Part – A (Answer All) **Multiple Choice Questions** **[35 x 1 = 35 Marks]**

Use the following options for question 1-8

(A) True (B) False

1. A data structure may be an element of another data structure.
2. A structure is an aggregate data type built using elements of other types.
3. An algorithm is a single instruction that may be used to solve a problem.
4. Queue data structure can be used to check if an expression has balanced parenthesis.
5. After declaring a pointer *p variable, it cannot be used like any other variable.
6. Generally, a program has three sections such as input, processing, and output.
7. A 2D array is also referred as an array of arrays.
8. *x represents the memory area of variable named x.

0 1 2 3 4 5

Consider the following line of codes & answer from 9-12

```
char alpha[6];
alpha[5] = 50;
```

9. Which of the following is correct for the array, considering ASCII value?

(A) Assigns 50 as string in array (B) It shows memory segmentation error
(C) Assigns 2 as character in array (D) None of them

10. Which of the following is correct to initialize the array?

(A) alpha[]={a,b,c,d,e,f} (B) alpha []={'a','b','c','d','e','f ','/0'}
(C) alpha []={'a','b','c','d','e', 'f '} (D) alpha []={"abcdef"}

11. What should be the maximum index for the array?

(A) 3 (B) 6 (C) 4 (D) 5

12. What will be the memory address of alpha [4] if the starting address is A100 in hexadecimal?

(A) A10C (B) A104 (C) A112 (D) A108

Consider the following array & answer from 13-17

```
int arr[3][3];
```

1	2	4
2	6	7
3	5	8

13. What is the memory address of arr[1][1] if the starting address is AA00 in hexadecimal?

(A) AA04 (B) AA10 (C) AA01 (D) AA08

14. Which of the following is correct for array initialization?

(A) arr[3][3]={ 1,2,4,2,6,7,3,5,8} (B) arr[3][3]={ {1,2,4},{2,6,7},{3,5,8}}
(C) arr[3][3]={ (1,2,4),(2,6,7),(3,5,8)} (D) All of the above

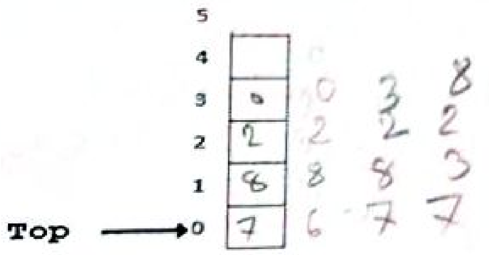
15. What will be the total size for the array if it is considered as 1D array?
(A) 6 (B) 7 (C) 8 (D) 9

16. What does &arr represent?
(A) It gives an error (B) It gives element of 1st index of the array
(C) It gives the memory location of the index 0 (D) None of them

17. What will be the sum of right diagonal elements of the array?
(A) 13 (B) 19 (C) 16 (D) None of them

Consider the following figure & answer from 18-23

```
1.  push(7);
2.  push(6);
3.  pop();
4.  push(8);
5.  push(2);
6.  push(0);
7.  pop( );
8.  push(3)
9.  pop( );
10. show( );
```



18. Which of the following is correct for line no. 6?
(A) Cannot be inserted because of underflow (B) Can be inserted at the top of the stack
(C) Cannot be inserted because of overflow (D) None of the above

19. After executing line no. 7, what is the value of Top?
(A) 2 (B) 4 (C) 3 (D) 0

20. Before executing line no. 8, what is the value of Top element?
(A) 2 (B) 3 (C) 4 (D) 0

21. Which of the following is correct?
(A) 7 is inserted at Top 1 (B) 3 is inserted at Top 3
(C) 0 never gets popped (D) None of them

22. What will be popped in line 9?
(A) 7 (B) 3 (C) 8 (D) 2

23. What is the correct sequence of output for line no. 10?
(A) 2,8,6 (B) 7,6,8,2 (C) 7,3,2,8 (D) 2,8,7

24. A linear queue is FULL when________ where rear is initialized as 0
(A) rear == Size – 1; (B) rear == Size; (C) front == rear+1; (D) front = Size

16 8 2 3
16 8/4
16 5

25. In queue data structure, elements are removed from____.
(A) front (B) rear (C) top (D) middle

26. What is the value of the prefix expression * / - 16 8 2 3?
(A) 16 (B) 22 (C) 12 (D) 15

27. Which of the following is/are ADT?
(A) Stack (B) Queue (C) Array (D) Both (A) & (B)

28. Array elements are always stored in ____________ memory address.
(A) Random (B) Consecutive (C) Both (A) & (B) (D) None

