**Question Bank , Class : XII**

**Chapter : Stacks & Queues**

**SLOW LEARNER**

( 1. ) Expand the terms LIFO and FIFO.

Ans. LIFO : Last In First Out.

FIFO : First In First Out.

(2.)What is a FIFO list technically called?

Ans. Queue.

( 3.) What is the difference between linear and non- linear data structures?

Ans. Single level data structures where elements form a sequence are called linear data structures eg. Stacks, queues, linked list etc. are linear data structures.Multilevel data structures are called non-liner data structures eg. Trees and graphs are non-linear data structures.

( 4. ) Describe the similarities and differences between queues and stacks.

Ans. Similarities:

(a) Both queues and stacks are special cases of linear lists.

(b) Both can be implemented as arrays or linked lists.

Dissimilarities :

(a) A Stack is a LIFO list, a queue is a FIFO list.

(b) There are no variations of a stack, a queue, however, may be circular or dequeue.

( 5. ) What is meant by the term “Overflow” & “Underflow”?

Ans. “Overflow” means attempt to INSERT when the list is full and no free space is available.

“Underflow” means attempt to DELETE an item from an empty list.

(6. ) Distinguish between infix, prefix and postfix algebraic expression giving examples of each.

Ans. Infix Notation : In this notation, the operator symbol is placed in between the operands eg.

A+B, (A-C)\* B

Prefix Notation : In this notation, the operator symbol is placed before its operands eg. +AB, \*-ACB

Postfix Notation : In this notation, the operator symbol is placed after its operands eg.

AB+, AC-B\*, ABC\*+

( 7. ) Write the algorithm for Insertion (i.e. PUSH) in a stack as an array ?

Ans. Algorithm :

1. If ( TOP = = N – 1 )

{

Write ( ‘ Stack Overflow ’ )

Go to Step – 4

}

1. TOP = TOP + 1
2. S [ TOP ] = DATA
3. End.

( 8. ) Write the algorithm to delete an element from a queue as an array?

Ans. Algorithm :

1. If ( F = – 1 )

{

Write ( ‘ Deletion Not Possible ’ )

Go to Step – 4

}

1. DATA = Q[F]
2. If ( F = = R )

{

F = 0

R = 0

}

Else

F = F + 1

1. End.

( 9. ) Obtain the prefix notation for the following infix notation of expression showing the contents of the stock and postfix expression formed after each step of conversion :

A \* B + (C – D / F)

Ans. Adding ) at the end of given expression and inserting ( in the stack.

A \* B + (C – D / F)

|  |  |  |
| --- | --- | --- |
| 1. Symbol | 1. Stack | 1. Postfix Expression |
| 1. A 2. \* 3. B 4. + 5. ( 6. C 7. - 8. D 9. / 10. F 11. ) 12. ] | 1. [ 2. [ \* 3. [ + 4. [+ ( 5. [+(- 6. [+(-/ 7. [+ 8. # | 1. A 2. AB 3. AB\* 4. AB \* C 5. AB \* CD 6. AB\*CDF 7. AB\*CDF/- 8. AB\*CDF/-+ |

( 10. ) Evaluate the following postfix notation of expression, show status of stack for each operation 500, 20, 30, + , 10, \* , +

Ans. Adding ] to mark the end of given postfix expression ie. 500, 20, 30 +, 10, \*, + ]

First 500 will be added to the stack, then 20 and then 30.

Then we get an operator + so 20 will be added to 30 and pushed to the stack. Then 10 is pushed to the stack.

Then \* operator is found thus it multiplies 10 with 50 and the result is pushed into the stack. Then again + operator is found which adds 500 with 500 that is placed in the stack. And we get the result 1000.

MEDIUM LEARNER

( 1.) Define the term STACK.

Ans. STACK : Stack is an important subclass of lists in which insertion or deletion of an element is performed only at one end. The insertion and deletion operations are known as PUSH and POP respectively. The most accessible element of a stack is called TOP of the stack.

( 2. ) Define the term Queue ?

Ans. QUEUE : A Queue is a subclass of lists in which insertion and deletion take place at specific ends i.e. Rear and Front respectively.

( 3. ) What are the draw – backs of a Linear – Queue as an Array?

Ans. Draw – Backs of Linear – queue as an array are as follows :

( I. ) Array is static in nature , so we have to declare the number of elements in the queue i.e. the size of the array at the time of declaration of the array.

( I. ) In the Linear Queue as an Array we may come across a situation where the array locations are empty , but we will not be able to insert a new element in the /queue.

( 4. ) Write an algorithm to insert a new element in a queue as an array?

