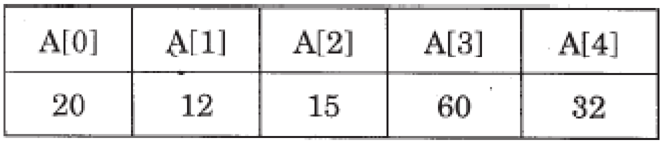
Linked List Stack & queue

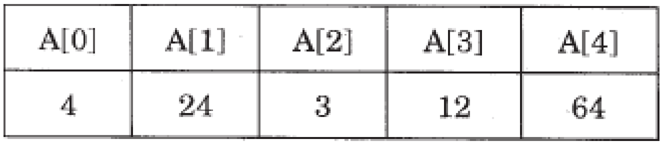
Q1. Write a function REASSIGNO in C++, which accepts an array of integers and its size as parameters and divide all those array elements by 5 which are divisible by 5 and multiply other array elements by 2.

Sample Input Data of the array



A[0]	A[1]	A[2]	A[3]	A[4]
20	12	15	60	32

Content of the array after calling REASSIGNO function



A[0]	A[1]	A[2]	A[3]	A[4]
4	24	3	12	64

Q2. Write a function in C++, whichaccepts an integer array and its size asarguments and swap the elements of everyeven location with its following oddlocation.

Example :

If an array of nine elements initially
contains the elements as
2,4,1,6,5,7,9,23,10
then the function should rearrange the
array as 4,2,6,1,7,5,23,9,10

Q3. Write a function in C++, which acceptsan integer array and its size as parametersand rearranges the array in reverse.

Example:

If an array of nine elements initially
contains the elements as 4, 2, 5, 1,6, 7, 8, 12, 10
Then the function should rearrange thearray as
10,12, 8, 7, 6, 1, 5, 2, 4

Q4. Write function in C++ which accepts an integer array and size as arguments and replaces elements having odd values with thrice its value and elements having even values with twice its value.

Example : if an array of five elements
initially contains elements as 3, 4, 5, 16, 9
The function should rearrange the content of the array as 9, 8, 75, 32,27

Q5. Write a function in C++ which accepts an integer array and its size as arguments and exchanges the values of first half side elements with the second half side elements of the array.

Example :

If an array of 8 elements initial content as 2, 4, 1, 6, 7, 9, 23, 10
The function should rearrange array as 7, 9, 23, 10, 2, 4, 1, 6

Q6. Write a Function to Search for an element from Array A by Linear Search.

Q7. Write a Function to Search for an element from Array A by Binary Search.

Q8. Write a function to Sort the array A by Bubble Sort.

Q9. Write a function to Sort the array A by Selection Sort.

Q10. Write a function to Sort the array A by Insertion Sort.

Q11. What will be the status of the following list after fourth pass ofbubble sort and fourth pass of selection sort used for arrangingthe following elements in descending order?
14, 10, -12,9, 15, 35

Q12. A two dimensional array P[20] [50] is stored in the memory along the row with each of its element occupying 4 bytes, find theaddress of the element P[10] [30],if the element P[5] [5] is storedat the memory location 15000.

Q13. A two dimensional array ARR[50][20] is stored in the memoryalong the row with each of its elements occupying 4 bytes. Findthe address of the element ARR[30][10], if the elementARR[10] [5] is stored at the memory location 15000.

Q14. An array T [25][20] is stored along the row in the memory with each elementrequiring 2 bytes of storage. If the base address of array T is 42000, find out thelocation of T[l0][15]. Also, find the total number of elements present in this array.

Q15. An array A[20][30] is stored along the row in the memory with each element requiring 4 bytes of storage. If the base address of array A is 32000, find out the location of A[15][10]. Also, find the total number of elements present in this array.

Q16. Given an array A[10][12] whose base address is 10000. Calculate the memory location of A[2][5] if each element occupies 4 bytes and array is stored column-wise.

Q17. An array P[15][10] is stored along the column in the memory with each elementrequiring 4 bytes of storage. If the base address of array P is 14000, find out thelocation of P[8][5].

Q18. An array T[15][10] is stored along the row in the memory with each element requiring 8 bytes of storage. Ifthe base address of array T is 14000, find out the location of T[10][7].

