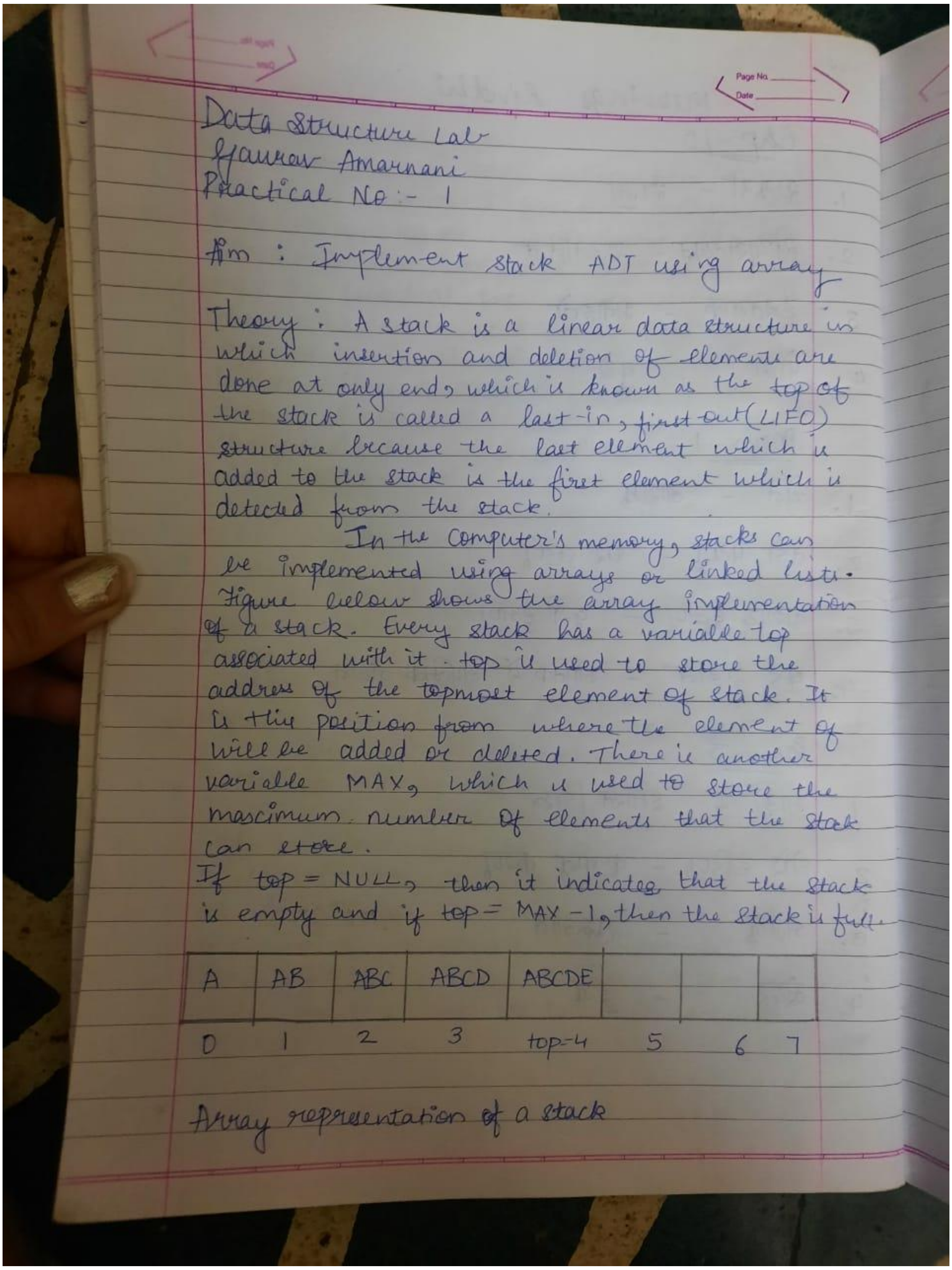
DS Assignment 1 - Gaurav Amarnani DSE CMPN.



Data Structure Lab
Gaurav Amarnani
Practical No :- 1

Aim : Implement stack ADT using array

Theory : A stack is a linear data structure in which insertion and deletion of elements are done at only end, which is known as the top of the stack is called a last-in, first-out (LIFO) structure because the last element which is added to the stack is the first element which is detected from the stack.

In the computer's memory, stacks can be implemented using arrays or linked lists. Figure below shows the array implementation of a stack. Every stack has a variable top associated with it. top is used to store the address of the topmost element of stack. It is the position from where the element of will be added or deleted. There is another variable MAX, which is used to store the maximum number of elements that the stack can store.

If top = NULL, then it indicates that the stack is empty and if top = MAX - 1, then the stack is full.

A	AB	ABC	ABCD	ABCDE			
0	1	2	3	top-4	5	6	7

Array representation of a stack

A stack supports three basic operations: pop and peek. The push operation adds an element to the top of the stack. The pop operation removes the element from the top of stack. The peep operations returns the value of the topmost element of the Stack.