Ans. Algorithm :

1. If ( R = = N – 1 )

{

Write ( ‘ Insertion Not Possible ’ )

Go to Step – 5

}

1. R = R + 1
2. Q[R] = DAA
3. If ( F = = – 1 )

F = 0

1. End

( 5. ) Write an algorithm to delete ( POP ) an element from a stack as an array?

Ans. Algorithm :

1. If ( TOP = = – 1 )

{

Write ( ‘ Stack Underflow ’ )

Go to Step – 4

}

1. DATA = S[TOP]
2. TOP = TOP – 1
3. End

( 6. ) Convert the following In Fix Notation Expression to Pre – Fix Notation Expression.

( A – B ) / C – D + E

Ans The given In Fix Expression is :

( A – B ) / C – D + E

( – A B ) / C – D + E

( / ( – A B ) C ) – D + E

( – ( / ( – A B ) C ) D ) + E

+ ( – ( / ( – A B ) C ) D ) E

+ – / – A B C D E

Which is a Pre – Fix Notation Expression.

( 7. ) Define the following terms :

( a. ) DeQueue ( b. ) Priority Queue

Ans : ( a. ) DeQueue : DeQueue stands for Double – Ended Queue . It is a linear – list in which insertions and deletions are made to or from either end of the queue.

( b. ) Priority Queue : A queue in which we can insert or delete (remove) elements from any position depending on some priority is known as “Priority Queue”.

( 8. ) Convert the following In Fix Notation Expression to Post – Fix Notation Expression.

NOT A OR NOT B AND NOT C

Ans The given In Fix Expression is :

NOT A OR NOT B AND NOT C

The actual order of evaluation will be :

( ( NOT A ) OR ( ( NOT B ) AND ( NOT C ) ) )

( ( A NOT ) OR ( ( B NOT ) AND ( C NOT ) ) )

( ( A NOT ) OR ( B NOT C NOT AND ) )

A NOT B NOT C NOT AND OR

Which is a Post – Fix Notation Expression.

( 9. ) Evaluate the following postfix notation of expression:

25 8 3 - / 6 \* 10 +

Ans.

|  |  |
| --- | --- |
| Operator Scanned | Stack Content |
| 25 | 25 |
| 8 | 25, 8 |
| 3 | 25, 8, 3 |
| - | 25, 5 |
| / | 5 |
| 6 | 5, 6 |
| \* | 30 |
| 10 | 30, 10 |
| + | 40 |

( 10. ) Write a function in C++ to perform Insert operation in a dynamically allocated Queue containing names of students.

struct stud

{

char Name[20];

stud \*Link;

};

Ans : struct stud

{

char Name[20];

stud \*Link;

} \*front, \*rear;

void Insert(stud \*np)

{

if (front = = NULL)

front = rear = np;

else

{

rear→Link = np;

rear = np;

}

}

**BRILLIANT LEARNER**

( 1. ) Write the algorithm to insert ( PUSH ) an element in a stack as a list.

Ans. Algoritm :

1. If ( AVAIL = = NULL )

{

Write ( ‘ Availability Stack Underflow ’ )

Go to Step – 6

}

1. NEWPTR = AVAIL
2. AVAIL = LINK ( AVAIL )
3. INFO (NEWPTR) = DATA

LINK ( NEWPTR ) = TOP

1. TOP = NEWPTR
2. End.

( 2. ) Evaluate the following postfix notation of expression :

15 3 2 + / 7 + 2 \*

Ans :

Evaluation of the given postfix expression is explained below

Operator Scanned Stack Content

15 15

3 15, 3

2 15, 3, 2

+ 15, 5

/ 3

7 3, 7

+ 10

2 10, 2

\* 20

( 3. ) 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 ;

};

Ans. Class stack

{

public:

book( )

{

top=NULL; }

void Push ( );

void Pop( );

void Display( );

~Book( );

};

void stack :: Pop( )

{

if (top != NULL)

{

stack \*temp;

temp =Top;

cout<<Top🡪Bno<<Top🡪 “deleted”<<endl;

Top= Top 🡪 Next ;

delete Temp;

}

Else

cout<< “stock Emplty” ;

}

( 4. ) 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;

};

Ans :

class QUEUE

{

Customer \*Rear,\*Front;

public:

QUEUE( ) { Rear=NULL; Front=NULL;}

void DELETE( );

~QUEUE( );

};

//Function definition DELETE()

void QUEUE::DELETE()

{

if(Front==NULL) // OR if(!Front)

cout<<”\n Queue Underflow\n”;

else

{

Customer \*Temp;

Temp=Front;

cout<<Front->Cno<<”:”<<Front->Cname <<”Deleted”<<endl;//To be ignored

Front=Front->Link;

delete Temp;

if (Front==NULL)