Q19. An array T[20][10] is stored in the memory along the column with each of the element occupying 2 bytes, findout the memory location of T[10][5], if an element T[2][9] is stored at location 7600.

Q20. An array P[20] [50] is stored in the memory along the column with each of itselement occupying 4 bytes, find out the location of P[15][10], if P[0][0] isstored at 5200.

Q21. Convert the following infix expression to its equivalent postfix expression, showing the stack contents for each step ofconversion.
X / Y + U* (VW)

Q22.Convert the following infix expression to its equivalent Postfix expression, showing the stack contents for each step ofconversion.
U * V + R/ (ST)

Q23. Translate, following infix expression into its equivalent postfix expression:((A-B)*(D/E))/(F*G*H)

Q24. Translate, following infix expression into its equivalent postfix expression:A*(B+D)/E-F-(G+H/K)

Q25. Write the equivalent infix expression for10,3,*,7,1,-,*,23,+

Q26. Write the equivalent infix expression for a, b, AND, a, c, AND, OR.

Q27. Evaluate the infix expression.
P: 12 , 7 , 3 , - , / , 2 , 1 , 5 , + , * , + ,)

Q28. Give postfix form of the following expression A*(B+(C+D)*(E+F)/G)*H

Q29. Give postfix form expression for: NOT A OR NOT B AND NOT C

Q30. Consider the infix expressionQ : A+B * C ↑ (D/E)/F.
Translate Q into P, where P is the postfix equivalent expression of Q. what will be the result of Q if this expressionis evaluated for A, B, C, D, E, F as 2, 3, 2, 7, 2, 2 respectively.

Q31. Change the following infix expression into postfix expression.
(A+B) *C+D/E-F

Q32. Evaluate the following postfix expression. Show the status of stack after execution of each operationseparately;
F, T, NOT, AND, F, OR, T, AND

Q33. Evaluate the following postfix expression. Show the status of stack after execution of each operationseparately;
T, F, NOT, AND, T, OR, F, AND

Q34. Evaluate the following postfix expression. Show the status of stack after execution of each operation:
5, 2, *, 50, 5, /, 5, –, +

Q35. Evaluate the following postfix expression. Show the status if stack after execution of each operation;
60, 6, /, 5, 2, *, 5, –, +

Q36. Convert the following infix expression to its equivalent postfix expression. Showing stack contents for theconversion:
(X – Y / (Z + U)* V)

4 Marks Questions

Q1. Write the definition of a member function PUSH() in C++, to add a new book in a dynamic stack of BOOKS considering the following code is already included in the program:

```
     struct BOOKS
     {
     char ISBN[20], TITLE[80];
     BOOKS *Link;
     };
     class STACK
     {
     BOOKS *Top;
     public:
     STACK()
     {Top=NULL;}
     void PUSH();
     void POP();
     ~STACK();
     };
```

Q2. Write a complete program in c++ to implement a dynamically allocated Stack containing names of Countries.

Q3. Write a complete program in C++ toimplement a dynamically allocated Queuecontaining names of Cities.

Q4. Write a function QUEINS( ) in C++ to insertan element in a dynamicallyallocated Queue containing nodes of thefollowing given structure:

```
struct Node
{intPId ; //Product Id
charPname [20] ;
NODE *Next ;
} ;
```

Q5. Write a function QUEDEL( ) in C++ to display and delete an element from a dynamically allocated Queue containing nodes of the following given structure:

```
     struct NODE
     {
     intItemno;
     charItemname[20];
     NODE *Link;
     } ;
```

Q6. Write a function in C++ to **delete** a nodecontaining Book's information, from a**dynamically allocated Stack** of Booksimplemented with the help of the followingstructure.

```
     struct Book
     {
     intBNo ;
     charBName[20] ;
     Book *Next ;
     } ;
```

Q7. Write a function in C++ to perform Insertoperation in dynamically allocated Queuecontaining names of students.

```
     Struct NODE
     { char Name[20];
     NODE *Link;
     };
```