However, before inserting an element in the stack, we must check for overflow conditions. An overflow occurs when we try to insert an element into a stack that is already full.

Similarly, before deleting an element from the stack, we must check for overflow conditions. An underflow conditions occurs when we try to delete an element from a stack that is already empty.

Algorithm :

```
1] For Push  :  step 1 : If TOP = MAX - 1
                            Print "Overflow"
~~Step 1: Start~~           Goto step 4
                         [End of If]
                 step 2 : SET TOP = TOP+1
                 step 3 : Set STACK[TOP] = VALUE
                 step4 :  END
```

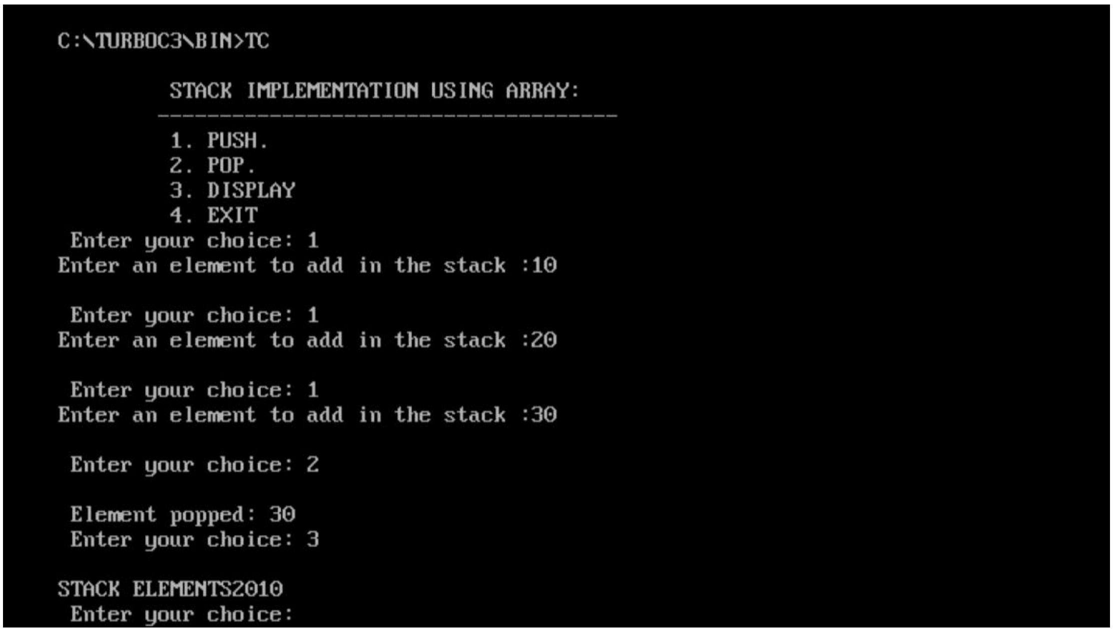
```
2] POP : Step 1: If Top = Null
                   Print "Underflow"
                   Go to Step 4
                 [End of If]
         Step 2 : SET VAL = STACK[TOP]
         Step3 : SET TOP = TOP - 1
         Step4 : END

3] PEEK : Step 1: If TOP = NULL
                    Print "Stack'is Empty"
                    Goto step 3
          step2 : RETURN STACK[TOP]
          step3 : END
```

Program:

```
 #include<stdio.h>
#include<conio.h>
#define SIZE 5
int stack[SIZE], c, t, e, i;
int is_full();
int is_empty();
void push();
void pop();
void display();

void main() {
  t = -1;
  printf("\n\t STACK IMPLEMENTATION 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 your choice: ");
    scanf("%d", &c);
    switch(c) {
      case 1: {
         push();
         break;
      }
      case 2: {
         pop();
         break;
      }
      case 3: {
         display();
         break;
      }
      case 4: {
```

