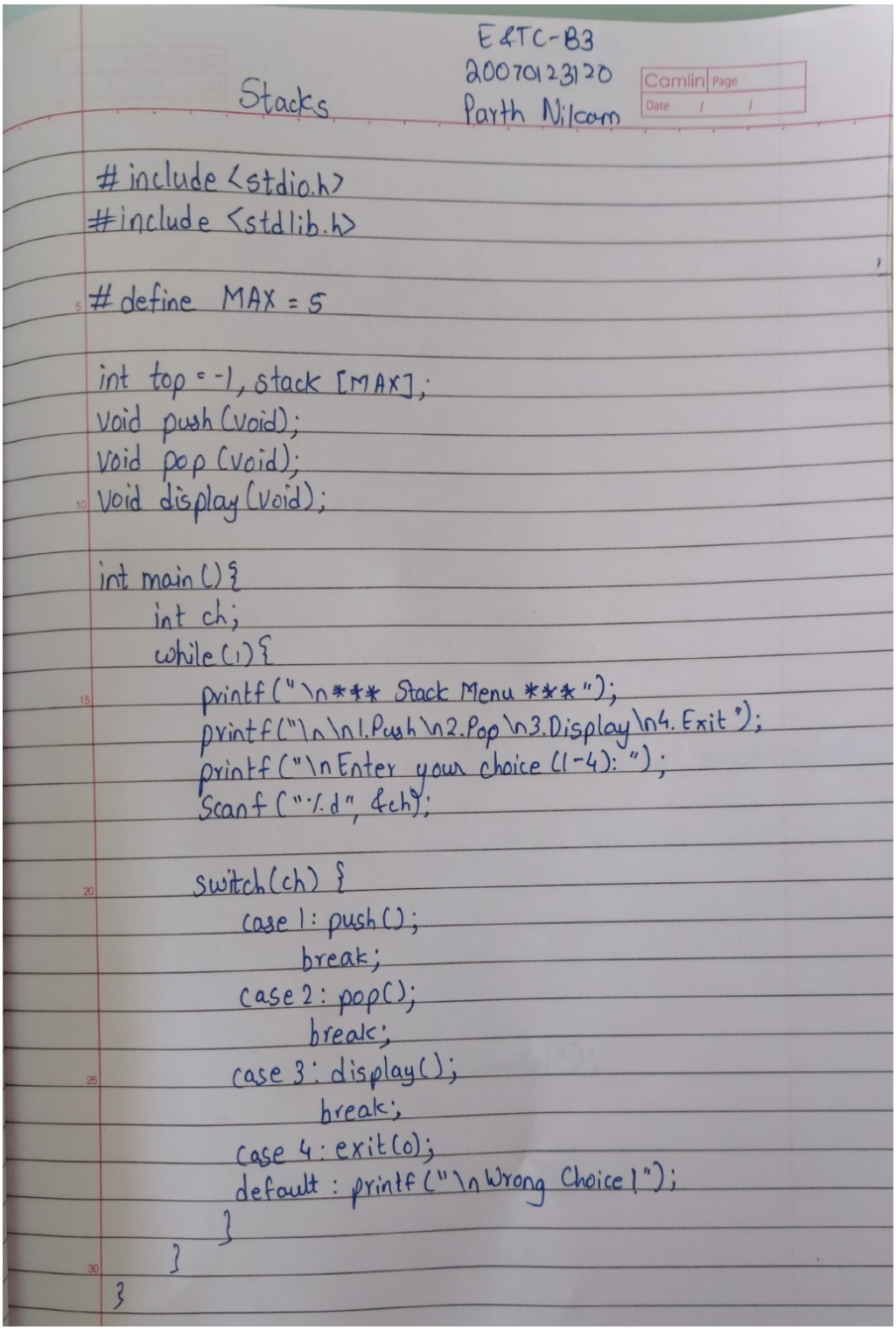
Data Structures using C

Parth Nikam
20070123120
E&TC – B3

Aim: - Studying and coding on stack.

