Question 1:-Write an algorithm to search an element in single linked list .

Answers:- Let x be the element to search

```
void SEARCH(x)
Begin
found =0
current =head
        while (current !=null)
         {
                if(current ->info=x
                {
                found=1
                break
                }
                current=current->next
        }
if(found=1)
print ”Element    found”
else
print “Not found"
End.
```

--

Question 2:-write an algorithm to insert an element in the single link list.

Answers :- Algorithm

```
Begin
Step 1   Read the element into x
Step 2   Create an temp node in memory as follows
         temp=(struct node *)size of (node)
Step 3   Set the values in temp node as follows
                temp-> info =x
                temp->next=null
Step 4   Search the element    after which node will be inserted
                current =SEARCH()
Step 5   insert temp node offer    current node as follows
                temp->next =current -> next
                current->next=temp
End.
```

--

Q 3:- write an algorithm to Create a single linked list .

Ans:- Algorithm has three parts

(a)Declaration

(b)initial Condition

(c)Steps for Algorithms

(a)Declaration

```
        struct node{
        int info;
        struct node * next;
        } *head,*current,*temp
```

(b)initial Condition

```
        head=null
        temp=nul
        current=null
```

(c)Steps for Algorithms

```
Begin
Step 1 Read the element into x
Step 2 Create a temp node in the memory
        temp =(struct node )sizeof (node)
Step 3 Assign the values in temp node as follows
        temp -> info =x
        temp ->next=null
Step 4 check whether head is null or not
if (head=null)
        {
        head=temp
        current=temp
        }
else
        {
          current ->next =temp
        current ->current ->next
        }
Step 5 follow step 1 to 4 to insert remaining element in the list.
End.
```

Q4 write an algorithm to traverse or print elements of a single linked list

Ans:-

```
void DISPLAY ()
Begin
        current=head
        while (current != null)
        {
        Print    "current -> info"
        current =current ->next
        }
End
```

Q 5:- write an algorithm to implement Queue

Ans:-

```
Assumption:-
                int max _size =10
                int queue[max_size]
                int front = -1
                int rear=    -1
```

INSERT(x)

```
Begin
if(rear =max_size-1)
print “queue is full”
else
{
rear=rear+1
queue[rear]=x
}
End.
```

int DELETE()

```
Begin
if (front = -1 AND rear= -1)
PRINT “Queue is empty ”
else
{
z=queue [front ]
front =front+1
}
return z
End.
```

--

Q6:- write an algorithm to push & pop of stack.

Ans:-

```
Assumption :-
            int max _size=10
            int stack[max_size]
            int top= -1
PUSH(x)
Begin
if (top=max_size-1)
PRiNT”stack is full”
  else
            {
            top=top+1
            stack[top]=x
            }
End.
int POP ()
Begin
if (top= -1)
PRINT” stack is empty”
else
            {
            z=stack[top]
            top=top – 1
            }
return z
End.
```