29. How many elements are to be shifted to delete the last element of an array with n elements?
(A) n (B) 0 (C) n+1 (D) n-1

30. Which is not the basic operation of Queue?
(A) isEmpty (B) isFull (C) FrontValue (D) None

31. If an element is inserted at index 0 of an array having ***n*** element, then number of shifts required is?
(A) n-1 (B) n (C) n+1 (D) 0

32. Which of the following is correct postfix expression of (x+y)/z?
(A) xy/z+ (B) xy+z/ (C) xzy+/ (D) Xyz/+

33. Which notation of expression is difficult for a computer to evaluate?
(A) Postfix (B) Infix (C) Prefix (D) All

34. Given an array `a[] = {2,4,7,5}`, what will the value of the statement `*(a+2)+3;` be ??
(A) 5 (B) 10 (C) 7 (D) 8

35. Which condition is checked before pushing data in a Stack?
(A) Stack Overflow (B) Stack Underflow (C) Both (D) None

Part – B (Answer 2 out of 3) **Pseudocode Writing** **[2 x 10 = 20 Marks]**

36. Write pseudocode to find the Multiplication of the boundary elements of a 2D array.

37. Write Pseudocode or Code for the following operations of a Circular Queue
   a. isEmpty( )- checks whether the Queue is empty or not
   b. isFull( ) - checks whether the Queue is full or not
   c. enQueue( )- add element to back i.e. at the rear
   d. frontValue ( ) - retrieve value of element from front
   e. evenOdd( ) – checks weather the front value is even or odd

38. Write pseudocode for evaluating a postfix expression.

1. A
2. A
3. B
4. A
5. B
6. A
7. A
8. B
9. C
10. C
11. D
12. B
13. B
14. D
15. D
16. C
17. D
18. B
19. A
20. A
21. B
22. B
23. D
24. A
25. A
26. C
27. D
28. B
29. B
30. D
31. B
32. B
33. B
34. B
35. A

06

```
array a[m][n];
var multi=1;
for(i=0..m)
    for(j=0..n)
        if(i==0 || i==m-1 || j==0 || j==n-1) then
            multi=multi* a[i][j];
        end-if
    end-for
end-for
print "multi:"
print multi;
```

07

```
struct Stack{
    var font=-1;
    var rear=-1;
    var n=5;
    array stack[n];
function isEmpty(){
    if(font==-1 && rear==-1) then
        return true;
    else
        return false;
    }
function isFull(){
    if((rear+1)%n==font) then
        return true;
    else
        return false;
        }
function Enqueue(var x){
    if (isfull()) then
        print "Queue is full";
    else if( rear==-1 & font ==-1) then
            rear = rear +1;
            font = font +1;

        else rear=(rear+1)%n;
        end-else
        set a[rear]= x;
}
function FontValue(){
        return a[rear];
    }
function evenOdd(){
    if( FontValue()%2==0) then
        print "Even";
    else
        print "Odd"
        }
};
```

```
(Q38) Define Global var postfixExp[100];

    Struct Stack{
        var top=-1, n=5;
        array a[n];
      function isfull(){
            if(top==n-1) then
               return true;
            else
               return false;
         }
      function isEmpty(){
          if(top==-1) then
            return true;
          else
            return false;
         }
      function push(var x){
         if(isfull())then
            print"Stack Overflow";
         else
              top = top+1;
            set a[top]=x;
          }
      function pop(){
            if(isEmpty()) then
              print"Stack Underflow";
            else
                var temp=a[top];
                  top=top-1;
                return temp;
            }
    };
```

```
Struct postFixEva{

    Create object Stack S1;
    function postfixEva(){
        Var a;
        Var b;
        for(i=0..strlen(postFixExp))
            if(postFixExp[i]>='0' && postFixExp[i]<='9') then
                s1.push(postFixExp[i] - '0');
            else
                a= s1.pop();
                b= S1.pop();
                Switch(postFixExp[i]){
                    Case '+':
                    S1.push(b+a); break;
                    Case '-':
                    S1.push(b-a); break;
                    Case '*':
                    S1.push(b*a); break;
                    Case '/':
                    S1.push(b/a); break;
                    Case '^':
                    S1.push(pow(b,a)); break;
                }
            end-else
        end-for
    print "postFix Expression result"
    print S1.pop();
};
```