Objective: - To perform push, pop, and peek operations on stack in C language

Code: -



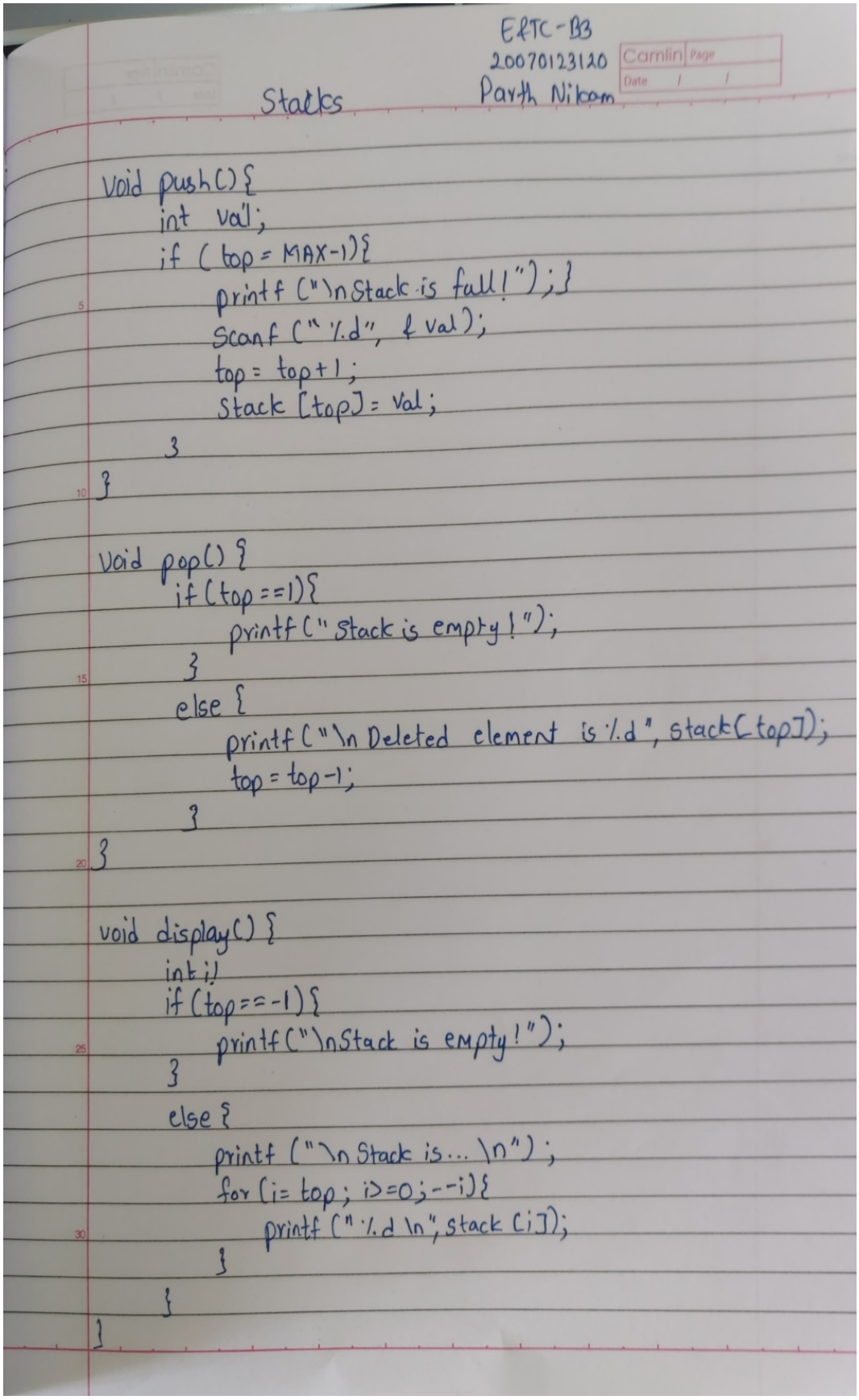
E&TC-B3
20070123120
Parth Nilcam

Stacks

```
#include <stdio.h>
#include <stdlib.h>

#define MAX = 5

int top = -1, stack [MAX];
void push (void);
void pop (void);
void display (void);

int main (){
    int ch;
    while (1){
        printf ("\n*** Stack Menu ***");
        printf ("\n\n1.Push \n2.Pop \n3.Display \n4. Exit");
        printf ("\nEnter your choice (1-4): ");
        Scanf ("%d", &ch);

        switch (ch) {
            case 1: push ();
                break;
            case 2: pop ();
                break;
            case 3: display ();
                break;
            case 4: exit (0);
            default : printf ("\n Wrong Choice !");
            }
    }
}
```