Q8. Write a function in C++ to perform aPUSH operation in a dynamically allocatedstack considering the following :

```
     struct Node
     {
     int X,Y ;
     Node *Link ;
     } ;
     class STACK
     {
     Node *Top ;
     public :
     STACK( )
     {Top = Null ;}
     void PUSH( ) ;
     void POP( ) ;
     ~STACK( ) ;
     } ;
```

Q9. Write a function QINSERT () in C++ to perform insert operation on a Linked Queue which contains client no and client name. Consider the following definition of NODE in the code of QINSERT()

```
struct Node
     {  longintCno;    //ClientNo
     charCname[20];  //CliehtName
     NODE*Next;

     };
```

Q10. Write a function PUSHBOOK() in C++ to perform insert operation on a Dynamic Stack, which contains Book_no and Book_Title. Consider the following definition of NODE, while writing your C++ code.

```
     struct NODE
      { intBook_No;
      charBook_Title[20];
       NODE *Next;
     };
```

LINKED LIST STACK AND Queue Solutions

```
Q1.  void REASSIGN (intArr[ ], int Size)
     {
     for (inti=0;i<Size;i++)
     if (Arr[i]%5==0)
     Arr[i]/=5;
     else
     Arr[i]*=2;
     }
```

```
Q2.  voidSwapArray(int A[ ], int N)
     {
     inti,j,temp;
     for(i=0;i<N-1;i+=2)
     {
     temp=A[i];
     A[i]=A[i+1];
     A[i+1]=temp;
          }
```

```
Q3.  void receive(int A[ ], int size)
     {
     int temp;
     for(i=0,j=size-1;i<size/2;i++,j--)
     {
     temp=A[i];
     A[i]=A[j];
     A[j]=temp;
     }
     }
```

```
Q4.  void manipulate (int a[ ],int size)
     {
     for (i=0;i<size;i++)
     {
     if (a[i]%2= =1)
     a[i]=a[i]*3;
     else
     a[i]=a[i]*2;
     cout<<a[i]<<',';
     }
     }
```

```
Q5.  .void exchange(int a[],int n)
     {    inti, mid,t,pos=0;  mid=n/2;
          if(n%2!=0)     pos=1;
          //swap
          for(i=0;i<mid;i++)
          {        t                    =       a[i]   ;
               a[i]                =    a[mid+pos+i);
               a[mid+pos+i]=      t;
          }
     }


Q6.  voidLsearch(int A[], int n, int Data)
     {     int I;
           for(I=0; I<n; I++)
           {        if(A[I]==Data)
                 {
                       cout<<"Data Found at : "<<I;
                 }
           }
           cout<<"Data Not Found in the array"<<endl;
     }

Q7.  intBsearchAsc(int A[], int n, int data)
     {      intMid,Lbound=0,Ubound=n-1,Found=0;
            while((Lbound<=Ubound) && !(Found))
            {  Mid=(Lbound+Ubound)/2;        //Searching The Item
                   if(data>A[Mid])
                          Lbound=Mid+1;
                   else if(data<A[Mid])
                          Ubound=Mid-1;
                   else
                          Found++;
            }       if(Found)
                   return(Mid+1);        //returning 1ocation, if present
                     else
                   return(-1);        //returning -1,if not present
     }

Q8.  void BSort(int A[], int n)
{
    intI,J,Temp;
    for(I=0;I<n-1;I++) //sorting
   {
       for(J=0;J<(n-1-I);J++)
            if(A[J]>A[J+1])
           {
               Temp=A[J]; //swapping
               A[J]=A[J+1];
               A[J+1]=Temp;
           }
    }
}
```

```
Q9.  voidSSort(int A[], int n)
               {   intI,J,Temp,Small;
                    for(I=0;I<n-1;I++)
                         {              Small=I;
               for(J=I+1;J<n;J++) //finding the smallest element
               if(A[J]<A[Small])
                      Small=J;
               if(Small!=I)
                         {
                    Temp=A[I]; //Swapping
                    A[I]=A[Small];
                    A[Small]=Temp;
                         } }  }
Q10. void ISort(int A[], int n)
     {
            intI,J,Temp;
            for(I=1;I<n;I++) //sorting
            {
                Temp=A[I];
                J=I-1;
                while((Temp<A[J]) && (J>=0))
                {
                    A[J+1]=A[J];
                    J--;
                }
                A[J+1]=Temp;
            }
     }
```

