I Write a program to simulate the rearking of stack using an array with the following: a) Push b) Pop Display. The program should print appropriate messages for stack overflow, stack underflow # include <stdio.h> #include < conio. h> #include <stalib.h> # define max 3 int st[max], top =-1; void push (int st[], int val); int pop (int St[]); int peck (int St[]); roid display (int St[]); int main (int auges char *augv[]) int val, oftion; perint (" \n **** MATN MENU *** "); printf ("In 1. PUSH"); printf("In 2. POP"); freintf ("In 3. PEEK"); printf (" In 4. DISPLAY"); pountf ("In S. EXIT");

printf ("In Enter your ofition:"); scarf ("%d", &option); sweetch (option) printf("In Enter the number to be pushed on stack: "); scant ("'o/od"; & val); frush (st, ral); val=pop(st); if (val)=-1)

printf ("In The value deleted

from stack is: %d", val);

break; if (val = -1) val = freek (st); if (vall =-1) printf ("In The value stored at top of stack is: "hol", vol); display (st); break; while (option! = 5); road push (int SEI) int wal)

if (top == max-1) printf ("In STACK OVERFLOW"). int pop (int St[]) perintf ("In STACK UNDERFLOW"). val = st[top]; void display (int StrJ) if (top = = -1) printf(")n STACK IS EMPTY"); for (i= top; i>=0 31-) pointf("In %d", st[i];

} frint("\n"); printf ("In STACK IS EMPTY");

oreturn -1; return (St[top]); Output: **** MAIN MENU**** 1. PUSH 2. POP 3. PEEK 4. DISPLAY S.EXIT Enter your option: 2 STACK UNDERFLOW * *** MAIN MENU**** 1. PUSH 4. DISPLAY S.EXIT Enter your option: 3

STACK IS EMPTY *** ** MAIN MENUX** 1. PUSH 2. POP 3. PEEK 4, DISPLAY 5. EXIT Enter your option: 1 Enter the number to be justed onstack:4 *****MAIN MENU**** 1. PUSH 2. POP 3. PEFK 4. DISPLAY 5. EXIT Enter your option:1 Enter the number to be pushedon stack:6 ********* 1 PUSH 2 POP 3. PFFK 4. DISPLAY 5. EXIT Enter your option: Enter the number to be pushed on stack: 8

1	***** MAIN MENU****	-
	***** MAIN MENU****	1
	1 PUSH	/
	2. POP	/
	3. PEEK	-
	4. DISPLAY	-
	5. EXIT	-
	Enter your option:1	~
		-
	Enter the number to be pushed on stacky	1
		1
	STACK OVERFLOW	1
	****** MAIN MENU****	
	1.PUSH	
	2.POP	
	3. PEFK	
	4. DTSPLAY	
	5. EXTT	
	Enter your ofition: 3	
	The value stored at top of stack is:8	-
	, 0	
110	***** MATN MENU****	
	1. PUSH	-
	2.POP	
	3.PEFK	
	4. DTSPLAY	
	SEXIT	
	Enter your ofition: 4	
	8	
	6	
	1	
		Victoria Service
		-

1	****************
-	1. PUSH
-	2.POP
	3. PEEK
-	4. DISPLAY
	s. FXIT
	Enter your option:5
	The state of the s
	LANGE BUILDING
	in a det test
	Cha mada de trais Etra de teles
	I Thomas and a late to the town the
	The state of the s
	I The state of the
	Cox note (12 april de la
	The population of the second state of the seco
	Donor Late
	I had the training the state of
	the teles were adulated transferred
	The state of the s
/	
-	

Perograms on Dynamic Memory allocation 1. Perogram to calculate the sum of n numbers entered by the user using malloc and free the sum of n # include <stdio.h> # include <stdlib.h> ent main() int n, i, *ptr, sum=0; printf ("Enter number of elements:"). scarf ("%d", &n). ptr = (int *) mallor (n * size of (int);

// if memory cannot be allocated

if (ptr == NULL) of (ptr==NULL) printf ("Everor! memory not allocated.") founts ("Enter elements:").

for (i=0; i<n;++i) Scanf("%d" ptr+i); Sum+= * (fiter+i); freint ("Sum = %d" sum); //deallocating the memory free (ptr); return 0;

Output: Enter elements: 12 34 56 2. Perogram to calculate the sum of n numbers externed by the user using calloc and free. #include <stdio.h> #include <stdlib.h> int main () int nois *ptr, sum=0; prints ("Enter number of elements:"). scanf ("%d", &n); ptr = (int*) calloc (n, size of (int)); if (ptr == NULL) printf ("Error! memory not allocated."). for (i = 0; i<n; ++i) scanf("0/od"; pten+i);

sumf + = *(ptr+i); free (ptr). Joreturn O:

Cutput: Enter number of elements: 4 Enter elements: 24 23 28 20 Sum = 95 3. Perogram to demonstrate dynamic memory allocation using realloc # include < stdio.h> #include <adlib.h.> int main() int *ptr, is n1, n2; printf("Enter size:"); scanf("%d; 4n1); pti(int*) malloc (n1* sizeof (int)).

printf ("Addresses of foreviously
allocated memory: \n"); for (i=0; i<ni; ++i)

printf ("%pc\n"; ptr+i);

printf ("\nEnter the now size."); sconf ("lod", 4n2); Heredlocating the memory

ptr = realloc (ptr > n2 * size of (int);

printf ("Addresses of newly allocated memory: 1n?); for (i=0; 1<n2; ++i) printf (" lopan " ptr+i). free (ptr). returno;

Enter size: 2 Enter size. 2 doldnesses of premionsly allocated memory: 000000000091430C 00000000009914340 Enter the new size.

Soldresses of newly allocated memory: Enter the new size: 3 00000000009914300 00000000000914340 000000000009914380