Simulation

[2 x 15 = 30 Marks]

Part – C (Answer 2 out of 3)

39. a) Consider the following ***Linear Queue of size 5*** and complete the following operations on the queue. Write the elements of the queue along with the value of front and rear after each operation. In each operation, you must use the modified queue of the previous step. 7.5

Position	*0*	*1*	*2*	*3*	*4*	*Value of Front*	*Value of Rear*
Initial Queue	G	E	N			0	2
Dequeue()	G	E	N			0	2
Enqueue(I)	G	E	N	I		0	1
Enqueue(U)	G	E	N	I	U	0	0
Dequeue()	E	N	I	U		0	1
Enqueue(S)	E	N	I	U	S	0	0

b) Consider the following **Stack of size 6** and execute the operations mentioned below. Write the elements in the stack along with the **value of top** after executing the operations. In each operation, you must use the modified stack of the previous operation and show the value of top. 7.5

	Initial Stack	Push(B)	Push(A)	Pop()	Push(S)	Push(E)
	top = A	top = B	top = A	top = B	top = S	top = E
5			A		S	E
4		B	B	B	B	S
3	A	A	A	A	A	B
2	T	T	T	T	T	A
1	A	A	A	A	A	T
0	D	D	D	D	D	A

40. a) **Trace the output for the following code:** 5

```
#include <iostream>
using namespace std;
int main(){
    int x = 10;
    int *p1 = &x;
    int *p2 = &x;
    cout<<(*p2)++<<endl;
    cout<<*p1<<endl;
    x = 60;
    int *p3 = p1;
    cout<<++(*p1)<<endl;
    cout<<*p2<<endl;
    cout<<(*p1)++<<"  "<<++*p2<<"  "<<(*p3)++<<endl;
    return 0;
}
```

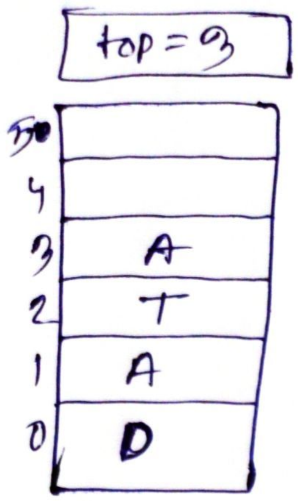
Output

(37)

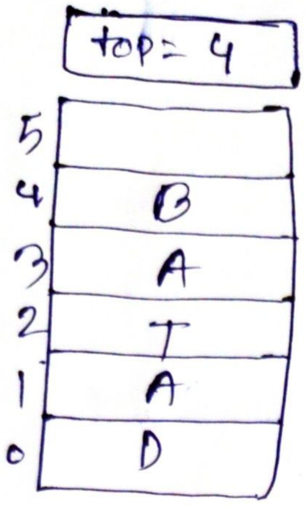
	0	1	2	3	4	Font	Rare
Initial Queue	G	E	N			0	2
Deque()		E	N			1	2
Enqueue(I)		E	N	I		1	3
Enqueue(U)		E	N	I	U	1	4
Dequeue()			N	I	U	2	4
Enqueue(S)			N	I	U	2	4

For this Linear Queue is already full.

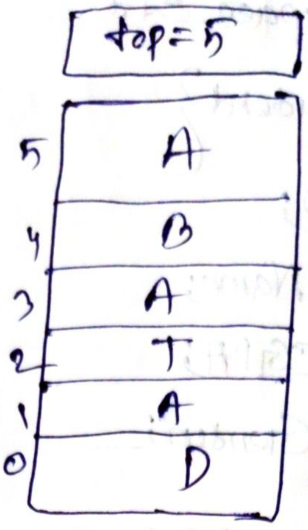
(b)