Q11. Bubble Sort

14,10,-12,9,15,35(Original Content)

i. 14,10,9,15,35,-12

ii. 14,10,15,35,9,-12

iii. 14,15,35,10,9,-12

iv. 15,35,14,10,9,-12(Unsorted statusafter 4th pass)

Selection Sort

14,10,-12,9,15,35(Original Content)

i. 35,10,-12,9,15,14

ii. 35,15,-12,9,10,14

iii. 35,15,14,9,10,-12

iv. 35,15,14,10,9,-12

```
Q12. Loc(P[I][J]) along the row =BaseAddress+W [(I-LBR)*C+(J-LBC)]
(where C is the number of columns, LBR=LBC=0)
LOC(P[5][5])= BaseAddress + W*[I*C + J]
15000 = BaseAddress + 4*[5*50 + 5]
= BaseAddress + 4*[250 + 5]
= BaseAddress + 4*255
= BaseAddress + 1020
BaseAddress = 15000-1020
= 13980
LOC(P[10][30])= 13980 + 4*[10*50+30]= 13980 + 4*530
= 13980 + 2120
= 16100
```

```
Q13. Loc(ARR[I][J]) along the row =BaseAddress + W[( I - LBR)*C+(J - LBC)]
(where C is the number of columns, LBR = LBC = 0
LOC(ARR[10][5])= BaseAddress + W [ I*C + J]
15000 = BaseAddress + 4[10*20 + 5]
= BaseAddress + 4[200 + 5]
= BaseAddress + 4 x 205
= BaseAddress + 820
BaseAddress = 15000-820
= 14180
LOC(ARR[30][10])= 14180 + 4[30 * 20 + 10]
                 = 14180 + 4 * 610    = 14180 + 2440
               = 16620
OR
LOC(ARR[30][10])= LOC(ARR[10][5])+ W[( I-LBR)*C + (J-LBC)]
= 15000 + 4[(30-10)*20 + (10-5)]
= 15000 + 4[ 20*20 + 5]
= 15000 + 4 *405
= 15000 + 1620
= 16620
```

```
Q14. T[i][j]=Base Addr + [i * number of columns+j]* size of each element
     T[10][15]= 42000+[(10*20)+15]*2
               = 42000+215*2
               =42000+430=42430
     Total number of elements in array is = 25*20=500
```

```
Q15. T[i][j]=Base Addr + [i * number of columns+j]* size of each element
     A[15][10]= 32000+[(15*30)+10]*4
               = 32000+460*4
               =32000+1840=33840
     Total number of elements in array is = 20*30=600
```

Q16. B=10000, W=4, N=10, I=2, J=5
A[I][J]=B+W[(I-LBR)+(J-LBC)*N]
A[2][5]=10000+4(2+10*5)
=10000+4(52)
=10000+208=10208

Q17. B=14000, W=4, N=15, I=8, J=5
P[I][J]=B+W[(I-LBR)+(J-LBC)*N]
P[8][5]=14000+4(8+5*15)
=14000+4(83)
=14000+332=14332

Q18. Address of T[i][j]=address of T[0][0]+(i*number of columns present in array +j)*sizeof(element)
Address of T[10][7]=14000+(10*10+7)*8
=14000+(107)*8
=14000+856
=14856

Q19. T[2][9]=Base addr+2[2+9*20]
7600=Base addr+2*(182)
Base addr=7600-364=7236
T[10][5]=7236+2(10+5*20)
= 7236+110*2
= 7456

