LINKED LISTS , STACKS AND QUEUES

DELHI 2008:

3.c) Write a function in C++ to **insert** an element into a **dynamically allocated Queue** where each node contains a name (of type string) as data.

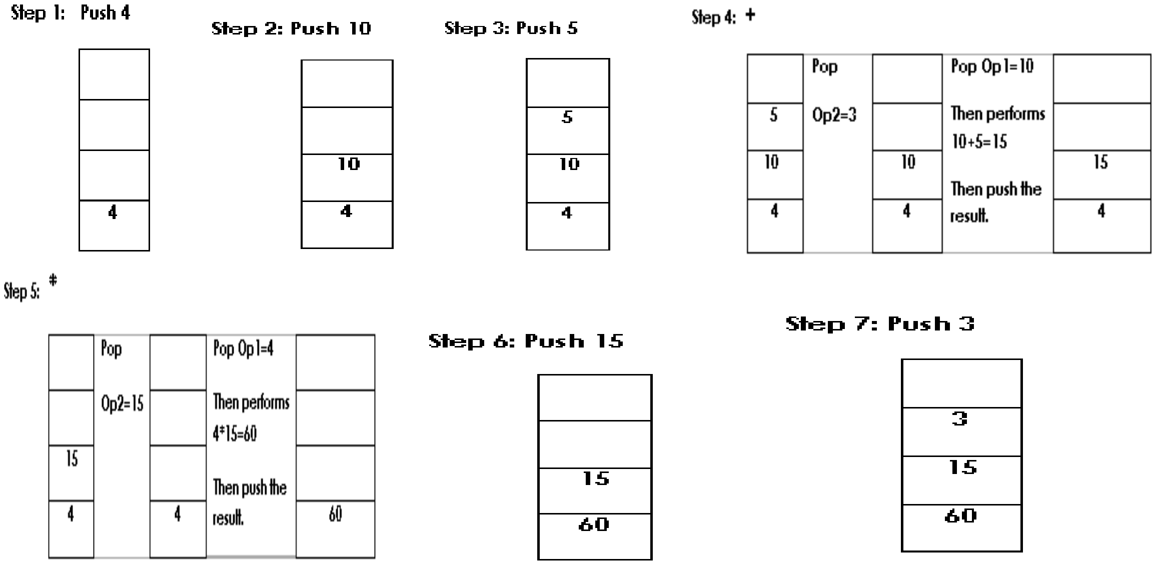
Assume the following definition of THENODE for the same.

```
struct THENODE
{
  char Name[20];
  THENODE *Link;
};
```

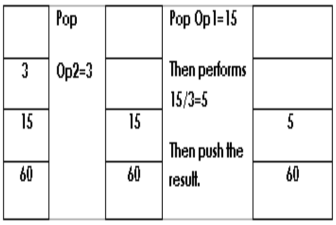
Solution:

```
struct THENODE
{
   char Name[20];
   THENODE *Link;
};
class Queue
{
   THENODE *front,*rear;
public:
   Queue( )
   {    front = rear = NULL;  }
   void Insert( );
   void Delete( );
   void Display( );
};
void Queue::Insert( )
{
    THENODE *ptr;
    ptr=new THENODE;
    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;
     }
}
```

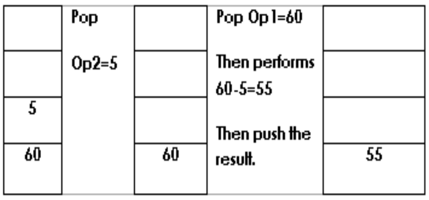
3.e) Evaluate the following postfix notation of expression (Show status of stack after execution of each operation): **4, 10, 5, +, *, 15, 3, /, -** 2



Step 8: - /



Step 9: -



OUTSIDE DELHI 2008:

3.c) Write a function in C++ to **Delete** an element into a **dynamically allocated Queue** where each node contains a real number as data. 4

Assume the following definition of MYNODE for the same:

```
struct MYNODE
{
        float NUM;
        MYNODE * Link;
};
```

Solution:

```
struct MYNODE
{
  float NUM;
  MYNODE *Link;
};
class Queue
{
   MYNODE *front,*rear;
public:
   Queue( )
    {   front=rear=NULL; }
    void Insert( );
    void Delete( );
    void Display( );
};
void Queue::Delete( )
{
    MYNODE *temp;
    if(front= = NULL)
         cout<<"Queue Underflow";
    else
    {
         cout<<"\nThe content of the element to delete: "<<front→NUM;
         temp=front;
         front=front→Link;
         delete temp;
     }
 }
```

3.e) Evaluate the following **postfix** notation of expression (Show status of stack after execution of each operations): 5, 20, 15, -, *,25, 2, *, + 2

Ans) Children, Try this answer as an assignment.

