Stack using Array

1. A stack, S1, contains some numbers in arbitrary order. Using another stack, S2, for temporary storage, design an algorithm to sort the numbers in S1 such that the smallest is at the top of S1 and the largest is at the bottom.
2. Implement a stack S of n elements using arrays. Write functions to perform PUSH and POP operations. Implement queries using the push and pop functions too
3. Retrieve the mth element of the stack S from the top (m<n) , leaving the stack without its top m-1 elements.
4. Retain only the elements in the odd position of the stack and pop out all even positioned elements.

#include<stdio.h>

int stack[100],choice,n,top,x,i;

void push(void);

void pop(void);

void display(void);

int main()

{

    //clrscr();

    top=-1;

    printf("\n Enter the size of STACK[MAX=100]:");

    scanf("%d",&n);

    printf("\n\t STACK OPERATIONS USING ARRAY");

    printf("\n\t--------------------------------");

    printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t 4.EXIT");

    do

    {

        printf("\n Enter the Choice:");

        scanf("%d",&choice);

        switch(choice)

        {

            case 1:

            {

                push();

                break;

            }

            case 2:

            {

                pop();

                break;

            }

            case 3:

            {

                display();

                break;

            }

            case 4:

            {

                printf("\n\t EXIT POINT ");

                break;

            }

            default:

            {

                printf ("\n\t Please Enter a Valid Choice(1/2/3/4)");

            }

        }

    }

    while(choice!=4);

    return 0;

}

void push()

{

    if(top>=n-1)

    {

        printf("\n\tSTACK is over flow");

    }

    else

    {

        printf(" Enter a value to be pushed:");

        scanf("%d",&x);

        top++;

        stack[top]=x;

    }

}

void pop()

{

    if(top<=-1)

    {

        printf("\n\t Stack is under flow");

    }

    else

    {

        printf("\n\t The popped elements is %d",stack[top]);

        top--;

    }

}

void display()

{

    if(top>=0)

    {

        printf("\n The elements in STACK \n");

        for(i=top; i>=0; i--)

            printf("\n%d",stack[i]);

        printf("\n Press Next Choice");

    }

    else

    {

        printf("\n The STACK is empty");

    }

}

**#include<stdio.h>**

**void put\_last(float a[],int n,float b) {a[n+1]=b;}**

**void remove\_last(float a[],int n){}**

**float last(float a[],int n){return a[n];}**

**void print(float a[],int n){int i;for (i=1;i<=n;i++) printf("%.1f,",a[i]);}**

**main()**

**{ float x[50],element; int size=0,i,choice;**

**do{printf("\nGive choice: 0:put 1:remove 2:top 3:print 4:exit");**

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

**if (choice==0){**

**printf("Give element to be put ");scanf("%f",&element);**

**put\_last(x,size,element);size++;}**

**if (choice==1){remove\_last(x,size);size--;}**

**if (choice==2){printf("The element is %.1f\n",last(x,size));}**

**if (choice==3){print(x,size);}**

**}while(choice!=4);**

**}**