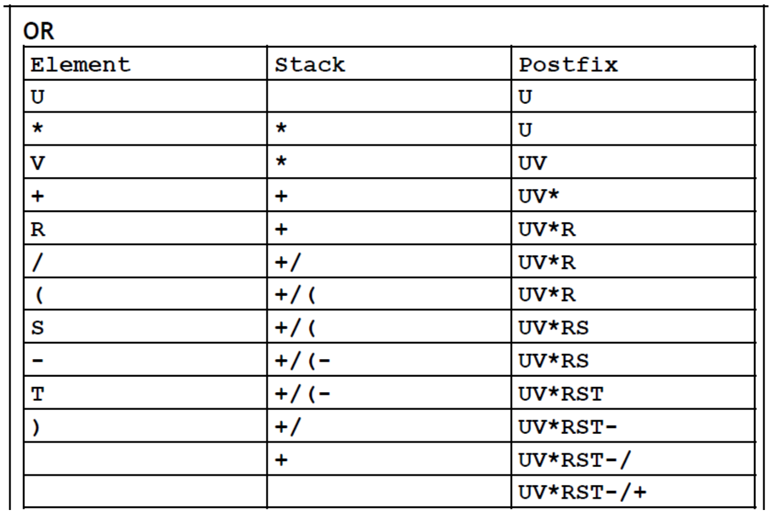
Q20. Assuminq LBR=LBC=0
B=5200
W=4 bytes
Number of Rows(N)=20
Number of Columns(M)=50
LOC(Arr[I] [J]) = B +(I + J*N)*W
LOC(Arr[15][10]) = 5200+(15+10*20)*4
= 5200 + (215*4)
= 5200 + 860
= 6060

Q21. X / Y + U* (VW)=((X / Y)+(U*(VW)))

Element	Stack	Postfix
(		
(		
X		X
/	/	X
Y	/	XY
)		XY/
+	+	XY/
(	+	XY/
U	+	XY/U
*	+*	XY/U
(	+*	XY/U
V	+*	XY/UV

-	+*-	XY/UV
W	+*-	XY/UVW
)	+*	XY/UVW-
)	+	XY/UVW-*
)		XY/UVW-*+

Q22.



OR

Element	Stack	Postfix
U		U
*	*	U
V	*	UV
+	+	UV*
R	+	UV*R
/	+/	UV*R
(	+/(	UV*R
S	+/(	UV*RS
-	+/(-	UV*RS
T	+/(-	UV*RST
)	+/	UV*RST-
	+	UV*RST-/
		UV*RST-/+

23. Equivalent postfix expression:
=((A-B)*(D/E))/(F*G*H)
=((AB-)*(DE/))/(FG*H*)
=AB – DE /* FG* H*/

24. Equivalent postfix expression:
= A*(B+D)/E-F-(G+H/K)
= (A*(B+D)/E) – (F – (G + (H/K)))
= (A*(BD+)/E) – (F – (G + (HK/)))
= ((ABD+*) / E) – (F – (GHK/+))
= ABD+* E/F – GHK/+ -

25. 10 * 3 * (7 – 1) + 23

26. a, b, AND, a, c, AND, OR
(a AND b), (a AND c), OR
(a AND b) OR (a AND c)

27.

Symbol	Stack
12	12
7	12,7
3	12,7,3
-	12,4
/	3
2	3,2
1	3,2,1
5	3,2,1,5
+	3,2,6

*	3,12
+	15
)	15

```
28.  A*(B+(CD+EF+*)/G)*H
     A*(B+CD+EF+*G/))*H
     (A*(BCD+EF+*G/+))H
     (ABCD+EF+*G/+*)*H
     ABCD+EF+*G/+*H*

29.  =((A NOT) OR ((B NOT) AND (C NOT)))
     =((A NOT) OR ((B NOT C NOT AND))
     =A NOT B NOT C NOT AND OR

30.       P = ABCDE/^*F/+
          2,3,2,7,2
          2,3,2->7/2->3
          2,3,2,3
          2,3->2^3->8
          2,3,8
          2->3*8->24
          2,24,2
          2->24/2->12
          2,12
          2+12
          Result of evaluation = 14



31.  Equivalent postfix expression:
     = (A+B) *C+D/E-F
     = (((A+B)*C) + (D/E)) - F
     = (((AB+)*C) + (DE/)) - F
     = AB+ C* DE/ + F-
```

32. F, T, NOT, AND, F, OR, T, AND

Scanned Element	Operation	Stack Status
F	Push	F
T	Push	F, T
NOT	Pop one operand from stack	F
	NOT T = F	
	Push	F, F
AND	Pop two operands from stack	
	F AND F = F	
	Push	F
F	Push	F, F
OR	Pop two operands from stack	
	F OR F = F	
	Push	F
T	Push	F, T
AND	Pop two operands from stack	
	F AND T = F	
	Push	