DELHI : 2007

3.c) Write a function in C++ to **delete** a node containing Book's information, from a **dynamically allocated Stack** of Books implemented with the help of the following structure.

```
struct Book
{   int BNo ;
    char BName[20] ;
    Book *Next ;
} ;
```

Solution:

```
struct Book
 {    int BNo ;
      char BName[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;
      }
}
```

3.e)Evaluate the following **postfix** notation of expression : 25 8 3 - / 6 * 10 + 2

Ans) Children, Try this answer as an assignment.

OUTSIDE DELHI : 2007

3.c) Write a function in C++ to **delete** a node containing customer's information, from a **dynamically allocated Queue** of Customers implemented with the help of the following structure:

```
struct Customer
{     int CNo ;
      char CName[20] ;
      Customer *Link ;
} ;
```

Solution:

```
struct Customer
{      int CNo ;
       char CName[20] ;
       Customer *Link ;
} ;
class Queue
{  Customer *front,*rear;
public:
   Queue( )
    {   front=rear=NULL; }
   void Insert( );
   void Delete( );
   void Display( );
};
void Queue::Delete( )
{     Customer *Temp;
      if(front= =NULL)
           cout<<"Queue Underflow.  No element to delete";
      else
      {    cout<<"\n The customer number for the element to delete"<<front→CNo;
           cout<<"\n The customer name for the element to delete"<<front→CName;
           Temp=front;
           front = front→Link;
           delete Temp;
       }
}
```

3.e) Evaluate the following **postfix** notation of expression : 15 3 2 + / 7 + 2. * 2

Ans) Children, Try this answer as an assignment.

DELHI : 2006

3.c)
```
class queue
{    int data[10] ;
     int front, rear ;
public :
     queue( )   {    front = - 1 ; rear = - 1 ;    }
     void add( ) ;        //to add an element into the queue
     void remove( ) ;  //to remove an element from the queue
     void Delete(int ITEM( ) ;   //to delete all elements which are equal to ITEM
   } ;
```

Complete the class with all function definitions for a circular array Queue. Use another queue to transfer data temporarily.

Solution:
```
void queue::add( )
{  if((front= = 0 && rear = = 9) | | (front= =rear+1)
          cout<<"\nQueue Overflow";
   else if (rear= = -1)
   {      front=rear=0;
          cout<<"\nEnter the element to be inserted";
          cin>>data[rear];
   }
   else if(rear= =9)
   {      rear=0;
          cout<<"\nEnter the element to be inserted";
          cin>>data[rear];
    }
    else
    {     rear++;
          cout<<"\nEnter the element to be inserted";
          cin>>data[rear];
     }
}
void queue::remove( )
{    if(front= = -1)
          cout<<"\nQueue Underflow…";
     else
     {    cout<<"\nThe element to be deleted"<<data[front];
          if(front= =rear)
               front=rear=-1;
           else if (front= =9)
               front=0;
           else
               front++;
       }
}
void queue::Delete(int ITEM )
{




}
```

3.d) Write a function in C++ to perform a PUSH operation on a dynamically allocated stack containing real number.

```
struct Node
{    float Number ;
     Node *Link ;
} ;
class STACK
{    Node *Top ;
public :
     STACK( )  {Top = NULL ;}
     void PUSH( ) ;
     void POP( ) ;
     ~STACK( ) ;
} ;
```

Solution:

```
struct Node
{    float Number ;
     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 to create the node…";
          exit(1);
      }
     cout<<"\nEnter the Number to be inserted: ";
     cin>>Temp→Number;
     Temp→Link=Top;
      Top=Temp;
}
```

3.e) Write the equivalent infix expression for a, b, AND, a, c, AND, OR.