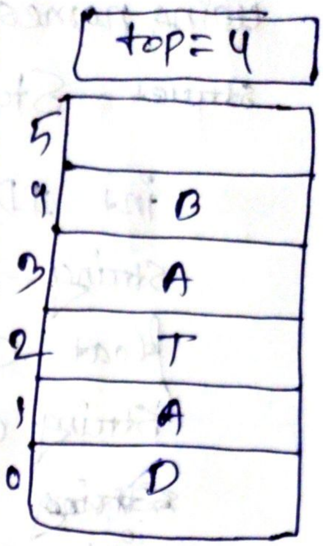
Initial Stack



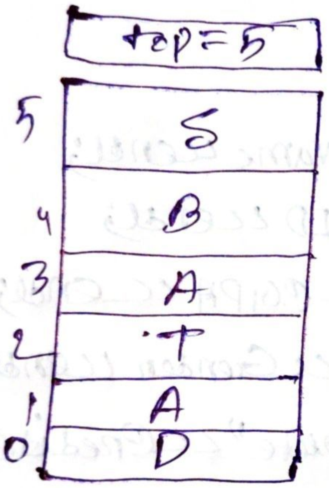
push(B)



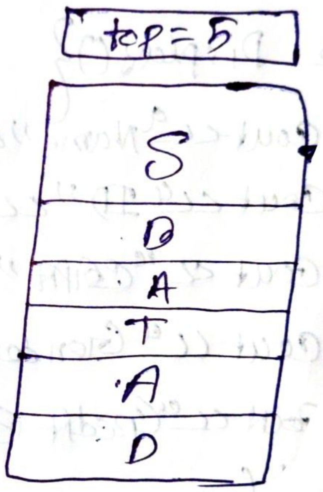
push(A)



pop()



push(S)



push(E)

For this Stack is Overflow.

4o

(a)

10

11

61

61

61 63 63

SET-

41. a) Draw simulation to check the ***validity*** of the following ***Infix Expression***. 5

b) Show simulation to ***convert*** it to ***postfix***. 10

A	+	{	(	B	*	C	)	-	D	}	/	(	E	-	F	+	G	)

a) A+{(B*C)-D}/(E-F+G)

= A+ {(B*C)-D}/(E-H)

= A+ {(B*C)-D}/I

= A+ B*C - D/I

= A+B*C-J.

= A+B*K

= A+L

= M

10

b) ABCDFFG+*-/-+

= ABCDEF/G+*--+

= ABCDEH

= ABCDF*H+--+

= ABCD-I+-+.

= ABC-J++

= AB+K+

(a)

(41) Expressions-

$A+\{(B*C)-D\}/(E-F+G)$

3	
2	(
1	~~{(~~

Stack

Scanned Symbol	Stack Current Situation
A	
+	
{	{
(	{, (
B	{, (
*	{, (
C	{, (
)	{
-	{
D	{
}	
/	
(	(
E	(
-	(
F	(
+	(
G	(
)	

→ Since the Stack in Empty So its a Valid expression

Infix Exp is:- (b)

A+{(B*C)-D}/(E-F+G)

Infix Exp/ Scaned Symbal	Stack	PostFix Exp
A		A
+	+	A
{	+{	A
(	+{(	A
B	+{(	AB
*	+{(*	AB
C	+{(*	ABC
)	+{	ABC*
-	+{-	ABC*
D	+{-	ABC*D
}	+	ABC*D-
/	+/	ABC*D-
(	+/(	ABC*D-
E	+/(	ABC*D-E
-	+/(-	ABC*D-E
F	+/(-	ABC*D-EF
+	+/(+	ABC*D-EF-
G	+/(+	ABC*D-EF-G
)	+/	ABC*D-EF-G+
		ABC*D-EF-G+/+

So. the PostFixExp = ABC*D-EF-G+/+

b) Define an array of structures. How can you store information of 10 students using this data structure? Describe with a suitable example. 10

b.

```
#include <iostream>
 using namespace std
 struct Student{
      int ID;
     String Name;
     float CGPA;
     String Gender;

      int creditcom;

   void Display(){
      cout<<"Name"<<Name<<endl;
     cout<<"ID"<< ID <<endl;
     cout<<"CGPA"<<CGPA<<endl;
     cout<<"Gender"<<Gender<<endl;
     cout<<"Credit Complete"<<Creditcom<<endl;
      }};


int main(){
     Student s1[10];
     s1[0]={1,"m1", 3.77, "male", 45};
    s2[1]={2, "m2", 3.86; "male"; 70};
   s3[2]={3, "m3"; 3.90, "Female", 20};
     :
     :
     :
  s[9]={10, "m10", 4, "male", 140};
      for(int i=0; i<10; i++){
            s1[i].Display();}
   }
```