	Pop all	F

Result F

33. T, F, NOT, AND, T, OR, F, AND

Scanned Element	Operation	Stack Status
T	Push	T
F	Push	T, F
NOT	Pop one operand from stack	T
	NOT F = T	
	Push	T, T
AND	Pop two operands from stack	
	T AND T = T	
	Push	T
T	Push	T, T
OR	Pop two operands from stack	
	T OR T = T	
	Push	T
F	Push	T, F
AND	Pop two operands from stack	
	T AND F = F	
	Push	F

Result F

34. 5, 2, *, 50, 5, /, 5, -, +

Scanned Element	Stack Status
5	5
2	5.2
*	10
50	10, 50
5	10, 50, 5
/	10, 10
5	10, 10, 5
-	10, 5
+	15

35. 60, 6, /, 5, 2, *, 5, -, +

Scanned Element	Stack Status
60	60
6	60, 6
/	10
5	10, 5
2	10, 5, 2
*	10, 10
5	10, 10, 5
-	10, 5

+	15

Q36. Let us rewrite like (X - Y / (Z + U) * V

Scanned Element	Stack Status	Expression
(	(	
X	(	X
-	(-	X
Y	(-	XY
/	(	

Answers to 4 marks Question

1.
```
void STACK::PUSH()
{
BOOKS *Temp;
Temp=new BOOKS;
gets(Temp->ISBN);
gets(Temp->TITLE);
Temp->Link=Top;
Top=Temp;
}
```

2.
```
#include<iostream.h>
#include<stdio.h>
struct Node
{    char Country [20] ; Node *Link; };
class Stack
{ Node *Top;
public:
Stack( )
{ Top = NULL; }
void Push() ;
void Pop() ;
void Display() ;
~Stack () ;
};
void Stack::Push( )
{
Node *Temp = new Node;
gets(Temp -> Country);
Temp -> Link = Top;
Top = Temp;
}
void Stack::Pop( )
{
if (Top !=NULL)
{
Node *Temp = Top;
Top = Top -> Link;
delete Temp;
}
else
cout<<"stack Empty";
}
void Stack::Display( )
{
Node *Temp = Top;
while (Temp! = NULL)
{
cout<<Temp -> Country <<endl;
Temp = Temp -> Link;
}
}
```

```
Stack::~Stack ( )
{
while (Top!=NULL)
{ NODE *Temp=Top;
Top=Top->Link;
delete Temp;
}
}
void main ( )
{ Stack ST;
charCh;
do
{ cout<<"p/O/D/Q" ;
cin>>Ch;
switch (Ch)
{
case 'P' : ST.Push( ); break;
case 'O' :ST.Pop(); break;
case 'D' :ST.Disp();
}
} while (Ch!='Q');
}
```

