Assignment No 4

Q1] Define the term stack overflow and underflow.

→ Stack Overflow:-

When we are trying to push element on the stack which is already full then stack overflow condition occurs.

```
Top = 3   40        Stack Size: 4.
      2   30
      1   20
      0   10
```

Stack Underflow:-

When we try to pop the element on stack which is already empty, stack underflow condition occurs.

```
      3             Stack Size: 4.
      2
      1
      0
Top = -1
```

Q2] Explain concept of recursion with example.

→ • Function calling itself is known as recursion.
• It is defined as calling function on its own body.

* Example:

```
#include <stdio.h>
#include <conio.h>

int fib(int num){
```

```
if (num == 0){
return 0; }
else if (num == 1){
return 1;
}
else {return (fib(n-1) + fib(n-2));
}}
void main(){
int n, i;
printf("Enter number of terms: ");
scanf("%d", &n);
for(i=0; i<n; i++){
printf(" %d", fib(i));
}
getch();
}
```

Q3] Write algorithm for performing push and pop operation on stack.

→ * Algorithm for push operation on stack:

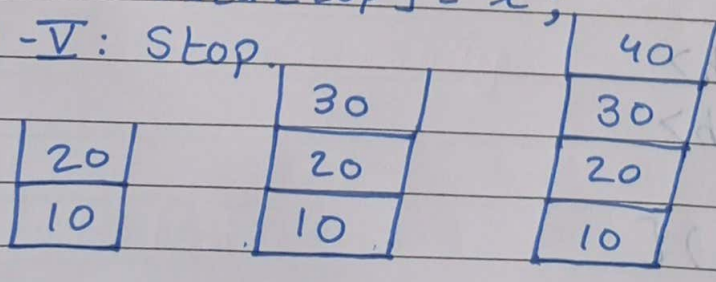
Step - I: Start.

Step - II: if (top == size-1), stack ~~empty~~ full. else, goto step III.

Step - III: Increment top by one:
i.e: top += 1;

Step - IV: Push new element:
data[top] = x;

Step - V: Stop.



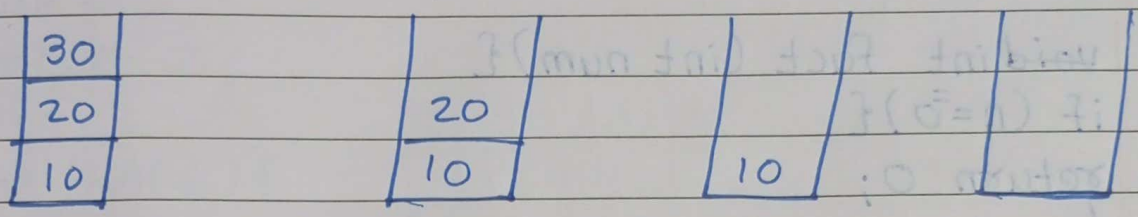
* Algorithm for POP operation on stack:

Step - I : Start

Step - II : if (top == -1), stack empty, else go to step - III.

Step - III : Decrement top by one:
top -= 1;

Step - IV : Stop.



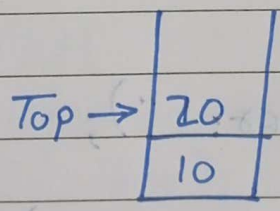
Q4] Define Stack and its operations.

→ *Stack:

It is linear data structure in which elements are inserted and deleted from one end.

It is LIFO structure.

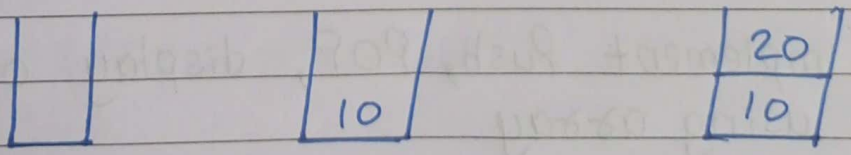
*Eg:



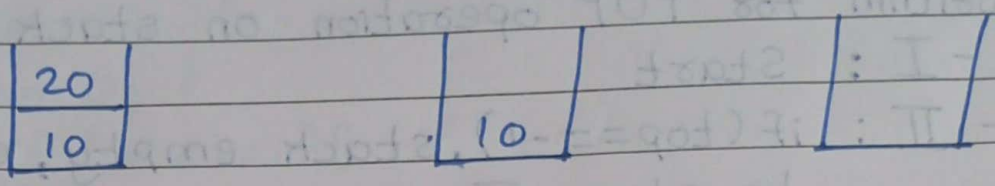
Size : 3

* Operations Performed On Stack:

i] Push: Used to insert data.



ii] POP: Used to delete data



Q5] WAP to calculate factorial using recursion.

