USN □□□□□□□□□□

Internal Assessment Test 1 – December 2023

Sub:	**DATA STRUCTURES AND APPLICATIONS**			Sub Code:	**BCS304**	Branch:	**AIML**	
Date:	**21/12/23**	Duration:	**90 minutes**	Max Marks:	**50**	Sem/Sec:	**III -A, B, C**	**OBE**

Answer any FIVE FULL Questions			**MARKS**	**CO**	**RBT**
1	a	Differentiate between a linear and a nonlinear data structure with suitable examples.	7M	1	L2
	b	Consider two polynomials, A(x)=9x^{15}+3x^{6}+5 and B(x)=x^{7}+7x^{4}+1 show diagrammatically how these two polynomials can be stored in an array	3M	1	L3
2	a	Differentiate between structure and unions with suitable examples.	5M	1	L2
	b	What are the various memory allocation techniques? Explain how memory can be dynamically allocated using malloc().	5M	1	L2
3	a	Define a string and explain any 3 built in functions used with strings in C with examples.	5M	1	L2
	b	Explain the representation of a sparse matrix and list any three functions of Sparse matrix ADT.	5M	1	L2
4	a	Write a C program to implement a stack with push(),pop() and display().	6M	1	L3
	b	Define 1)Polish Notaion and 2)Reverse Polish Notation with examples.	4M	1	L1
5	a	Write and apply the algorithm to convert the below infix expression to a postfix expression (((a/b)-c)+(d*e))-(a*c)	5M	1	L3
	b	Evaluate the following postfix expressions, 1. abc+*de/- where a=5, b=6, c=2, d=12, e=4 2. AB+CDE-+/ where A=5, B=5, C=4, D=7, E=1	5M	1	L3
6	a	Write the Knuth, Morris, Pratt pattern matching algorithm and apply the same to search the pattern ‘abcdabcy’ in the text ‘abcxabcdabxabcdabcaabcdabcyy’	8M	1	L3
	b	Data are pushed to (PUSH operation) and popped from (POP operation) a stack in the following order: PUSH 3; TOP; PUSH 7; TOP; PUSH 6; PUSH 9; TOP; POP; POP; TOP; where the PUSH inserts an item onto stack , POP deletes an item from the stack and TOP returns top position element of the stack. Write the values returned by TOP for the sequence of operations above.	2M	1	L3

CI **CCI** **HOD**

---All the Best---

USN

Sub:	**DATA STRUCTURES AND APPLICATIONS**					Sub Code:	**BCS304**
Date:	**21.12.23**	Duration:	**90 minutes**	Max Marks:	**50**	Sem/Sec:3 A,B,C	

Scheme and Solutions

1 a

Differentiate between a linear and a nonlinear data structure with suitable examples.

Answer:-

The non-primitive data structures is further classified into

1. Linear Data Structure

2. Non-linear Data Structure **2*3=6**

1. Linear Data Structure: A data structure is said to be linear if its elements form a sequence or a linear list. There are basically two ways of representing such linear structure in memory. a. One way is to have the linear relationships between the elements represented by means of sequential memory location. These linear structures are called arrays.

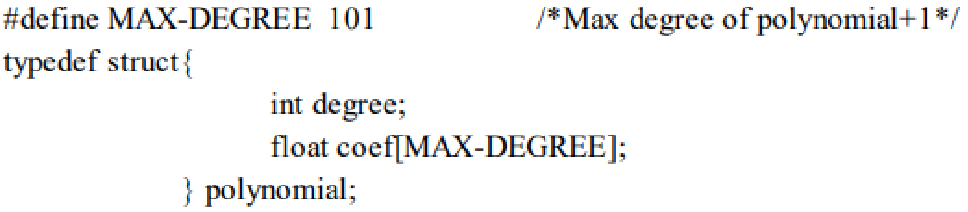
b. The other way is to have the linear relationship between the elements represented by means of pointers or links. These linear structures are called linked lists. The common examples of linear data structure are Arrays, Queues, Stacks, Linked lists

