Name-Ayan Sadhukhan

Registration No- 19BEC1448

*Question 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.

*Code:*

#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\n\t 5.Sort the Stack");

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;

}

case 5:

{ int tmpStack[100];

int input;

tmpStack = sortStack(input);

printf("Sorted Numbers are: ")

while (!tmpStack.empty())

{

printf("%d ",tmpStack.top());

tmpStack.pop();

}

}

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");

}

}

int sortStack(&input)

{

//stack<int> tmpStack;

int tmpStack[100];

while (!input.empty())

{

// pop out the first element

int tmp = input.top();

input.pop();

// while temporary stack is not empty and top

// of stack is greater than temp

while (!tmpStack.empty() && tmpStack.top() > tmp)

{

// pop from temporary stack and push

// it to the input stack

input.push(tmpStack.top());

tmpStack.pop();

}

// push temp in tempory of stack

tmpStack.push(tmp);

}

return tmpStack;

}

*Question 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

1. Retrieve the mth element of the stack S from the top (m<n) , leaving the stack without its top m-1 elements.

*Code:*

#include<stdio.h>

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

void push(void);

void pop(void);

void display(void);

void rem(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 \n\t 5. remove element till m th position");

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;

}

case 5:

{

rem();

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");

}

}

void rem()

{

printf("Enter the mth element from the top ");

scanf("%d",&m);

if(m>n)

{

printf("Invalid Input");

}

else

{

for(int i=0;i<m;i++)

{

pop();

}

printf("\n The %d th element in the stack is ",m);

printf("\n %d",stack[top]);

}

}

1. Retain only the elements in the odd position of the stack and pop out all even positioned elements.

*Code:*

#include<stdio.h>

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

void push(void);

void pop(void);

void display(void);

void rem(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 \n\t 5. remove the even positioned elements");

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;

}

case 5:

{

rem();

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");

}

}

void poppy(int position)

{

if (position >= n+1)

printf("Deletion not possible.\n");

else

{

for (int c = position - 1; c < n - 1; c++)

stack[c] = stack[c+1];

}

}

void rem()

{

if(top<=-1)

{

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

}

else

{

int t=top;

int f=0;

for (int i=0;i<t;i++)

{

if (i%2==0)

{

poppy(i);

f=f+1;

}

else

{

top=top-1;

}

}

//top=t-f;

}

}