// pause the loop to see the message
```



```
}  
  
else  
  
{  
  
printf("Enter new item ");  
  
//for(i=0;i<10;i++)  
  
scanf("%s",n);  
  
//Create a new node  
  
temp=(node*)malloc(sizeof(node));  
  
for(i=0;i<strlen(n);i++)  
  
temp->data[i]=n[i];  
  
temp->next=NULL;  
  
if(start==NULL)  
  
{  
  
start=temp;  
  
}  
  
else  
  
{  
  
// insert the new node before the first node  
  
temp->next=start;  
  
start=temp;  
  
}
```



---

```
top++;  
  
}  
  
break;  
  
case 2:  
  
if(start==NULL)  
{  
  
printf("No items selected in cart.");  
  
// pause the loop to see the message  
  
}  
  
else  
  
{  
  
//while(temp!=NULL)  
  
// {  
  
  
  
  
// temp=temp->next;  
  
//}  
  
printf("\n%s",start);  
  
}  
  
break;  
  
case 3:  
  
if(start==NULL)
```





---

```
{  
  
printf("No items in cart");  
  
// pause the loop to see the message  
  
}  
  
else  
  
{  
  
temp=start; // start from 1st node  
  
// display the nodes on the screen  
  
while(temp!=NULL)  
  
{  
  
printf("\n%s",temp->data);  
  
temp=temp->next;  
  
}  
  
// pause the loop to see the nodes  
  
}  
  
break;  
  
case 4:  
  
if (start == NULL)  
  
printf("\nNo items in cart\n");  
  
else {  
  
temp = start;
```



---

```
printf("Enter the name of item which you want to search:");
```

```
scanf("%s",ele);
```

```
//while (temp->data != ele)
```

```
while(temp->next!=NULL)
```

```
{
```

```
c=strcmp(temp->data,ele);
```

```
if(c==0)
```

```
{
```

```
printf("Item in cart = %s\n", temp->data);
```

```
break;
```

```
}
```

```
else
```

```
temp = temp->next;
```

```
}
```

```
//if(c==1)
```

```
//printf("Item in cart = %s\n", temp->data);
```

```
//else
```

```
//printf("\n not found...");
```

```
}
```

```
break;
```



case 5:

```
exit(0);
```

```
break;
```

default:

```
printf("Wrong Choice");
```

```
// pause the loop to see the message
```

```
}
```

```
}
```

```
return 0;
```

```
}
```

## Output:-

```
1. Insert item
2. Recently added item
3. View all items in cart
4. Search an item
5. Exit
```

```
Enter Choice: 1
Enter new item maggi
```

```
1. Insert item
2. Recently added item
3. View all items in cart
4. Search an item
5. Exit
```

```
Enter Choice: 1
Enter new item juice
```

```
1. Insert item
2. Recently added item
3. View all items in cart
4. Search an item
5. Exit
```

```
Enter Choice: 1
Enter new item ketchup
```

```
1. Insert item
2. Recently added item
3. View all items in cart
4. Search an item
5. Exit
```

```
Enter Choice: 1
Enter new item bread
```



```
1. Insert item
2. Recently added item
3. View all items in cart
4. Search an item
5. Exit
```

Enter Choice: 3

```
bread
ketchup
juice
maggi
```

```
1. Insert item
2. Recently added item
3. View all items in cart
4. Search an item
5. Exit
```

Enter Choice: 2

```
bread
```

```
1. Insert item
2. Recently added item
3. View all items in cart
4. Search an item
5. Exit
```

Enter Choice: 4

Enter the name of item which you want to search:juice

Item in cart = juice

```
1. Insert item
2. Recently added item
3. View all items in cart
4. Search an item
5. Exit
```

Enter Choice: 4

Enter the name of item which you want to search:bread

Item in cart = bread



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

```
1. Insert item
2. Recently added item
3. View all items in cart
4. Search an item
5. Exit
```