2. Non-linear Data Structure: A data structure is said to be non-linear if the data are not arranged in sequence or a linear. The insertion and deletion of data is not possible in linear fashion. This structure is mainly used to represent data containing a hierarchical relationship between elements. Trees and graphs are the examples of non-linear data structure.

b

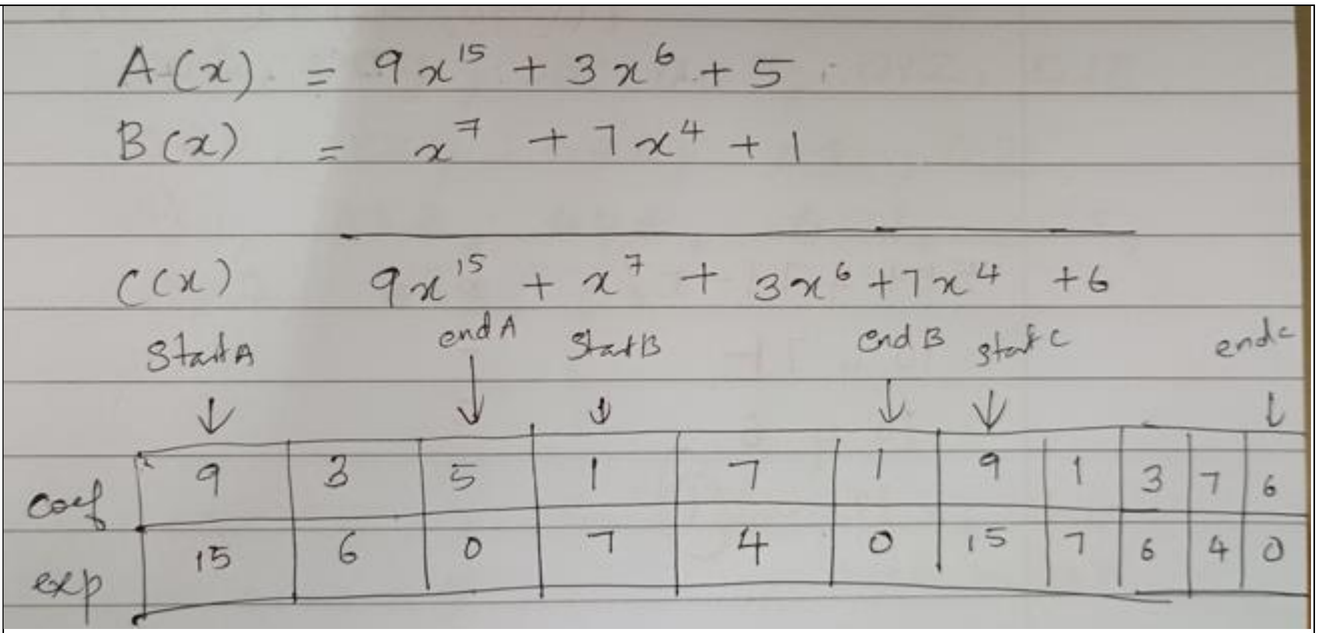
Write the Structure representation of a polynomial. Consider two polynomials, A(x)=9x^{15}+3x^{6}+5 and B(x)=x^{7}+7x^{4}+1 and show diagrammatically how these two polynomials and its resultant are represented after Addition.

Answer: - Structure Representation 2 M



```
#define MAX-DEGREE  101                /*Max degree of polynomial+1*/
typedef struct{
                        int degree;
                        float coef[MAX-DEGREE];
                } polynomial;
```