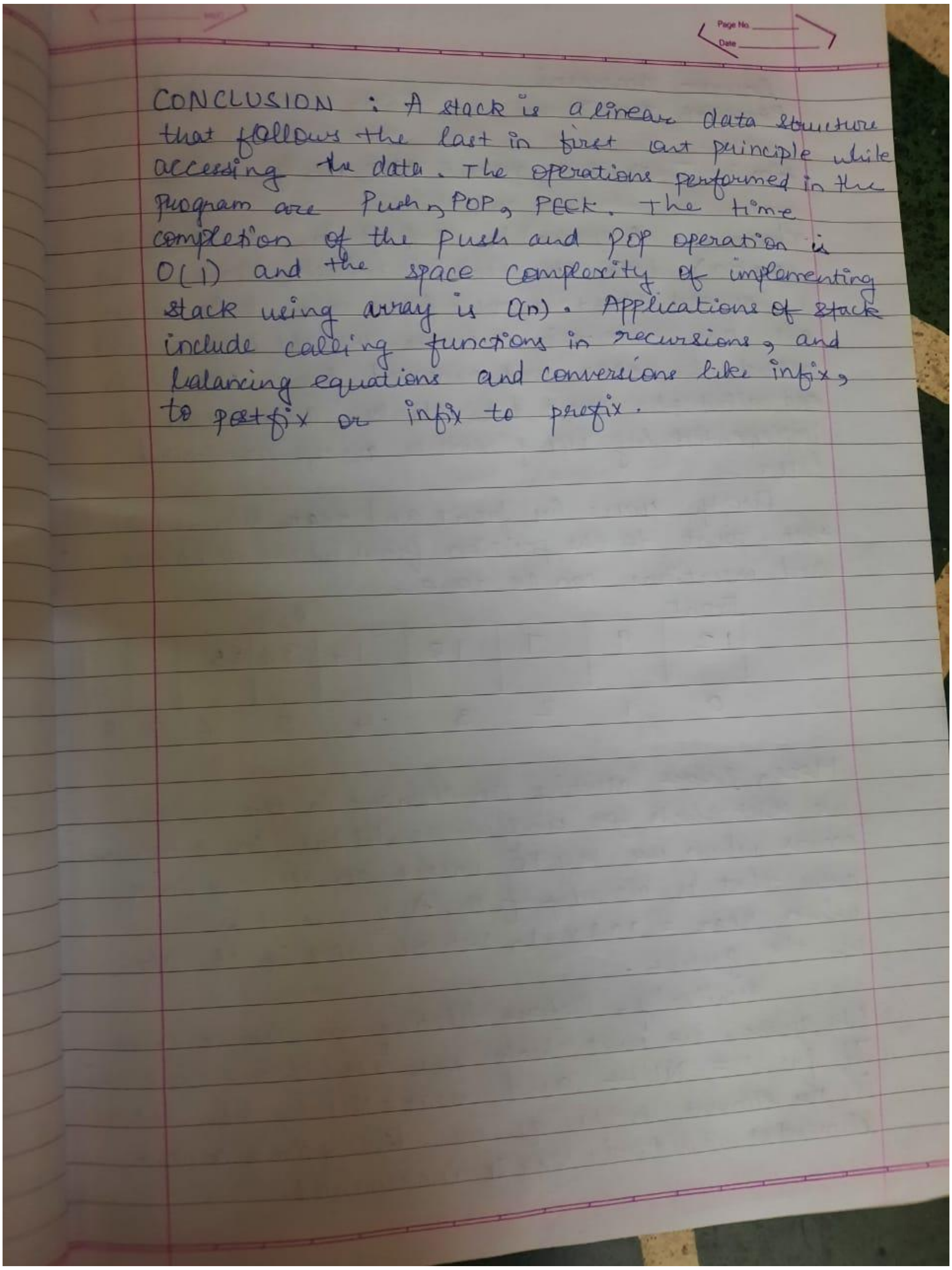
```
     printf("\n\t EXIT POINT.");
     break;
   }
   default: {
     printf("\n\t Plase enter a valid choice (1/2/3/4).");
   }
  }
 } while(c!=4);
}

 int is_full() {
  if(t == SIZE - 1)
   return(1);
  return(0);
 }

 int is_empty() {
  if(t == -1)
   return(1);
  return(0);
 }

 void push() {
  if(is_full() == 1)
   printf("\n\t STACK is full.");
  else {
   printf("Enter an element to add in the stack :");
   scanf("%d", &e);
   t++;
   stack[t] = e;
  }
 }

 void pop() {
  if(is_empty() == 1)
   printf("\n\t STACK is empty.");
  else {
    printf("\n Element popped: %d", stack[t]);
    t--;
  }
}

void display() {
  if(is_empty() == 1)
    printf("\n\t STACK is empty.");
  else {
    printf("\nSTACK ELEMENTS");
    for(i = t; i >= 0; i--)
       printf("%d",stack[i]);
  }
}
```

Output:



```
C:\TURBOC3\BIN>TC

         STACK IMPLEMENTATION USING ARRAY:
        -------------------------------------
         1. PUSH.
         2. POP.
         3. DISPLAY
         4. EXIT
 Enter your choice: 1
Enter an element to add in the stack :10

 Enter your choice: 1
Enter an element to add in the stack :20

 Enter your choice: 1
Enter an element to add in the stack :30

 Enter your choice: 2

 Element popped: 30
 Enter your choice: 3

STACK ELEMENTS2010
 Enter your choice:
```

Conclusion:



Page No.

Date

CONCLUSION : A stack is a linear data structure that follows the last in first out principle while accessing the data. The operations performed in the program are Push, POP, PEEK. The time completion of the push and pop operation is O(1) and the space complexity of implementing stack using array is O(n). Applications of stack include calling functions in recursions, and balancing equations and conversions like infix, to postfix or infix to prefix.