|  |
| --- |
| /\*Description: Array implementation of a stack |
|  | \* Learner: ARSHEE QURESHI |
|  | \* Created on: 13th JULY 2017 |
|  | \*/ |
|  | #include<stdio.h> |
|  | #include<stdlib.h> |
|  | #define MAXSIZE 10 |
|  |  |
|  | int stack[MAXSIZE], top=-1; |
|  | int isFull(){ |
|  | //return 1 if stack is full else returns -1 |
|  | return (top==MAXSIZE-1); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | int isEmpty(){ |
|  | //return 1 if stack is empty else returns-1 |
|  | return (top==-1); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | int peek(){ |
|  | //return elements at the top of stack |
|  | return stack[top]; |
|  | } |
|  |  |
|  | void push(int e){ |
|  | //inserts an element into stack |
|  | if(!isFull()) |
|  | { |
|  | stack[++top]=e;//increment the top and set top of stack equal to e |
|  | printf("\nYour element %d successfully inserted\n",e);//message as element successfully inserted |
|  | } |
|  | else |
|  | { |
|  | //message stack overflow |
|  | printf("\nSorry. Your element cannot be inserted. Stack overflow\n"); |
|  | } |
|  |  |
|  | } |
|  |  |
|  | void pop() |
|  | { |
|  | //deletes an element from the top of stack |
|  | int d; |
|  | if (!isEmpty()) |
|  | { |
|  | d=stack[top--];//set top of stack equal to d and decrement the top |
|  | printf("\nYour elements %d sucessfully deleted\n",d);//message about successful deletion of element from stack |
|  | } |
|  | else |
|  | { |
|  | //message stack uderflow |
|  | printf("\nSorry. Your element cannot be deleted. stack underflow\n"); |
|  | } |
|  | } |
|  | int main() |
|  | { |
|  |  |
|  | int ch,e; |
|  | do |
|  | { |
|  | //display menu 1.peek 2.push 3.pop 4.exit |
|  | printf("\nDisplay Menu \n 1.Peek\n 2.Push\n 3.Pop\n 4.Exit\n"); |
|  | printf("\nEnter your choice:"); |
|  | scanf("%d",&ch); //take input in choice variable |
|  | switch(ch){ |
|  |  |
|  |  |
|  | case 1: |
|  | e=peek();//call peek function |
|  | printf("\n Element at the top of the stack is :- %d\n",e); |
|  | break; |
|  | case 2: |
|  | printf("\nEnter the element to be inserted :- \n"); |
|  | scanf("%d",&e); |
|  | push(e);//call push function |
|  | break; |
|  | case 3:pop();//call pop function |
|  | break; |
|  |  |
|  | case 4: |
|  | exit(0);//call exit (0) function or return 0 |
|  | break; |
|  | default: |
|  | printf("\nInvalid choice\n");//choice is invalid |
|  | break; |
|  | } |
|  | }while (1); |
|  | return 0; |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  |  |
|  | /\* |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Push |
|  | 3.Pop |
|  | 4.Exit |
|  |  |
|  | Enter your choice:2 |
|  |  |
|  | Enter the element to be inserted :- |
|  | 10 |
|  |  |
|  | Your element 10 successfully inserted |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Push |
|  | 3.Pop |
|  | 4.Exit |
|  |  |
|  | Enter your choice:2 |
|  |  |
|  | Enter the element to be inserted :- |
|  | 20 |
|  |  |
|  | Your element 20 successfully inserted |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Push |
|  | 3.Pop |
|  | 4.Exit |
|  |  |
|  | Enter your choice:2 |
|  |  |
|  | Enter the element to be inserted :- |
|  | 30 |
|  |  |
|  | Your element 30 successfully inserted |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Push |
|  | 3.Pop |
|  | 4.Exit |
|  |  |
|  | Enter your choice:1 |
|  |  |
|  | Element at the top of the stack is :- 30 |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Push |
|  | 3.Pop |
|  | 4.Exit |
|  |  |
|  | Enter your choice:3 |
|  |  |
|  | Your elements 30 sucessfully deleted |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Push |
|  | 3.Pop |
|  | 4.Exit |
|  |  |
|  | Enter your choice:5 |
|  |  |
|  | Invalid choice |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Push |
|  | 3.Pop |
|  | 4.Exit |
|  |  |
|  | Enter your choice:4 |
|  |  |
|  |  |
|  | ------------------ |
|  | (program exited with code: 0) |
|  | Press return to continue |
|  |  |
|  | \*/ |