Representation -2M



$A(x) = 9x^{15} + 3x^{6} + 5$

$B(x) = x^{7} + 7x^{4} + 1$

$C(x) \quad 9x^{15} + x^{7} + 3x^{6} + 7x^{4} + 6$

StartA
endA
StartB
endB
startC
endc

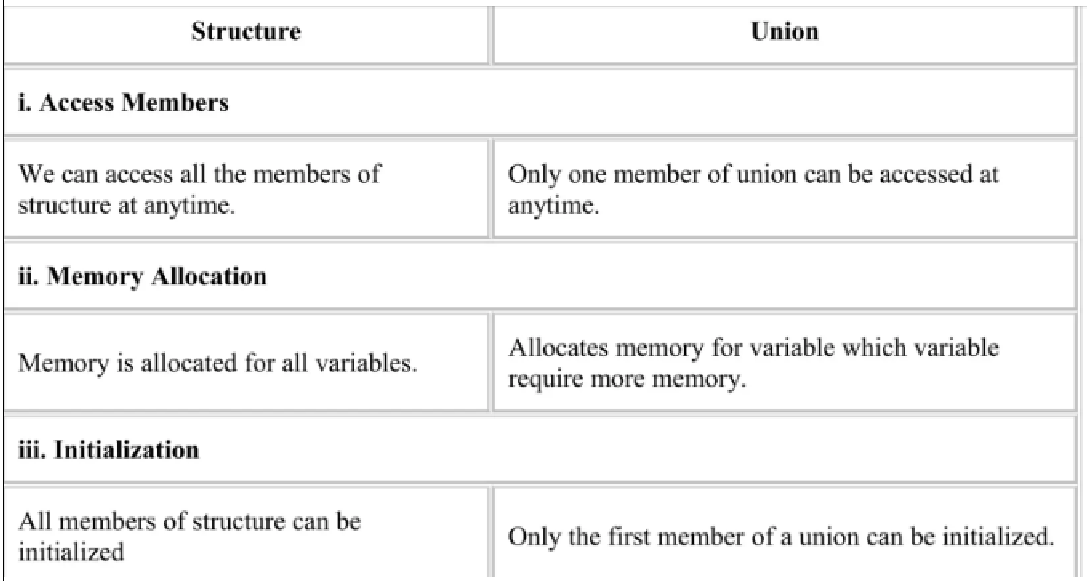
coef	9	3	5	1	7	1	9	1	3	7	6
exp	15	6	0	7	4	0	15	7	6	4	0

2 a

Differentiate between structure and unions with suitable examples.

Differences -3M

Example and Syntax -2M



Structure	Union
i. Access Members	
We can access all the members of structure at anytime.	Only one member of union can be accessed at anytime.
ii. Memory Allocation	
Memory is allocated for all variables.	Allocates memory for variable which variable require more memory.
iii. Initialization	
All members of structure can be initialized	Only the first member of a union can be initialized.

b

What are the various memory allocation techniques? Explain how memory can be dynamically allocated using malloc().

Answer:-

The various memory allocation techniques are

1.Static Memory Allocation

2.Dynamic Memory Allocation +Explanation on these- **1X 2=2M**

Malloc() -Definition,Syntax and Example **3M**

malloc( ): The function malloc allocates a user- specified amount of memory and a pointer to the start of the allocated memory is returned. If there is insufficient memory to make the allocation, the returned value is NULL.

Syntax: data_type *x; x= (data_type *) malloc(size);

Where, x is a pointer variable of data_type size is the number of bytes

Ex: int *ptr; ptr = (int *) malloc(100*sizeof(int));

3 a **Define a string and explain any 3 built in functions used with strings in C with examples.**

