2) Dynamic memory allocation	The state of the same of the
A least the second of the seco	
#include < stdio.h	-
void Mallor (int)	
roid (allow (int);	
(Childy bx) thing	
void man ()	
Elle De La reside Colore (the) = eta	
int no (three notes of each and	
printf (" exter the number of elements n: \n");	
scool (11/di, 2n) and with it	
Mallor (n): so demode alt al Itaria	
(alloe(n); (phi noi : 0 = i) and	
3 : (Filter, "b) Italy	
(2/1) 28/	
roid Mallor (int n)	
4	
int *ptr;	
intific a stismele in redomin but extre	1 19
pte = (int *) mallor (n * size of (int));	
for (i=0; i <n; en="" etranele="" i++)="" it<="" td=""><td></td></n;>	
pta[i] = i+1; 2 + 5 (0)	
printf ("mallor memory allocation")	
printf ("In the elements are: In")3	
for(i=0;i <n;i++)< td=""><td></td></n;i++)<>	
printf ("/d", ptr(il);	
Proe (ptr);	
3	
void Callor (int n)	
\{	
int * pta;	
intij	
ptr = (int *) callor (n sincel (int));	

for (i=0; icn; i++) ptr[i]= 1+1; printf ("rallor memory allocation: \\n")

printf ("the elements are: \\n");

for (i=0; i<n; i++)

printf ("%d", ptr [i]); etr = (int *) reallor (pte, n * size of (int)); Apa (i=5; i<n; i++) ptr[i]=i+1; prints ("In realer memory allocation");

prints ("In the elements are: In");

for (i = 0; i < n; i++)

prints ("'/d", prt[i]); free (etr) output: enter the number of elements n: mallor memory allastion the elements are:

3> 5-	tack Implementation	Land to the state of the state
	THE THE PARTY OF T	residence a succession de la secución de acua participa de la contractiva del la contr
#	include (stdio,h)	1114
11	include < stdlib.h>	
11	define SIZE 4	4
	top = -1;	
	la (SIZE) is It will be	Level 1) Maria
- 11	d puch ();	
11	d pop ();	· ·
	L show ();	
poie	d main()	
ર્વ		(I gra high)
	int ch;	
	while (1)	(1
	Ilou to "Die of wall	eahou") Atriag
	printf (operations	on the stack: In');
		it 2. Pap the element 3. Shaw 4. End
1	paintf "Enter the choi	reighn");
٤(
	paintf "Enter the chair	
>	scorf ("/d", & ch)	
(scorf ("/d", & ch)	i (- got = got
(sconf ("/d", &ch); ewitch (ch)	
)	peintf ("Enter the chair scanf ("/d", &ch); ewitch (ch) { case 1: push () break;	i (- got = got i
	peintf "Enter the chair scanf ("/d", & ch); switch (ch) { case 1: push () break; case 2: pop();	$(1 - \frac{1}{2}) + \frac{1}{2}$
	peintf "Enter the chair scanf ("/d", & ch); switch (ch) { case 1: push () break; case 2: pop();	$(1 - x_0 t) + x_0 t$ $(1 - x_0 t) + x_0 t$ $(1 - x_0 t) + x_0 t$
	peintf ("Enter the chair scanf ("/d", & ch); ewitch (ch) { case 1: push () break; case 2: pop();	$(1 - x_0 t) + x_0 t$ $(1 - x_0 t) + x_0 t$ $(1 - x_0 t) + x_0 t$
	peintf ("Enter the chair scanf ("/d", & ch); switch (ch) { case 1: push() break; case 2: pop(); break; case 3: show(); break;	$(1 - x_0 t) + x_0 t$ $(1 - x_0 t) + x_0 t$ $(1 - x_0 t) + x_0 t$
	peintf ("Enter the chair scarf ("/d", & ch); switch (ch) { case 1: push() break; case 2: pop(); break; case 3: show(); break; case 4: excit(0)	() contact in a single of the
	peintf ("Enter the chair scarf ("/d", & ch); switch (ch) { case 1: push() break; case 2: pop(); break; case 3: show(); break; case 4: excit(0)	Tivalid choice ")

```
roid push ()
      if (top == SIZE -1)

paintf ("overflow in");
      else
           scarf (" Later the element to be added!
           top = top +1;
           a[top] = 2;
roid pop ()
     if (top = = - 1)
         printf ("Underflow In");
                pentf ("/d \n", a[i])
```



Output:
Operations on the stack:
1. Push the element 2. Pop the element 3. show. 4. End
Enter the choice: 1
Enter the element to be added: 3
operations on the stark:
1. Push the element 2. Pop the element 3. Show 4. End
Enter the choice: 1
Enter the element to be added: 2
Operation on the stack:
1 Push the element 2 Pop the element 3. Show 4. End
Erter the choice: 2
Popped clement: 2
Operations on the stack:
1 Push the element 2 Prop the clament 3 show 4. End
Enter the choice: 3
Elements is the stack are: 3
Operations on the stack?
1 rush the element 2 Pap the element 3 show 4. Frd
Enter the choice: 4
1 12
지어 보면 사람들이 어느 아니는 그는 그는 그는 그는 그를 가면 하고 있다면 하는데 그는 것이다.