**DS Algorithm Essentials | December 2020**

**Assignment Day 6 | 31th December 2020**

**Name : S.Rohit**

**Code :**

**#include<stdio.h>**

**#include<stdlib.h>**

**int data[100000],top,max=0;**

**void push()**

**{**

**int item;**

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

**top++;**

**data[top]=item;**

**if(max<data[top])**

**max=data[top];**

**}**

**void pop()**

**{**

**int i;**

**if(max==data[top])**

**max=0;**

**top--;**

**for(i=top;i>=0;i--)**

**if(max<data[i])**

**max=data[i];**

**}**

**int main()**

**{**

**int t,n,choice;**

**top=-1;**

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

**while(t--)**

**{**

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

**switch(choice)**

**{**

**case 1 : push();**

**break;**

**case 2: pop();**

**break;**

**case 3: printf("%d\n",max);**

**break;**

**}**

**}**

**return 0;**

**}**

**typedef struct {**

**int str[8000];**

**int top;**

**int min[20];**

**int mincnt;**

**} MinStack;**

**MinStack\* minStackCreate() {**

**MinStack \*Min;**

**Min=(MinStack\*)malloc(sizeof(MinStack));**

**Min->top=-1;**

**Min->mincnt=0;**

**return Min;**

**}**

**void minStackPush(MinStack\* obj, int x) {**

**obj->top++;**

**obj->str[obj->top]=x;**

**printf("mincnt=%d push:%d\n",obj->mincnt,x);**

**if(obj->mincnt==0||x<=obj->min[obj->mincnt-1])**

**{**

**obj->min[obj->mincnt++]=x;**

**printf("%d\*",x);**

**}**

**}**

**void minStackPop(MinStack\* obj) {**

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

**return ;**

**if(obj->mincnt==0)**

**;**

**else if(obj->str[obj->top]==obj->min[obj->mincnt-1])**

**obj->mincnt--;**

**obj->top--;**

**}**

**int minStackTop(MinStack\* obj) {**

**return obj->str[obj->top];**

**}**

**int minStackGetMin(MinStack\* obj) {**

**return obj->min[obj->mincnt-1];**

**}**

**void minStackFree(MinStack\* obj) {**

**free(obj);**

**}**