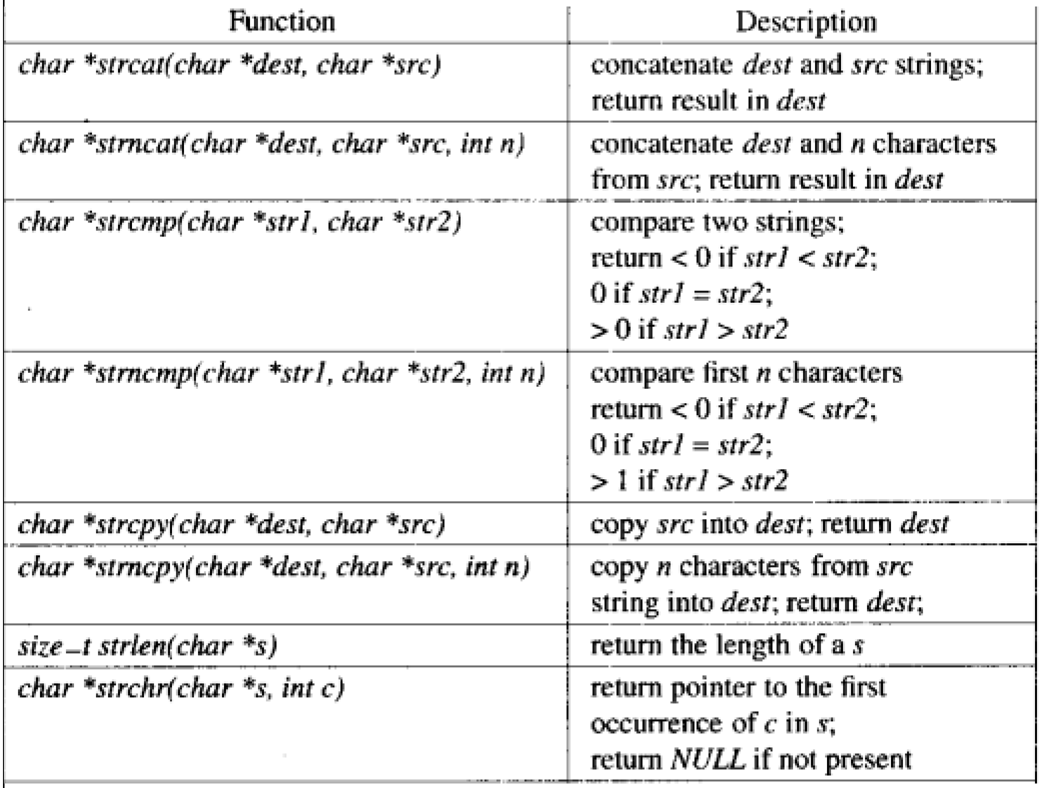
Answer:-

Definition-2M

A finite sequence S of zero or more Characters is called string.

In C, the strings are represented as character arrays terminated with the null character \0

C String Functions(Any 3) 3x1=3M



Function	Description
*char *strcat(char *dest, char *src)*	concatenate *dest* and *src* strings; return result in *dest*
*char *strncat(char *dest, char *src, int n)*	concatenate *dest* and *n* characters from *src*; return result in *dest*

*char *strcmp(char *str1, char *str2)*	compare two strings;
	return < 0 if *str1* < *str2*;
	0 if *str1* = *str2*;
	> 0 if *str1* > *str2*

*char *strncmp(char *str1, char *str2, int n)*	compare first *n* characters return < 0 if *str1* < *str2*; 0 if *str1* = *str2*; > 1 if *str1* > *str2*
*char *strcpy(char *dest, char *src)*	copy *src* into *dest*; return *dest*
*char *strncpy(char *dest, char *src, int n)*	copy *n* characters from *src* string into *dest*; return *dest*;
*size_t strlen(char *s)*	return the length of a *s*
*char *strchr(char *s, int c)*	return pointer to the first occurrence of *c* in *s*; return *NULL* if not present

b Explain the representation of a sparse matrix and list any three functions of Sparse matrix ADT.

Answer:-

Representation of Sparse Matrix-2M

Triple Representation <r,c,value> and its advantages.

Example of converting sparse matrix to Triple format.

Any three functions of Sparse Matrix-3M

Transpose()

Add()

Create()

Multiply()