Rear=NULL;

}

}

( 5. ) 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;

};

Ans : struct node

{ int x,y;

node \*Link;

}\*top, \*temp;

void PUSH(node \*np)

{

if(top = = NULL)

top = np;

else

{

temp = top;

top = np; // New node becomes to the first node

np→Link = temp;

}

}

( 6. ) Write a function in C++ to perform Push operation on a dynamically allocated Stack containing real numbers.

struct NODE

{

float Data; NODE \*Link;

};

Ans: class STACK

{

NODE \*Top;

public:

STACK();

void Push();

void Pop();

};

void STACK::Push()

{

NODE \*Temp;

Temp=new NODE;

cin>>Temp→Data;

Temp→Link=Top;

Top=Temp;

}

(7. ) Evaluate the following postfix notation of expression:

True, False, AND, True, True, NOT, OR, AND

Ans :

Step 1: Push

|  |
| --- |
|  |
|  |
|  |
| True |

Step 2: Push

|  |
| --- |
|  |
|  |
| False |
| True |

Step 3: AND Push

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pop |  | Pop |  |
|  | Op2=True |  | Op1=False |  |
|  |  |  | Op2=True |  |
| True |  |  |  | False |

Step 4: Push

|  |
| --- |
|  |
|  |
| True |
| False |

Step 5: Push

|  |
| --- |
|  |
| True |
| True |
| False |

Step 6: NOT Push

|  |  |  |
| --- | --- | --- |
|  | Pop |  |
|  | Op2=True | False |
| True |  | True |
| False |  | False |

Step 7: OR Push

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pop |  | Pop |  |
|  | Op2=False |  | Op1=True |  |
| True |  |  | Op2=False | True |
| False |  | False |  | False |

Step 8: AND Push

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pop |  | Pop |  |
|  | Op2=True |  | Op1=False |  |
|  |  |  | Op2=True |  |
| False |  |  |  | False |

Step 9: Pop

|  |  |
| --- | --- |
|  |  |
|  |  |
|  | Result |
|  | False |

( 8. ) Evaluate the following postfix notation of expression:

25 8 3 – / 6 \* 10 +

Ans:

|  |  |
| --- | --- |
| Operator Scanned | Stack Content |
| 25 | 25 |
| 8 | 25, 8 |
| 3 | 25, 8, 3 |
| - | 25, 5 |
| / | 5 |
| 6 | 5, 6 |
| \* | 30 |
| 10 | 30, 10 |
| + | 40 |

So, the answer is 40.

( 9. ) Convert A + ( B \* C – ( D / E )) \* F into postfix form showing stack status after every step.

Ans :

|  |  |  |  |
| --- | --- | --- | --- |
| Step No. | Symbol Scanned | Stack | Expression |
| 1 | A | ( | A |
| 2 | + | ( + | A |
| 3 | ( | ( + ( | A |
| 4 | B | ( + ( | A B |
| 5 | \* | ( + ( \* | A B |
| 6 | C | ( + ( \* | A B C |
| 7 | – | ( + ( - | A B C \* |
| 8 | ( | ( + ( - ( | A B C \* |
| 9 | D | ( + ( - ( | A B C \* D |
| 10 | / | ( + ( - ( / | A B C \* D |
| 11 | E | ( + ( - ( | A B C \* D E |
| 12 | ) | ( + ( | A B C \* D E / – |
| 13 | ) | ( | A B C \* D E / – + |
| 14 | \* | ( \* | A B C \* D E / – + |
| 15 | F | ( \* | A B C \* D E / – + F |
| 16 | ) |  | A B C \* D E / – + F \* |

So, the postfix form is: A B C \* D E / – + F \*

( 10. ) Convert NOT A OR NOT B AND NOT C into postfix form.

Ans :

|  |  |  |  |
| --- | --- | --- | --- |
| Step No. | Symbol Scanned | Stack | Expression |
| 1 |  | ( |  |
| 2 | NOT | ( NOT |  |
| 3 | A | ( NOT | A |
| 4 | OR | ( OR | A NOT |
| 5 | NOT | ( OR NOT | A NOT |
| 6 | B | ( OR NOT | A NOT B |
| 7 | AND | ( OR AND | A NOT B NOT |
| 8 | NOT | ( OR AND NOT | A NOT B NOT |
| 9 | C | ( OR AND | A NOT B NOT C NOT |
| 10 |  | ) | A NOT B NOT C NOT AND OR |

So, the postfix form is: A NOT B NOT C NOT AND OR

**Designed By : Group – 08 ( PARAM )**