**VARUN KUMAR**

**2K19-IT-140**

**DS LAB - 4**

7: Write a program to find the minimum element of the stack in constant time with using extra space.

**#include <stdio.h>**

**int a[100],b[100],top=-1,n;**

**void push()**

**{**

**int x;**

**if(top==n-1)**

**printf("Stack is FULL\n");**

**else if(top==-1)**

**{**

**top++;**

**printf("Enter the element to be pushed: ");**

**scanf("%d",&x);**

**a[top]=b[top]=x;**

**}**

**else**

**{**

**top++;**

**printf("Enter the element to be pushed: ");**

**scanf("%d",&x);**

**a[top]=x;**

**if(x>=b[top-1])**

**b[top]=b[top-1];**

**else**

**b[top]=x;**

**}**

**}**

**void pop()**

**{**

**if(top==-1)**

**printf("Top is EMPTY\n");**

**else**

**{**

**printf("Element to be popped is: %d\n",a[top]);**

**top--;**

**}**

**}**

**void peek()**

**{**

**if(top==-1)**

**printf("Top is EMPTY\n");**

**else**

**printf("Element at the top is: %d\n",a[top]);**

**}**

**void display()**

**{**

**if(top==-1)**

**printf("Top is EMPTY\n");**

**else**

**{ int i;**

**printf("Elements in the stack are\n");**

**for(i=top;i!=-1;i--)**

**{**

**printf("%d ",a[i]);**

**}**

**printf("\n");**

**}**

**}**

**void min()**

**{**

**if(top==-1)**

**printf("Top is EMPTY\n");**

**else**

**{**

**printf("The MINIMUM element is: %d\n",b[top]);**

**}**

**}**

**int main() {**

**int choice;**

**printf("Enter the size of the stack: ");**

**scanf("%d",&n);**

**printf("PRESS\n");**

**printf("1. PUSH\n2. POP\n3. PEEK\n4. DISPLAY\n5. MINIMUM ELEMENT\n6. EXIT\n");**

**do**

**{**

**printf("Enter your choice: ");**

**scanf("%d",&choice);**

**switch(choice)**

**{**

**case 1:**

**{**

**push();**

**break;**

**}**

**case 2:**

**{**

**pop();**

**break;**

**}**

**case 3:**

**{**

**peek();**

**break;**

**}**

**case 4:**

**{**

**display();**

**break;**

**}**

**case 5:**

**{**

**min();**

**break;**

**}**

**case 6:**

**{**

**printf("END\n");**

**break;**

**}**

**default:**

**{**

**printf("Enter correct choice\n");**

**}**

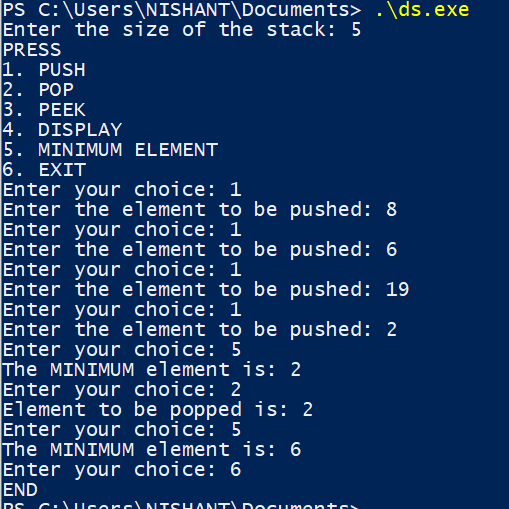
**}**

**}**

**while(choice!=6);**

**return 0;**

**}**



# 8: Write a program to find the minimum element of the stack in constant time without using extra space

**#include <stdio.h>**

**int a[100],minEle,top=-1,n;**

**void push(int x)**

**{**

**// Insert new number into the stack**

**if (top==-1)**

**{**

**minEle = x;**

**top++;**

**a[top]=x;**

**printf ("Number Inserted: %d\n",x);**

**return;**

**}**

**// If new number is less than minEle**

**if (x < minEle)**

**{**

**top++;**

**a[top]=2\*x - minEle;**

**minEle = x;**

**}**

**else{**

**top++;**

**a[top]=x;**

**}**

**printf( "Number Inserted: %d\n",x);**

**}**

**void pop()**

**{**

**if (top==-1)**

**{**

**printf ("Stackintf is empty\n");**

**return;**

**}**

**printf( "Top Most Element Removed: ");**

**int t = a[top];**

**top--;**

**// Minimum will change as the minimum element**

**// of the stack is being removed.**

**if (t < minEle)**

**{**

**printf ("%d\n",minEle);**

**minEle = 2\*minEle - t;**

**}**

**else**

**printf ("%d\n",t);**

**}**

**void peek()**

**{**

**if (top==-1)**

**{**

**printf ("Stack is empty ");**

**return;**

**}**

**int t = a[top]; // Top element.**

**printf ("Top Most Element is: ");**

**// If t < minEle means minEle stores**

**// value of t.**

**(t < minEle)? printf("%d",minEle ): printf("%d",t);;**

**}**

**void display()**

**{**

**if(top==-1)**

**printf("Top is EMPTY\n");**

**else**

**{ int i;**

**printf("Elements in the stack are\n");**

**for(i=top;i!=-1;i--)**

**{**

**printf("%d ",a[i]);**

**}**

**printf("\n");**

**}**

**}**

**void min()**

**{**

**if (top==-1)**

**printf ("Stack is empty\n");**

**// variable minEle stores the minimum element**

**// in the stack.**

**else**

**printf("Minimum Element in the stack is: %d\n " ,minEle);**

**}**

**int main() {**

**int choice;**

**printf("Enter the size of the stack: ");**

**scanf("%d",&n);**

**printf("PRESS\n");**

**printf("1. PUSH\n2. POP\n3. PEEK\n4. DISPLAY\n5. MINIMUM ELEMENT\n6. EXIT\n");**

**do**

**{**

**printf("Enter your choice: ");**

**scanf("%d",&choice);**

**switch(choice)**

**{**

**case 1:**

**{ int x; scanf("%d",&x);**

**push(x);**

**break;**

**}**

**case 2:**

**{**

**pop();**

**break;**

**}**

**case 3:**

**{**

**peek();**

**break;**

**}**

**case 4:**

**{**

**display();**

**break;**

**}**

**case 5:**

**{**

**min();**

**break;**

**}**

**case 6:**

**{**

**printf("END\n");**

**break;**

**}**

**default:**

**{**

**printf("Enter correct choice\n");**

**}**

**}**

**}**

**while(choice!=6);**

**return 0;**

**}**