3.

```
#include <iostream.h>
#include <conio.h>

struct NODE
{ char City[20];
NODE *Next;
};
class Queue
{ NODE *Rear,*Front;
puplic:
Queue( )
{ Rear=NULL;Front=NULL;
}
voidQinsert( );
voidQdelete( );
voidQdisplay( );
~Queue( );
} ;
void Queue::Qinsert( )
{
NODE *Temp;
Temp=new NODE;
cout<<"Data:";
gets (Temp->City);
Temp->Next=NULL;
if (Rear==NULL)
{
Rear=Temp;
Front=Temp;
}
```

```
     else
     {
     Rear->Next=Temp;
     Rear=Temp;
     }
     }
     void Queue::Qdelete( )
     {
     if (Front!=NULL)
     {
     NODE *Temp=Front;
     cout<<Front->City<<”Deleted \n”;
     Front=Front->Next;
     delete Temp;
     if (Front==NULL)
     Rear=NULL;
     }
     else
     cout<<”Queue Empty..”;
     }
     Queue::Qdisplay( )
     { NODE *Temp=Front;
     while (Temp!=NULL)
     {
     cout<<Temp->City<<endl;
     Temp=Temp->Next;
     }
     }
     Queue:: ~Queue( )//Destructor Function
     { while (Front!=NULL)
     { NODE *Temp=Front;
          Front=Front->Next; delete Temp;

     }
     }
     void main( )
     { Queue QU;
     charCh;
     do
     {
     :
     :
     } while (Ch!=’Q’);
          }

4.   void QUEINS (Node *&Front, Node *&Rear)
     { Node *Temp = new Node;
     cin>>Temp->PId;
     gets (Temp->Pname);
     //or cin>>Temp->Pname;
     //cin.get1ine(Temp->Pname);
     Temp->Next = NULL;
     if(Rear == NULL)
     Front = Temp;
     e1se
     Rear -> Next = Temp;
     Rear = Temp;
     }
```

```
5.   class Queue
     {Node *Front, *Rear;
     public:
     QUEUE( )//Constructor to initialize Front and Rear
     {
     Front = NULL;
     Rear = NULL;
     }
     void QUEINS( );     //Function to insert a node
     void QUEDEL( );     //Function to delete a node
     void QUEDISP( );    //Function to displaynodes
     ~Queue();           //Destructor to delete allnodes
     };
     void Queue::QUEDEL( )
     { if (Front!=NULL)
     {NODE *Temp=Front;
     cout<<Front->Itemno<<” “;
     cout<<Front->Itemname<<”Deleted“;
     Front=Front->Link;
     delete Temp;
     if (Front NULL)
     Rear=NULL;
     }
     else
     cout<<”Queue Empty..”;
     }
```

```
6.   struct Book
     {
     intBNo ;
     charBName[20] ;
     Book *Next ;
     } ;
     class Stack
     {
     Book *Top;
     public:
     Stack( )
     {
     Top = NULL;
     }
     void Push( );
     void Pop( );
     void Display( );
     };
     void Stack::Pop( )
     {
     Book *Temp;
     if( Top= = NULL)
     cout<<”Stack Underflow…”;
     else
     {
     cout<<”\nThe Book number of the
     element to delete: “<<Top->BNo;
     cout<<”\nThe Book name of the
     element to delete: “<<Top->BName;
     Temp=Top;
     Top=Top->Next;
     delete Temp;
     }
          }

7.   class Queue
     {    NODE *front,*rear;
     public:
     Queue( )
     {    front = rear = NULL;     }
     void Insert( );
     void Delete( );
     void Display( );
     };
     void Queue::Insert( )
     {
     NODE *ptr;
     ptr=new NODE;
     if(ptr= = NULL)
     {    cout<<”\nNo memory to create a
     new node….”;
     exit(1);
     }
     cout<<”\nEnter the name….”;
     gets(ptr->Name);
     ptr->Link=NULL;
     if(rear= = NULL)
     front=rear=ptr;
     else
     {
     Rear->Link=ptr;
     rear=ptr;
     }
     }

8.   struct Node
     {int X,Y ;
     Node *Link ;
     } ;
     class STACK
     {Node *Top ;
     public :
     STACK( )
     { Top = NULL;
     }
     void PUSH( ) ;
     void POP( ) ;
     ~STACK( ) ;
     } ;
     void STACK::PUSH( )
     {Node *Temp;
     Temp=new Node;
     if(Temp= =NULL)
     {cout<<”\nNo memory tocreate the node…”;
     exit(1);
     }cout<<”Enter the value of X and Y”;
     cin>>Temp->X>>Temp->Y;
     Temp->Link=Top;
     Top=Temp;
     }

9.   void QINSERT()
{ NODE *P=new NODE();
cout<<"Enter the client number";
cin>>P->Cno;
cout<<"enter the client name";
gets(P->Cname);
 P->Next=NULL;
if(front==NULL && rear==NULL)
{ front=P;
rear=P;
  }
else
{ rear->Next=P;     rear=P;   }  }

10.  void PUSHBOOK(NODE *top)
     { NODE *NEW=new NODE;
     cout<<"Enter the book number";
     cin>>NEW->Book_No;
     cout<<"Enter book title";
     gets(NEW->Book_Title);
       NEW->Next=NULL;
     if(top==NULL)
     top=NEW;
     else { NEW->Next=top; top=NEW;}
       }
10.  void POPBOOK(NODE *top)
     {
     cout<<"deleting top element from stack\n";
     cout<<"Book No"<<top->Book_No<<endl;
     cout<<"Book title"<<top->Book_Title<<endl;
       NODE *temp=top;
     top=top->Link;
     delete(temp);
```