Q. Implement push, pop and find the minimum element in a stack in O(1) time complexity.

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

int stack[100], support_stack[100];

int push(int element, int *top, int *stack)
{
    *top = *top + 1;
    stack[*top] = element;
}
```

National Day (China, Hong Kong, Nigeria), Muharram (India)

1
Sunday
274-91

```
int pop(int *stack, int *top)
{
    int element;
    if (*top > -1)
    {
        element = stack[*top];
        *top = *top - 1;
        return element;
    }
```

October 2017

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

```
    else
    {
        printf("\n STACK EMPTY\n");
        return -99999; // means nothing is popped.
    }
}

int main()
{
    int choice, element, top_main = -1;
    int top_support = -1, i, supp_pop_element;
    int pop_element;
    printf(" Enter the operation:\n");
    printf("\n1. Push\n 2. Pop\n 3. Check min\n");
    printf("4. Stop\n");
    scanf("%d", &choice);
    while (choice != 5)
    {
        if (choice == 1)
        {
            printf("\nEnter num");
            scanf("%d", &element);
```

```
else if (choice == 3)
{
  if(top_support > -1)
   printf("\n Min elent = %d\n", support_stack[top_sup]);
  else
   printf("\n STACK EMPTY");
}

else if (choice == 4)
{
  if(top_main > -1)
  {
     printf ("\n MAIN STACK\n");
     for(i=top_main; i>=0; i--)
     {
       printf ("\n%d", stack[i]);
     }
  }
else
    printf ("\n STACK EMPTY\n");
    printf("\n top_support = %d", top_support);
    for(i = top_support; i>=0; i--)
       {
          printf("\n%d", support_stack[i]);
       }
}
```

```
8  printf("\n Enter the operation : \n 1. Push\n");
   printf("\n 2. Pop \n 3. check minimum\n 4. See full stack\n");
9  printf("\n 5. STOP \n");

10 scanf(" %d", &choice);
}
11 return 0;
}
12
```