4 a Write a C program to implement a stack with push(),pop() and display() functions.

```
#include<stdio.h>
int stack[100],choice,n,top,x,i;
void push(void);
void pop(void);
void display(void);
```

```
int main()
{
    top=-1;
    printf("\n Enter the size of STACK[MAX=100]:");
    scanf("%d",&n);
    printf("\n\t STACK OPERATIONS USING ARRAY");
    printf("\n\t--------------------------------");
    printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t 4.EXIT");
    do
    {
        printf("\n Enter the Choice:");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            {
                push();
                break;
            }
            case 2:
            {
                pop();
                break;
            }
            case 3:
            {
                display();
                break;
            }
            case 4:
            {
                printf("\n\t EXIT POINT ");
                break;
            }
            default:
            {
                printf ("\n\t Please Enter a Valid Choice(1/2/3/4)");
            }

        }
    }
    while(choice!=4);
    return 0;
}
void push()
{
    if(top>=n-1)
    {
        printf("\n\tSTACK is over flow");
    }
    else
    {
        printf(" Enter a value to be pushed:");
        scanf("%d",&x);
        top++;
        stack[top]=x;
    }
}
void pop()
{
    if(top<=-1)
    {
        printf("\n\t Stack is under flow");
    }
    else
    {
        printf("\n\t The popped elements is %d",stack[top]);
        top--;
    }
}
void display()
{
    if(top>=0)
    {
        printf("\n The elements in STACK \n");
        for(i=top; i>=0; i--)
            printf("\n%d",stack[i]);
        printf("\n Press Next Choice");
    }
    else
    {
        printf("\n The STACK is empty");
    }

}
```

Marks 6M

b

Define 1)Polish Notation and 2)Reverse Polish Notation with examples.

Answer: -

1)Polish Notation -Prefix Expression

Definition with examples-2M

2)Reverse Polish Notation

Definition with examples -2M

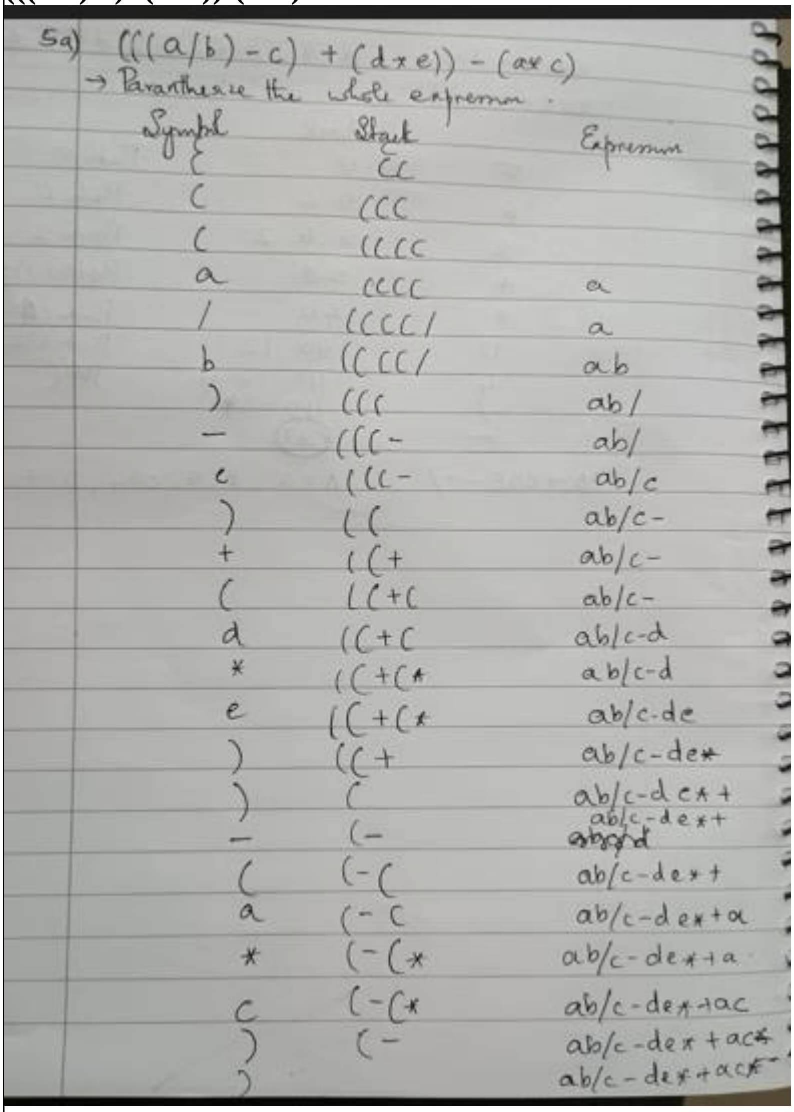
5 a

Write and apply the algorithm to convert the below infix expression to a postfix expression (((a/b)-c)+(d*e))-(a*c)

Answer:-

Infix to postfix Algorithm -2M

(((a/b)-c)+(d*e))-(a*c) Solution-3M



5a) $(((a/b)-c)+(d*e))-(a*c)$

→ Paranthesize the whole expression.