Ans) a, b, AND, a, c, AND, OR
(a AND b), (a AND c), OR
(a AND b) OR (a AND c)

OUTSIDE DELHI : 2006

3.c) Introduction class stack

```
{    int data[10] :
     int top ;
public :
     stack( )          {    top = - 1;    }
     void push( ) ;       //to push an element into the stack
     void pop( ) ;        //to pop an element from the stack
     void Delete(int ITEM) ; //To delete all elements which are equal to ITEM.
} ;
```

Complete the class with all function definitions. Use another stack to transfer data temporarily.

Solution:

```
void stack::push( )
{    if(top>=9)
          cout<<"Stack Overflow…";
     else
      {   top++;
          cout<<"\nEnter the element to be inserted…";
      cin>>data[top];
    }
}
void stack::pop( )
{   if(top= =-1)
        cout<<"\nStack Underflow";
    else
    {   cout<<"\nThe element to be deleted = "<<data[top];
        top--;
    }
}
void stack::Delete(int ITEM)
{









}
```

3.d) Write a function in C++ to perform **Insert** operation in **dynamically allocated Queue** containing names of students.

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

Solution: 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;
     }
}
```

3.e) Write the **equivalent infix** expression for 10, 3, *, 7, 1, --,*, 23, +

Solution:

10, 3, *, 7, 1, - , *, 23, + This is in Postfix form(ie Operator will come after the operand(s));. Infix form means Operator must come in between the operands.

10, 3, *, 7, 1, - , *, 23, +

Prefix: 10 * 3, 7 – 1,*,23,+
(10 * 3) * (7 – 1),23,+
(10 * 3) * (7 – 1) + 23 **(Infix form)**

DELHI : 2005

3.c) Write a function in C++ to perform a **PUSH** operation in a **dynamically allocated stack** considering the following :

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

Solution:

```
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 to create the node…";
             exit(1);
        }
        cout<<"Enter the value of X and Y";
        cin>>Temp→X>>Temp→Y;
        Temp→Link=Top;
         Top=Temp;
        }
 }
```

3.e) Evaluate the following postfix notation of expression : 10 20 + 25 15 - * 30 /

Ans) Children, Try this answer as an assignment.

OUTSIDE DELHI : 2005

3.c) Write a function in C++ to perform a **DELETE** operation in a **dynamically allocated queue** considering the following description :

```
struct Node
  {    float U, V ;
       Node *Link ;
  } ;
  class QUEUE
  {
       Node *Rear, *Front ;
   public :
       QUEUE( ) {Rear = NULL ; Front = NULL ;}
        void INSERT( ) ;
        void DELETE( ) ;
        ~ QUEUE( ) ;
   } ;
```

```
Solution:      void Queue::DELETE( )
                 {
                    NODE *temp;
                    if(front= = NULL)
                         cout<<"\nQueue Underflow";
                    else
                    {
                         cout<<"\nThe value of U  of the element to delete: "<<Front→U;
                         cout<<"\nThe value of V of the element to delete: "<<Front→V;
                         temp=Front;
                         Front=Front→Link;
                         delete temp;
                     }
                 }
```

3.e) Evaluate the following **postfix** notation of expression :   20 10 + 5 2 * - 10 /

Ans) Children, Try this answer as an assignment.

2004

3.d) Obtain the postfix notation for the following infix notation of expression showing the contents of the stack and postfix expression formed after each step of conversion : (P—Q)/(R*(S—T)+U)

(Ans).((P-Q)/((R*(S-T))+U))

S.No	Symbol Scanned	Stack	Expression Y
1	(	(	
2	(	((	
3	P	((	P
4	-	((-	P
5	Q	((-	P Q
6	)	(	P Q -
7	/	(/	P Q -
8	(	(/ (	P Q -
9	(	(/ ((	P Q -
10	R	(/ ((	P Q - R
11	*	(/ ((*	P Q - R
12	(	(/ ((* (	P Q - R
13	S	(/ ((* (	P Q - R S
14	-	(/ ((* (-	P Q - R S
15	T	(/ ((* (-	P Q - R S T
16	)	(/ ((*	P Q - R S T -
17	)	(/ (	P Q - R S T - *
18	+	(/ (+	P Q - R S T - *
19	U	(/ (+	P Q - R S T - * U
20	)	(/	P Q - R S T - * U +
21	)		P Q - R S T - * U + /

Postfix Form: PQ-RST-*U+/

3.e) Define member functions **queins( )** to insert nodes and **quedel ( )** to delete nodes of the **linked list implemented class queue**, where each node has the following structure:

```
struct node
{       char name[20] ;
        int age ;
        node *Link ;
} ;
class queue
{       node *rear, *front ;
public :
        queue( ) { rear = NULL; front = NULL} ;
        void queins( ) ;
        void quedel( ) ;
} ;
```

Solution:

```
void queue::queins( )
{   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);
cout<<"\nEnter the age…";
cin>>ptr→age;
ptr→Link=NULL;
if(rear= = NULL)
      front=rear=ptr;
else
    {
      rear→Link=ptr;
      rear=ptr;
    }
}
void queue::quedel( )
 {   node *temp;
     if(front= = NULL)
          cout<<"Queue Underflow";
    else
    {     cout<<"\nThe name of the element to delete: "<<front→name;
          cout<<"\nThe age of the element to delete: "<<front→age;
          temp=front;
          front=front→Link;
          delete temp;
    }
  }
```

DELHI : 2003

3.c) Evaluate the following **postfix** expression using a stack and show the contents of stack after execution of each operation: 20, 45, +, 20, 10, -, 15, +, *

Ans) Children, Try this answer as an assignment.

3.e) Consider the following portion of a program, which implements passengers Queue for a train. Write the definition of function. **Insert** (whose prototype is shown below); to insert a new node in the queue with required information.

```
struct NODE
     {    long Ticketno;
          char PName[20];//Passengers Name
          NODE * Next;
     };
     class Queueoftrain
     {     NODE * Rear, * Front;
     public :
```

```
                    Queueoftrain( ) { Rear = NULL; Front = NULL:}
                    void Insert( );
                    void Delete( );
                    ~Queueoftrain( );
                } ;
```

Solution:

```
        void Queueoftrain::Insert( )
        {   NODE *ptr;
            ptr=new NODE;
            if(ptr= = NULL)
            {
                        cout<<"\nNo memory to create a new node….";
                        exit(1);
            }
            cout<<"\nEnter the Ticket Number….";
            cin>>ptr→Ticketno;
            cout<<"\nEnter the Passenger Name..";
            gets(ptr→PName);
            ptr→Next=NULL;
            if(rear= = NULL)
                  front=rear=ptr;
            else
            {
                  rear→Next=ptr;
                   rear=ptr;
            }
          }
```

DELHI : 2002

3.b) Given the following class, 4

```
     char *msg[ ]={"over flow","under flow"};
     class Stack
     {    int top;          //the stack pointer
           int stk[5];       //the elements
          void err_rep(int e_num)
          {                  cout<<msg[e_enum];      //report error message
         }
        public:
            void init( )
            {                 top=0;
            } //initialize the stack pointer
           void push(int);               //put new value in stk
           void pop( );                //get the top value.
        };
```

Define pop outside the Stack. In your definition take care of under flow condition. Function pop should invoke err_rep to report under flow.

Solution:

```
void Stack::pop( )
{







}
```

3.c) Change the following **infix expression into postfix** expression. (A+B)*C+D/E-F 3

Ans) Children, Try this answer as an assignment.

DELHI : 2001

3.d) Write an **algorithm to convert an infix expression to postfix** expression.

Ans)

The following algorithm transforms the infix expression X into its equivalent postfix expression Y. The algorithm uses a stack to temporarily hold operators and left parentheses. The postfix expression Y will be constructed from left to right using the operands from X and the operators which are removed from STACK. We begin by pushing a left parenthesis onto STACK and adding a right parenthesis at the end of X. The algorithm is completed when STACK is empty.

Algorithm:

Suppose X is an arithmetic expression written in infix notation. This algorithm finds the equivalent postfix expression Y.

1. Push “(“ onto STACK, and add “)” to the end of X.
2. Scan X from left to right and REPEAT Steps 3 to 6 for each element of X UNTIL the STACK is empty.
3. If an operand is encountered, add it to Y.
4. If a left parenthesis is encountered, push it onto STACK.
5. If an operator is encountered, then:
   (a) Repeatedly pop from STACK and add to Y each operator(on the top of STACK) which has the same precedence as or higher precedence than operator.
   (b) Add operator to STACK.
   /* End of If structure */
6. If a right parenthesis is encountered, then:
   (a) Repeatedly pop from STACK and add to Y each operator (on the top of STACK) until a left Parenthesis is encountered.
   (b) Remove the left parenthesis. (Do not add the left parenthesis to Y).
   /* End of If structure */
7. End.

3.e) Each node of a STACK contains the following information, in addition to pointer field:

(i).Pin code of city (ii).Name of city

Give the structure of node for the linked STACK in question. TOP is a pointer that points to the topmost node of the STACK. Write the following functions:

4

(i) **PUSH( )** – To push a node into the STACK, which is allocated dynamically.

(ii) **POP( ) –** To remove a node from the STACK, and release the memory.

Solution:

```
struct City
{   long Cpin ;
    char CName[20] ;
    City *Next ;
} ;
class Stack
{   City *Top;
   public:
    Stack( ) {   Top = NULL; }
    void Push( );
    void Pop( );
    void Display( );
};
void Stack::PUSH( )
{    City *Temp;
     Temp=new City;
      if(Temp= =NULL)
      {
            cout<<”\nNo memory to create the node…”;
            exit(1);
      }
      cout<<”\nEnter the City Pin Code to be inserted: “;
      cin>>Temp→Cpin;
      cout<<”\nEnter the City Name to be inserted: “;
```

```
          gets(Temp→CName);
          Temp→Next=Top;
           Top=Temp;

}
void Stack::POP( )
{         City *Temp;
          if( Top= = NULL)
                    cout<<"Stack Underflow…";
          else
          {         cout<<"\nThe City Pin Code for the element to delete: "<<Top→Cpin;
                    cout<<"\nThe City name of the element to delete: "<<Top→CName;
                    Temp=Top;
                    Top=Top→Next;
                    Delete Temp;
           }
}
```

DELHI : 2000

3.d)Evaluate the following **postfix** expression using a stack. Show the contents of stack after execution of each operation: 20, 8, 4, /, 2, 3, +, *, -

Ans) Children, Try this answer as an assignment.

.**3.e)**Give necessary declarations for a queue containing float type numbers; also write a user defined function in C++ **to insert** a float type number in the queue. You should use **linked representation of queue.**

Solution:

```
struct NODE
{    float Number;
   NODE *Link;
};
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 Number….";
  cin>>ptr→Number;
  ptr→Link=NULL;
  if(rear= = NULL)
          front=rear=ptr;
  else
  {
          rear→Link=ptr;
         rear=ptr;
   }
}
```

1999:

3 (d) Evaluate the following postfix expression using a stack and show the contents of the stack after execution of each operation   5,11,-,6,8,+,12,*,/

Ans) Children, Try this answer as an assignment.

3 (e) Give the necessary declaration of a **linked list implemented queue** containing float type elements. Also write a user defined function in C++ to **delete** a float type number from the queue.

```
 struct MYNODE
 {       float NUM;
        MYNODE * Link;
 };
```

Solution:

```
struct MYNODE
{  float NUM;
   MYNODE *Link;
};
class Queue
{  MYNODE *front,*rear;
public:
    Queue( )
     {   front=rear=NULL; }
    void Insert( );
    void Delete( );
    void Display( );
};
void Queue::Delete( )
{   MYNODE *temp;
     If(front= = NULL)
            cout<<”Queue Underflow”;
     else
     {      cout<<”\nThe content of the element to delete: “<<front→NUM;
            temp=front;
            front=front→next;
            delete temp;
     }
}
```

1998:

3 (d) Evaluate the following **postfix** expression using a stack and show the contents of stack after execution of each operation: 50, 40, +, 18, 14, -, 4, *, +

Ans) Children, Try this answer as an assignment.

3 (e) Give the necessary declaration of a **linked implemented stack** containing integer type numbers; also write a user defined function in C++ to **pop** a number from this stack.

Solution:

```
 struct Node
 {   float Number;
     Node *Next ;
 } ;
class Stack
{    Node *Top;
  public:
     Stack( )
     {    Top = NULL; }
     void Push( );
     void Pop( );
     void Display( );
};
void Stack::Pop( )
{    Node *Temp;
     If( Top= = NULL)
            cout<<”Stack Underflow…”;
     else
     {      cout<<”\nThe Number of the element to delete: “<<Top→Number;
            Temp=Top;
            Top=Top→Next;
             Delete Temp;
      }
}
```