```
Enter Choice: 5
```



PARSHVANATH CHARITABLE TRUST'S

# A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering  
Data Science

---

## Department of Computer Science and Engineering Data Science

Academic Year: 2022-2023

Semester: III

Class / Branch: SE Data Science

Subject: Data Structure Lab

Name of Instructor: Prof. Rajashri Chaudhari

Name of Student: Arya Patil

Student ID: 21107009

Date Of Performance: 11/10/2022

Date Of Submission: 11/10/2022

---

### Experiment No. 13

**Aim:-** To implement quick sort and merge sort algorithm

#### Program13(a) : Quick Sort

```
#include<stdio.h>
```

```
void swap(int *a,int *b)
```

```
{
```

```
int t=*a;
```

```
*a=*b;
```

```
*b=t;
```

```
}
```

```
int partition(int array[],int low,int high)
```

```
{
```

```
int pivot=array[high];
```

```
int i=(low-1);
```



---

```
for(int j=low;j<high;j++)  
  
{  
  
if(array[j]<=pivot)  
  
{  
  
i++;  
  
swap(&array[i],&array[j]);  
  
}  
  
}  
  
swap(&array[i+1],&array[high]);  
return (i+1);  
  
}  
  
void quicksort(int array[],int low,int high)  
  
{  
  
if (low<high)  
  
{  
  
int pi=partition(array,low,high);  
  
quicksort(array,low,pi-1);  
  
quicksort(array,pi+1,high);  
  
}
```



---

}

```
int main()
```

```
{
```

```
int data[15],n=0,i=0;
```

```
printf("\n Enter the size of an array:");
```

```
scanf("%d",&n);
```

```
printf("Enter array elements: ");
```

```
for(i=0;i<n;i++)
```

```
scanf("%d",&data[i]);
```

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

```
for(i=0;i<n;++i)
```

```
{
```

```
printf("%d",data[i]);
```

```
}
```

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

```
quicksort(data,0,n-1);
```

```
printf("Sorted array in ascending order: \n");
```

```
for(i=0;i<n;++i)
```

```
{
```





---

```
printf("%d",data[i]);
```

```
}
```

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

```
}
```

## Program13(b) : Merge Sort

```
#include<stdio.h>
```

```
int a[5];
```

```
int b[4];
```

```
void merging(int low,int mid,int high)
```

```
{
```

```
int l1,l2,i;
```

```
for(l1=low, l2=mid+1, i=low; l1<=mid && l2<=high; i++)
```

```
{
```

```
if(a[l1]<=a[l2])
```

```
{
```

```
b[i]=a[l1++];
```

```
}
```

```
else
```



```
{  
  
b[i]=a[l2++];  
  
}  
  
}  
  
while(l1<=mid)  
  
b[i++]=a[l1++];  
  
while(l2<=high)  
  
b[i++]=a[l2++];  
  
  
for(i=low;i<=high;i++)  
  
{  
  
a[i]=b[i];  
  
}  
  
}  
  
void sort(int low,int high)  
  
{  
  
int mid;  
  
if(low<high)  
  
{  
  
mid=(low+high)/2;  
  
sort(low,mid);
```



---

```
sort(mid+1,high);

merging(low,mid,high);

}

else

{

return;

}

}


int main()

{

int i,n;

printf("\n Enter the size of an array: ");

scanf("%d",&n);

printf("\n enter array elements:");

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

{

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

}

printf("List before sorting\n");

for(i=0;i<n;i++)
```



```
printf("%d\t",a[i]);  
  
sort(0,n-1);  
  
printf("\n List after sorting\n");  
  
for(i=0;i<n;i++)  
  
printf("%d\t",a[i]);  
  
}
```

**Output:-**

**Output 13 (a)**

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ gcc -o exp13 exp13.c  
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ ./exp13  
  
Enter the size of an array:5  
Enter array elements: 6  
12  
7  
2  
9  
Unsorted Array  
612729  
Sorted array in ascending order:  
267912  
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$
```



## Output 13 (b)

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ gcc -o exp13 exp13.c
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ ./exp13
```

```
Enter the size of an array: 4
```

```
enter array elements:12
```

```
33
```

```
53
```

```
11
```

```
List before sorting
```

```
12      33      53      11
```

```
List after sorting
```

```
11      12      33      53
```

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$
```