Symbol	Stack	Expression
(	((	
(	(((	
(	((((	
a	((((	a
/	((((/	a
b	((((/	ab
)	(((	ab/
-	(((-	ab/
c	(((-	ab/c
)	((	ab/c-
+	((+	ab/c-
(	((+(	ab/c-
d	((+(	ab/c-d
*	((+(*	ab/c-d
e	((+(*	ab/c-de
)	((+	ab/c-de*
)	(	ab/c-de*+
-	(-	ab/c-de*+
(	(-(	ab/c-de*+
a	(-(	ab/c-de*+a
*	(-(*	ab/c-de*+a
c	(-(*	ab/c-de*+ac
)	(-	ab/c-de*+ac*
)		ab/c-de*+ac*-

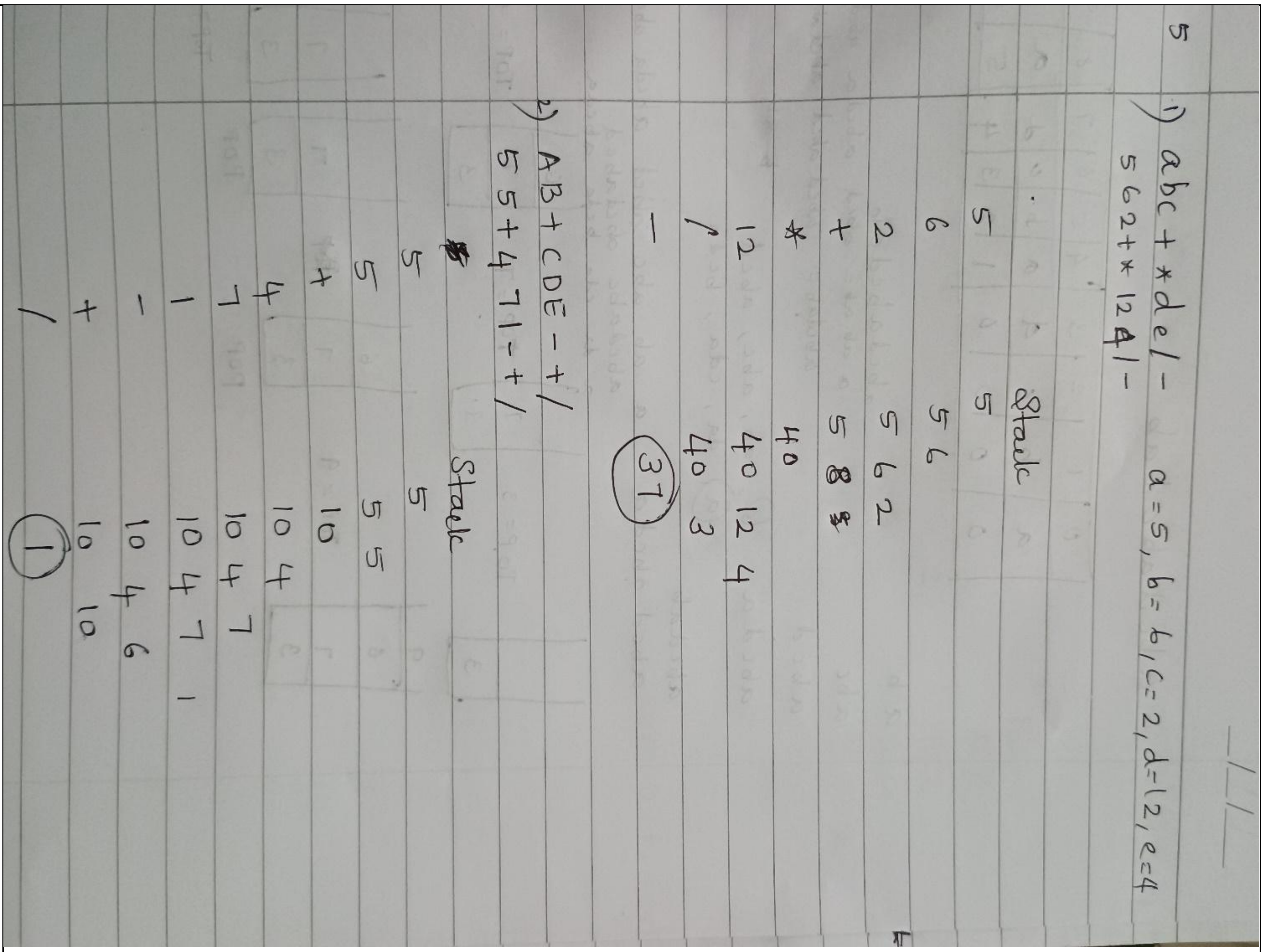
b

Evaluate the following postfix expressions,

1. abc+*de/- where a=5, b=6, c=2, d=12, e=4