Data Structures using C

Parth Nikam
20070123120
E&TC – B3



E&TC - B3
20070123120
Parth Nikam

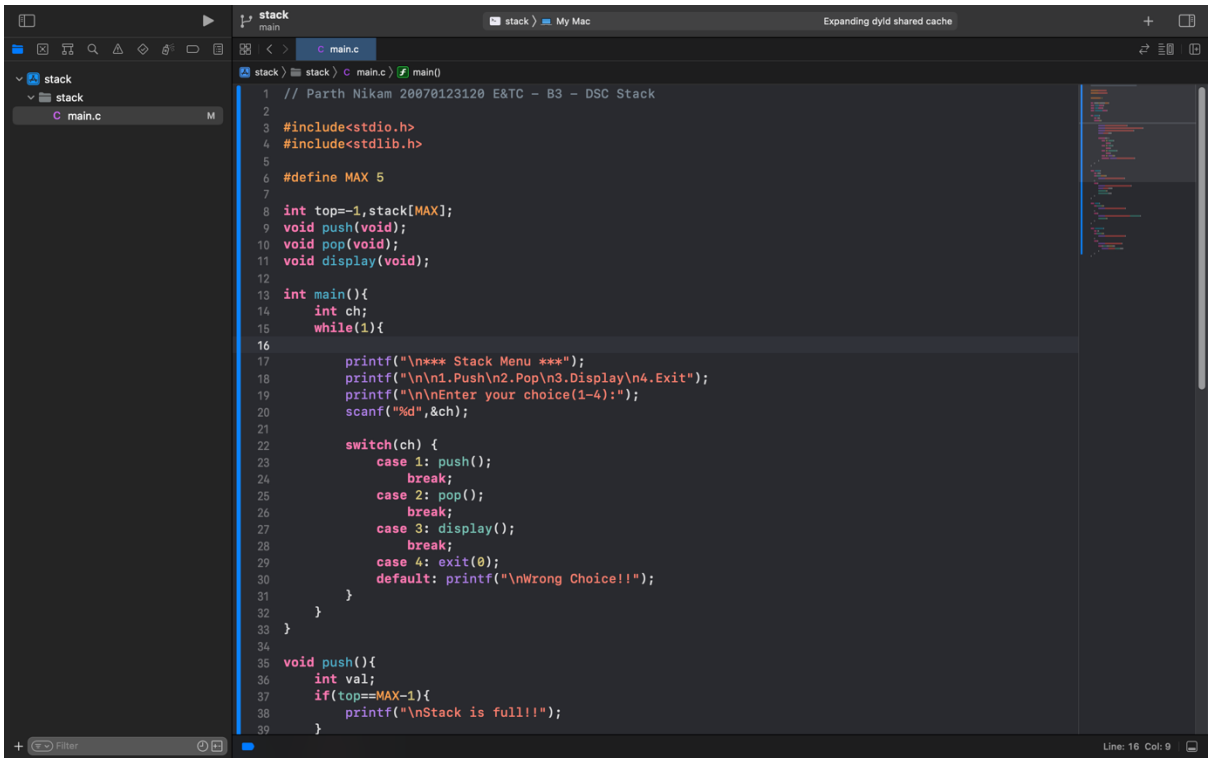
Stacks

```
Void push(){
    int val;
    if (top = MAX-1){
        printf ("\n Stack is full!");}
        Scanf (" %d", & val);
        top = top+1;
        Stack [top] = val;
    }
}

Void pop() {
    if (top==1){
        printf (" Stack is empty!");
    }
    else {
        printf ("\n Deleted element is %d", stack[top]);
        top = top-1;
    }
}

void display() {
    int i;
    if (top==-1){
        printf ("\n Stack is empty!");
    }
    else {
        printf ("\n Stack is... \n");
        for (i= top; i>=0;--i){
            printf (" %d \n", stack [i]);
        }
    }
}
```