→
```
#include <stdio.h>
#include <conio.h>

~~void~~ int fact (int num){
if (n==0){
return 0;
}
if (n==1){
return 1;
}
else{
return n * fact (n-1);
}}
void main (){
int a;
printf ("Enter a number: ");
scanf ("%d", &a);
printf("Factorial: %d", fact(a));
getch ();
}
```

Q6] WAP to implement Push, POP, display operation on stack using array.

→
```
#include <stdio.h>
#include <conio.h>
int data [5]; top;
```

```
void init(){
top =-1;
}
int
void empty(){
if (top==-1){
return 1;
}
else{
return 0;
}}
int full(){
if (top == size-1){
return 1;
}
else{
return 0;
}}
void Push(){
int x;
if(!full()){
printf("Enter element: ");
scanf("%d", &x);
top += 1;
data[top] = x;
}
else{
printf("Stack is full!");
}}
void POP(){
int x;
if(!empty()){
```

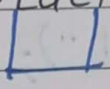
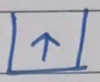
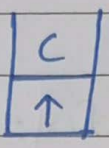
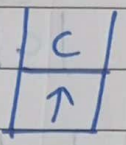
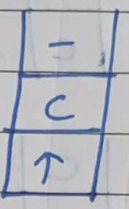
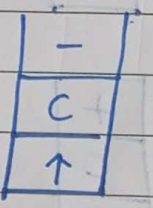
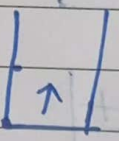
```
x= data[top];
top -=1;
printf("%d Popped! ", x);
}
else{
printf("Stack is empty! ");
}}
void display (){
int i;
for (i= top; i>=0; i--){
printf("%d ", data[top]);
}}
void main (){
int c;
init ();
do{
printf ("Enter choice: \n 1] Push \n 2] Pop \n 3] Display
\n 4] Exit \n choice: ");
scanf ("%d", &c);
switch (c){
case 1:
push();
break;
case 2:
pop ();
break;
case 3:
display ();
break;
case 4:
exit (0);
break;
```

```
default:
printf("Invalid Choice!");
}} while(1);
getch();
}
```

Q7] Convert infix to postfix: ((A+B)*D)↑(E-F)

→

I/P	Stack	O/P
(	(	—
(	((	—
A	((	A
+	+ ((	A
B	+ ((	AB
)	(	AB+
*	* (	AB+
D	* (	AB+D

I/P	Stack	O/P
)		AB+D*
↑		AB+D*
C		AB+D*
E		AB+D*E
−		AB+D*E
F		AB+D*EF
)		AB+D*EF−

*Post Fix Expression: AB+D*EF−↑

Q8] Evaluate following postfix expression:

5, 6, 2, +, *, 12, 4, /, −

→

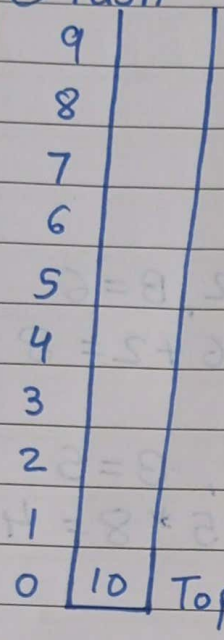
I/P	Stack	Operation.
5	5	
6	6 5	
2	2 6 5	
+	8 5	A = 2, B = 6 B + A = 6 + 2 = 8
*	40	A = 8, B = 5 B * A = 5 * 8 = 40
12	12 40	
4	4 12 40	
/	3 40	A = 4, B = 12 B / A = 12 / 4 = 3
-	37	A = 3, B = 40 B - A = 40 - 3 = 37

∴ Answer: 37.

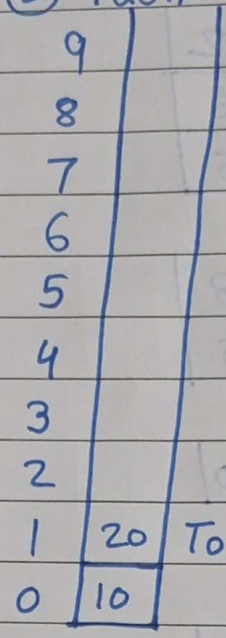
Q9] Show effect of Push and POP on stack using diagram: (size 10)

i] Push 10
ii] Push 20
iii] POP
iv] Push 30.

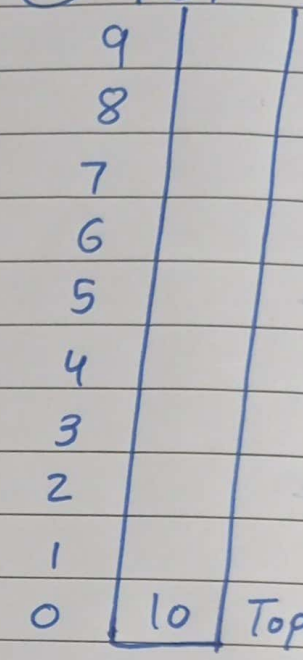
→ ① Push 10



② Push 20



③ POP



④ Push 30