2. AB+CDE-+/ where A=5, B=5, C=4, D=7, E=1

Answer 1: 37 2.5M
Answer 2: 1 2.5M



5

i) abc+*de/-   a=5, b=6, c=2, d=12, e=4

562+* 124/-

	Stack
5	5
6	5 6
2	5 6 2
+	5 8
*	40
12	40 12 4
/	40 3
-	37

2) AB+CDE-+/

55+4 71-+/

	Stack
5	5
5	5 5
+	10
4	10 4
7	10 4 7
1	10 4 7 1
-	10 4 6
+	10 10
/	1

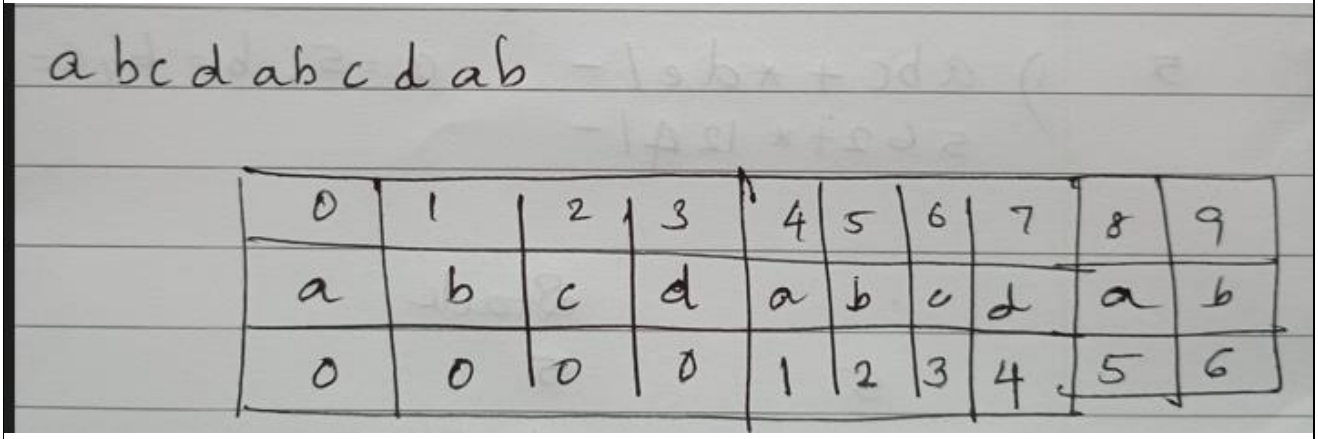
6 a

Write the Knuth, Morris, Pratt pattern matching algorithm and compute the failure function for the pattern: 'abcdabcdab'.