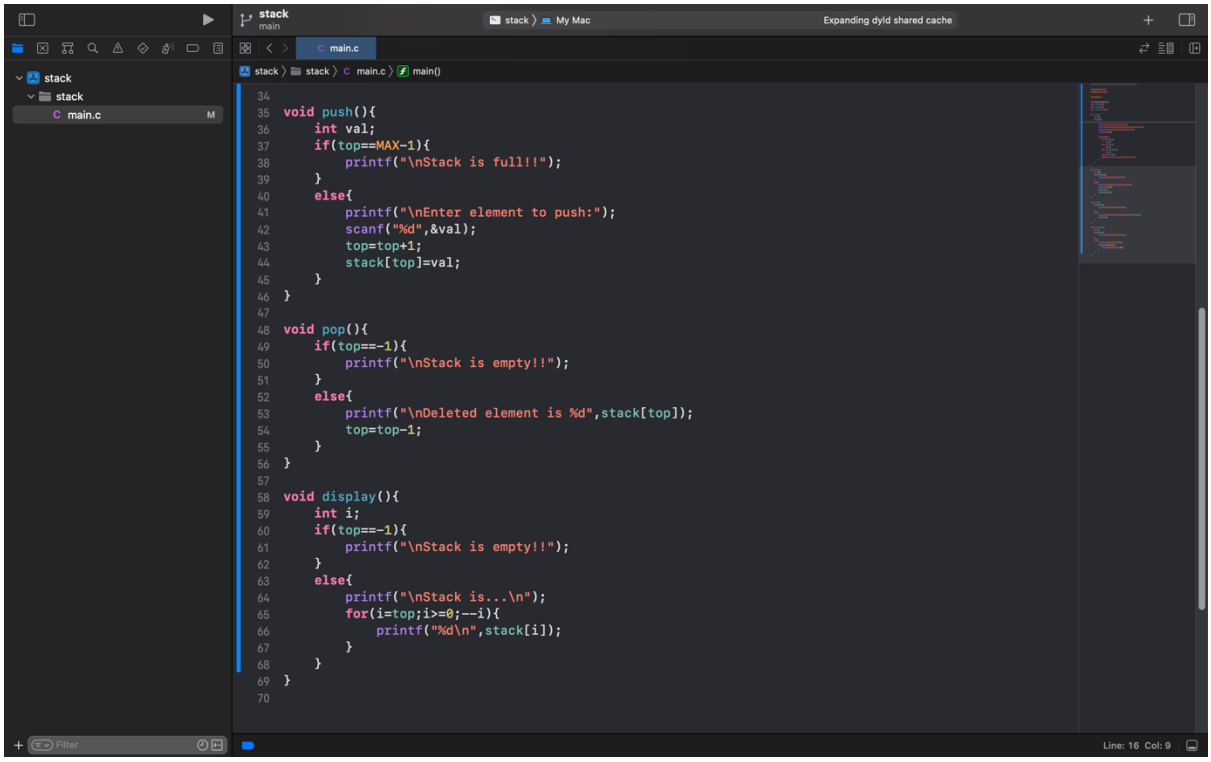
Data Structures using C

Parth Nikam
20070123120
E&TC – B3

Screenshot: -



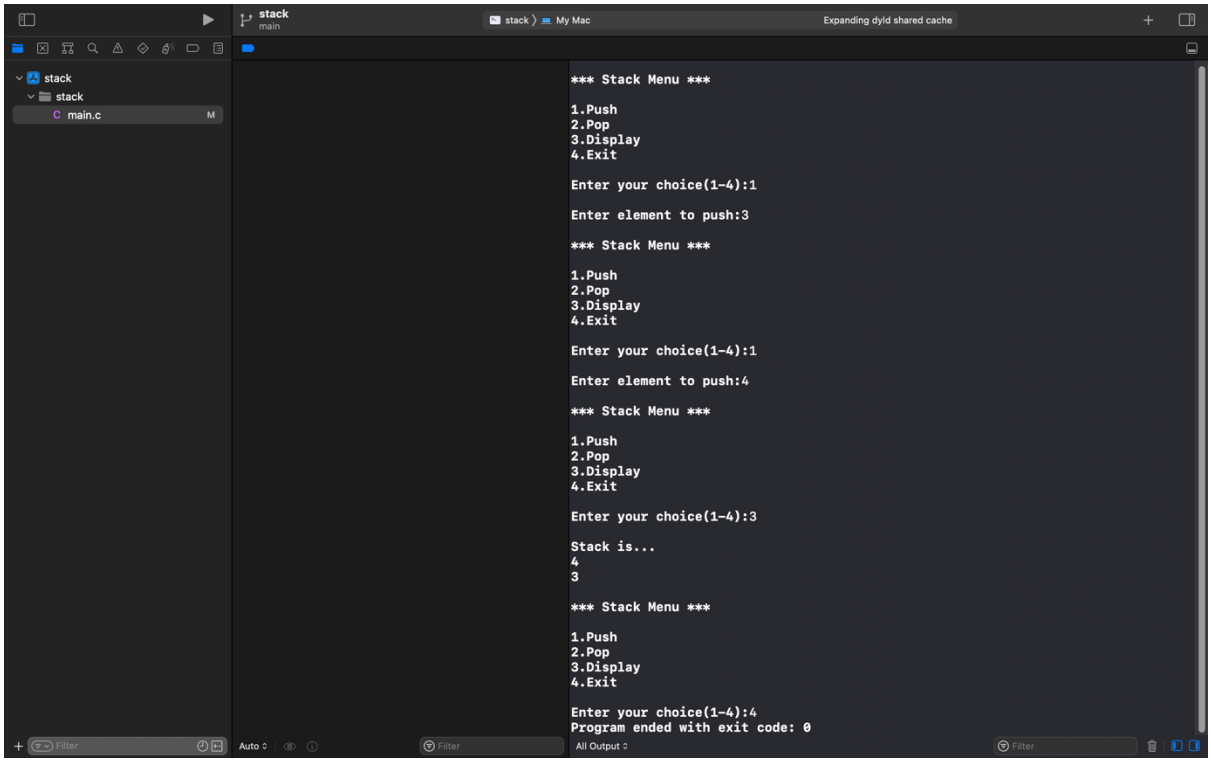
```c
// Parth Nikam 20070123120 E&TC - B3 - DSC Stack

#include<stdio.h>
#include<stdlib.h>

#define MAX 5

int top=-1,stack[MAX];
void push(void);
void pop(void);
void display(void);

int main(){
    int ch;
    while(1){

        printf("\n*** Stack Menu ***");
        printf("\n\n1.Push\n2.Pop\n3.Display\n4.Exit");
        printf("\n\nEnter your choice(1-4):");
        scanf("%d",&ch);

        switch(ch) {
            case 1: push();
                break;
            case 2: pop();
                break;
            case 3: display();
                break;
            case 4: exit(0);
            default: printf("\nWrong Choice!!");
        }
    }
}

void push(){
    int val;
    if(top==MAX-1){
        printf("\nStack is full!!");
    }
```



```c
void push(){
    int val;
    if(top==MAX-1){
        printf("\nStack is full!!");
    }
    else{
        printf("\nEnter element to push:");
        scanf("%d",&val);
        top=top+1;
        stack[top]=val;
    }
}

void pop(){
    if(top==-1){
        printf("\nStack is empty!!");
    }
    else{
        printf("\nDeleted element is %d",stack[top]);
        top=top-1;
    }
}

void display(){
    int i;
    if(top==-1){
        printf("\nStack is empty!!");
    }
    else{
        printf("\nStack is...\n");
        for(i=top;i>=0;--i){
            printf("%d\n",stack[i]);
        }
    }
}
```

Data Structures using C

Parth Nikam
20070123120
E&TC – B3



Result: - Stack is a linear data structure that follows a particular order in which the operations are performed. The order may be LIFO (Last in First Out) or FILO (First in Last Out). It was studied, analyzed, and understood.