SET- A

D (Answer All)	Analytical Question	[1 x 15 = 15 Marks]
answer of this part will be Evaluated for the following OBE Course Outcome: : Solve real-life problems using advanced concepts of data structure.		
Knowledge and proper use of Data Structure (5)	Problem Analysis (5)	Solution (5)

Name: Faisal Ahmed Fuad	ID: 23-50~~9~~16-1

2. A ball-by-ball scorecard of a cricket match can be displayed where the row number represents the number of overs, and the column number represents the number of balls for a particular over. Any element of the score card represents either a run of the scored ***(0, 1, 2, 3, 4, 6)*** or a wicket ***(w)*** on that ball. You need to find the followings:

- Define the proper data structure to solve the problem and represent its syntax.
- Write the steps for analysis of the problem to find the maximum run score.
- Write the solution (pseudo code) for counting the number of maiden overs. (Maiden over is an over where no runs were scored.)

[Hint: A match can be played for ***N*** overs and each over has **6** balls. For simplicity, we are assuming that there were no wide-balls and no no-balls.***]***

Sample Input:	**Sample Output:**
N=5 Score Card: 1 2 0 0 4 0 w 0 0 2 1 1 0 0 0 0 0 0 1 0 w 2 0 2 0 0 4 4 4 0	Maximum runs scored in over: 5 Number of maiden over: 1

```
#include <iostream>

using namespace std;

struct Stack2{
  int top;
  int arr[100];

  Stack2(){
    top=-1;
    for(int i=0; i<5; i++){
      arr[i]=0;
    }
  }


  bool isEmpty(){
    if(top==-1){
      return true;
    }
    else{
      return false;
    }
  }


  bool isFull(){
    if(top==4){
      return true;
```

```
   }
   else{
      return false;
   }
}


void push(int value){
   if(isFull()){
      cout<<"Stack Overflow"<<endl;
   }
   else{
      top++;
      arr[top]=value;
      }
}


int pop(){
   if(isEmpty()){
      cout<<"Stack Underflow"<<endl;
      return 0;
   }
   else{
      int topValue=arr[top];
      arr[top]=0;
      top--;
      return topValue;
   }
```

```
    }
};


struct Over {
    Stack2 s2;
    char eachOverRunWicket[6];
    int totalRunInEachOver[6];

    Over() {
        for(int i = 0; i < 6; i++) {
            eachOverRunWicket[i] = '0';
            totalRunInEachOver[i] = 0;
        }
    }

    int TotalRunInEachOver() {
        int a = 0;
        int b = 0;
        int c=0;
        for(int i = 0; i < 6; i++) {
            if (eachOverRunWicket[i] >= '0' && eachOverRunWicket[i] <= '9'){
                s2.push(eachOverRunWicket[i] - '0');

            }
            if(s2.top==1){
                a=s2.pop();
                b=s2.pop();
                c=a+b;
        s2.push(c);

      }
    }


    int popedvalues=s2.pop();
    return popedvalues;


  }



};

int main() {
  int n;
  cout << "Enter the number of overs played: ";
  cin >> n;

  Over o[n];
  cout << "Enter runs and wickets for each over:"<<endl;
  for(int i = 0; i < n; i++) {
    cout << "Over " << i + 1 << ":";
    for(int j = 0; j < 6; j++) {
      cin >> o[i].eachOverRunWicket[j];
    }
  }
```

```
cout<<endl;
cout<<endl:
cout << "Runs and wickets for each over:" << endl;
for(int i = 0; i < n; i++) {
   cout << "Over " << i + 1 << ":";
   for(int j = 0; j < 6; j++) {
      cout << o[i].eachOverRunWicket[j] << " ";
   }
   cout << endl;
}
int num=0;
int max=0;
int over=0;
cout << "total Runs each over:" << endl;
for(int i = 0; i < n; i++) {
   cout << "Over " << i + 1 << ":";
   cout << o[i].TotalRunInEachOver() << " ";
   cout << endl;
}

cout<<endl;
cout<<endl;


cout << "maximum Runs scoreed in over: ";
for(int i = 0; i < n; i++) {
   if(max<o[i].TotalRunInEachOver()){
```

```
            max=o[i].TotalRunInEachOver();
            over=i+1;
        }


    }
    cout << over<< " ";
    cout << endl;


    cout << "Number of maidenOver: ";
    for(int i = 0; i < n; i++) {


        if(o[i].TotalRunInEachOver()==0){
            num++;
        }
    }
    cout << num << " ";
        cout << endl;


    return 0;
}
```