Answer:-

Algorithm 4M

Failure Function Calculation 2M



abcdabcdab

0	1	2	3	4	5	6	7	8	9
a	b	c	d	a	b	c	d	a	b
0	0	0	0	1	2	3	4	5	6

b

Write the code snippets for isFull() and isEmpty() of a Stack.
Consider a stack where the data are pushed to (PUSH operation) and popped from (POP operation) in the following order:
PUSH 3; TOP; PUSH 7; TOP; PUSH 6; PUSH 9; TOP; POP; POP; TOP;
where the PUSH inserts an item onto the stack, POP deletes an item from the stack and TOP returns the top position element of the stack.
Write the values returned by TOP for the sequence of operations above.

Answer

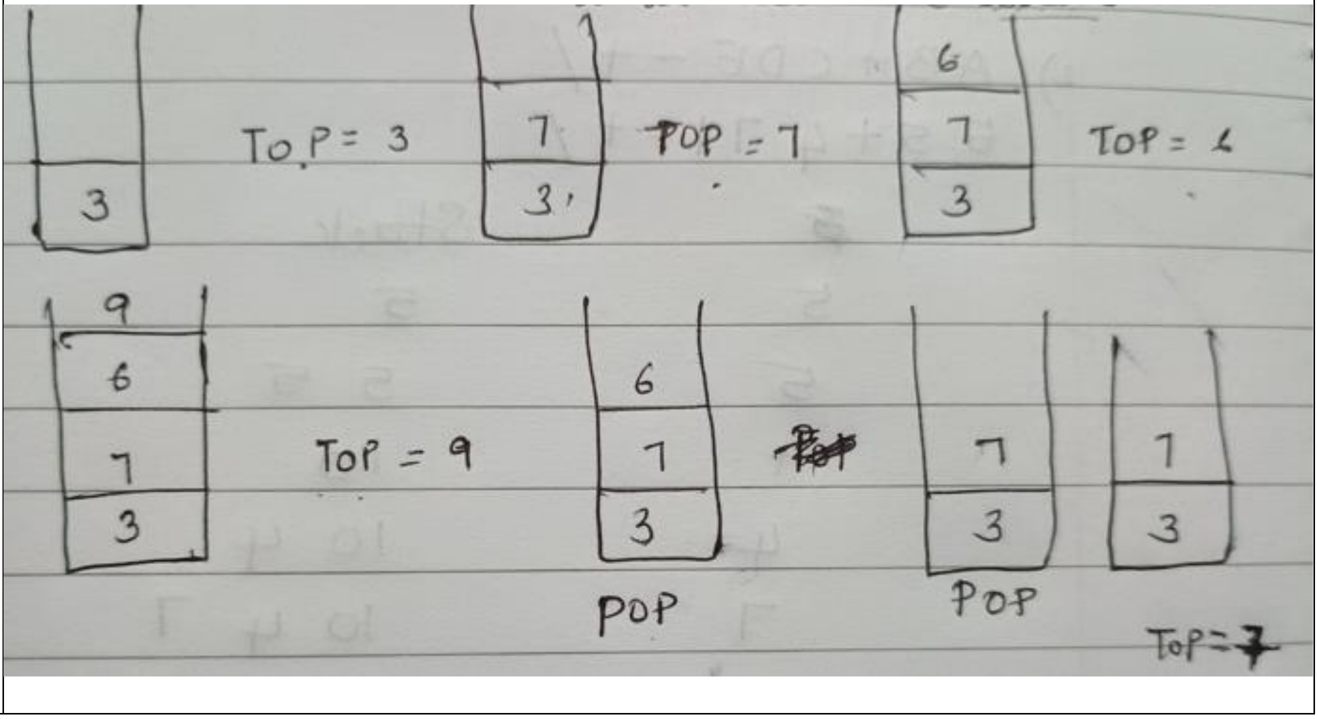
isFull()-1M isEmpty()-1M

```
bool IsFull() {
   if(top == SIZE-1)  //Here, SIZE is the total stack capacity
      return true;
   else
      return false;
}

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


}
```

Stack -2M



CI **CCI** **HOD**

--All the Best--