

LAB – 4 – PROGRAMMING TOOLS IN LINUX-I

Exercise :

1) In C, write a program to implement a stack with push, pop operations using suitable functions. Create static libraries for various operations on stack. Create a header file for function declaration.

Answer :

stack.h

```
#include<stdbool.h>
#define MAX 6
typedef struct {
char data[MAX];
int top;
} stack;
bool isFull(stack* s);
void push(stack *s, char c);
bool isEmpty(stack* s);
char pop(stack *s);
void display(stack* s);
```

push.c

```
#include<stdio.h>
#include<stdbool.h>
#include "stack.h"
bool isFull(stack* s){
if(s->top == MAX-1){
return true;
}
else
return false;
}
void push(stack *s,char c)
{
if(isFull(s)){
printf("Stack is Full\n");
return;
}
s->top++;
s->data[s->top] = c;
}
```

pop.c

```
#include<stdio.h>
#include<stdbool.h>
#include "stack.h"
bool isEmpty(stack* s){
    if(s->top == -1){
        return true;
    }
    else
        return false;
}
char pop(stack *s)
{
    if(!isEmpty(s)){
        return(s->data[s->top--]);
    }
}
```

display.c

```
#include<stdio.h>
#include "stack.h"
void display(stack* s){
    if(isEmpty(s)){
        printf("Stack is empty\n");
        return;
    }
    int count = s->top;
    while(count>-1){
        printf("%c\n",s->data[count--]);
    }
}
```

program.c

```
#include<stdio.h>
#include<stdbool.h>
#include "stack.h"
void main(){
    stack st;
    stack* s = &st;
    s->top = -1;
    push(s,'r');
    push(s,'e');
    push(s,'m');
    push(s,'r');
    push(s,'a');
```

```
push(s,'f');  
push(s,'m');
```

```
printf("\n");  
display(s);  
pop(s);  
pop(s);  
printf("\n");  
display(s);  
pop(s);  
pop(s);  
pop(s);  
pop(s);  
printf("\n");  
display(s);  
}
```

Terminal commands :

```
gcc -c push.c pop.c display.c  
ar crv stack.a push.o pop.o display.o  
// ls *.o - to check the creation of object files  
ranlib stack.a  
gcc -c program.c  
gcc -o program program.o stack.a  
./program
```

```
student@dslab:~/Desktop/200905130/lab4$ ls  
display.c pop.c program.c push.c stack.h  
student@dslab:~/Desktop/200905130/lab4$ gcc -c push.c pop.c display.c  
student@dslab:~/Desktop/200905130/lab4$ ls *.o  
display.o pop.o push.o  
student@dslab:~/Desktop/200905130/lab4$ ar crv stack.a push.o pop.o display.o  
a - push.o  
a - pop.o  
a - display.o  
student@dslab:~/Desktop/200905130/lab4$ ranlib stack.a  
student@dslab:~/Desktop/200905130/lab4$ gcc -c program.c  
student@dslab:~/Desktop/200905130/lab4$ ls  
display.c display.o pop.c pop.o program.c program.o push.c push.o stack.a stack.h
```

Output :

```
student@dslab:~/Desktop/200905130/lab4$ gcc -c program.c
student@dslab:~/Desktop/200905130/lab4$ gcc -o program program.c stack.a
student@dslab:~/Desktop/200905130/lab4$ ./program
Stack is Full

f
a
r
m
e
r
r
m
e
r

Stack is empty
```
