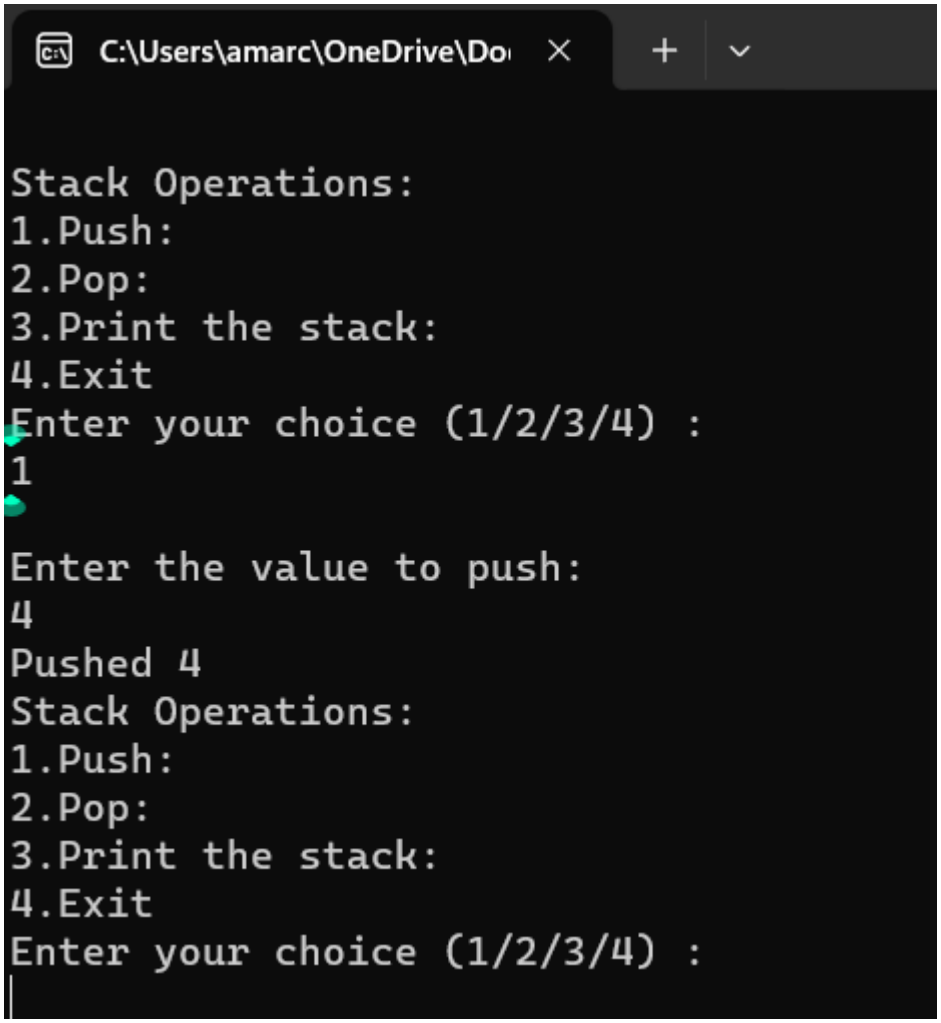
```
#include<stdio.h>
#include<stdlib.h>
#define MAX_SIZE 100
typedef struct{
	int items[MAX_SIZE];
	int top;
}Stack;
void intistack(Stack *stack)
{
	stack->top=-1;
}
int isempty(Stack *stack)
{
	return stack->top==-1;
}
int isfull(Stack *stack)
{
	return stack->top==MAX_SIZE-1;
}
void push(Stack *stack, int val)
{
	if(isfull(stack))
	{
		printf("The stack is full, can't push %d",val);
		return;
	}
	stack->top++;
	stack->items[stack->top]=val;
	printf("Pushed %d",val);
}
int pop(Stack *stack)
```

```
{
	if(isempty(stack))
	{
		printf("The stack is empty, can't pop an empty stack\n");
		return -1;
	}
	int popval=stack->items[stack->top];
	stack->top--;
	printf("Popped %d",popval);
	return popval;
}
void print(Stack *stack)
{
	if(isempty(stack))
	{
		printf("The stack is empty.\n");
		return;
	}
	printf("Current Stack:\n");
	for(int i=stack->top;i>=0;i--)
	{
		printf("%d ",stack->items[i]);
	}
	printf("\n");
}
int findmin(Stack *stack)
{
	if(isempty(stack))
	{
		printf("The Stack is empty.\n");
		return INT_MAX;
```

```
        }
        int min=stack->items[0];
        for(int i=1;i<=stack->top;i++)
        {
                if(stack->items[i]<min)
                {
                        min=stack->items[i];
                }
        }
        return min;
}
int main()
{
        Stack stack;
        intistack(&stack);
        int choic,value;
        do{
                printf("\nStack Operations:\n");
                printf("1.Push:\n");
                printf("2.Pop:\n");
                printf("3.Print the stack:\n");
                printf("4.Exit\n");
                printf("Enter your choice (1/2/3/4) : \n");
                scanf("%d",&choic);
                switch(choic)
                {
                        case 1:
                                printf("\nEnter the value to push: \n");
                                scanf("%d",&value);
                                push(&stack,value);
                                break;
```

```
                case 2:
                        pop(&stack);
                        break;
                case 3:
                        print(&stack);
                        break;
                case 4:
                        printf("Exiting...\n");
                        break;
                default:
                        printf("Invalid choice, please enter again!\n");
        }
}while(choic!=4);
printf("The Final stack: \n");
print(&stack);
printf("The Top pointer is at %d",stack.top);
int Min=findmin(&stack);
if(Min!=INT_MAX)
{
        printf("\nThe Minimum element in the stack: %d\n",Min);
}
else{
        printf("Final Stack is empty.\n");
}


}
```


C:\Users\amarc\OneDrive\Do
Stack Operations:
1.Push:
2.Pop:
3.Print the stack:
4.Exit
Enter your choice (1/2/3/4) :
1
Enter the value to push:
4
Pushed 4
Stack Operations:
1.Push:
2.Pop:
3.Print the stack:
4.Exit
Enter your choice (1/2/3/4) :