VP Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: - DSU	**Subject Code:** 313301
Semester: - III	**Course:** Computer Engineering
Laboratory No: L003	**Name of Subject Teacher:** Prof. Imran S.
Name of Student: Mohd Saad Khan	**Roll Id: -** 24203A0007
Experiment No:	17
Title of Experiment	* Write a 'C' Program to perform PUSH and POP Operations on Stack using Linked List.

Aim: * Write a 'C' Program to perform PUSH and POP Operations on Stack using Linked List.

Algorithm:

Step 1: Start

Step 2: Define a structure Node with members:
- a) int data to store value
- b) struct Node* next to store address of next node

Step 3: Initialize a global pointer top = NULL to represent the top of the stack

Step 4: Define function createNode(data)
- a) Allocate memory for a new node
- b) Store data in the node
- c) Set next = NULL
- d) Return the created node

Step 5: Define function push()
- a) Input data to be pushed
- b) If stack is empty (top == NULL), create a node and assign it to top
- c) Else create a node, set newNode->next = top, update top = newNode

Step 6: Define function pop()
- a) If top == NULL, display "Stack is Empty"
- b) Else print the value of top->data, move top = top->next, free the removed node

Step 7: Define function Display()
- a) Start from top
- b) Traverse the linked list while temp != NULL
- c) Print each node's data
- d) Move to temp->next

Step 8: In main()
- a) Repeat until user exits
  - i) Display menu: 1. Push 2. Pop 3. Display 4. Exit
  - ii) Read user choice c
  - iii) If c == 1, call push()
  - iv) If c == 2, call pop()
  - v) If c == 3, call Display()
  - vi) If c == 4, exit program
  - vii) Else display "Invalid Input"

Step 9: End

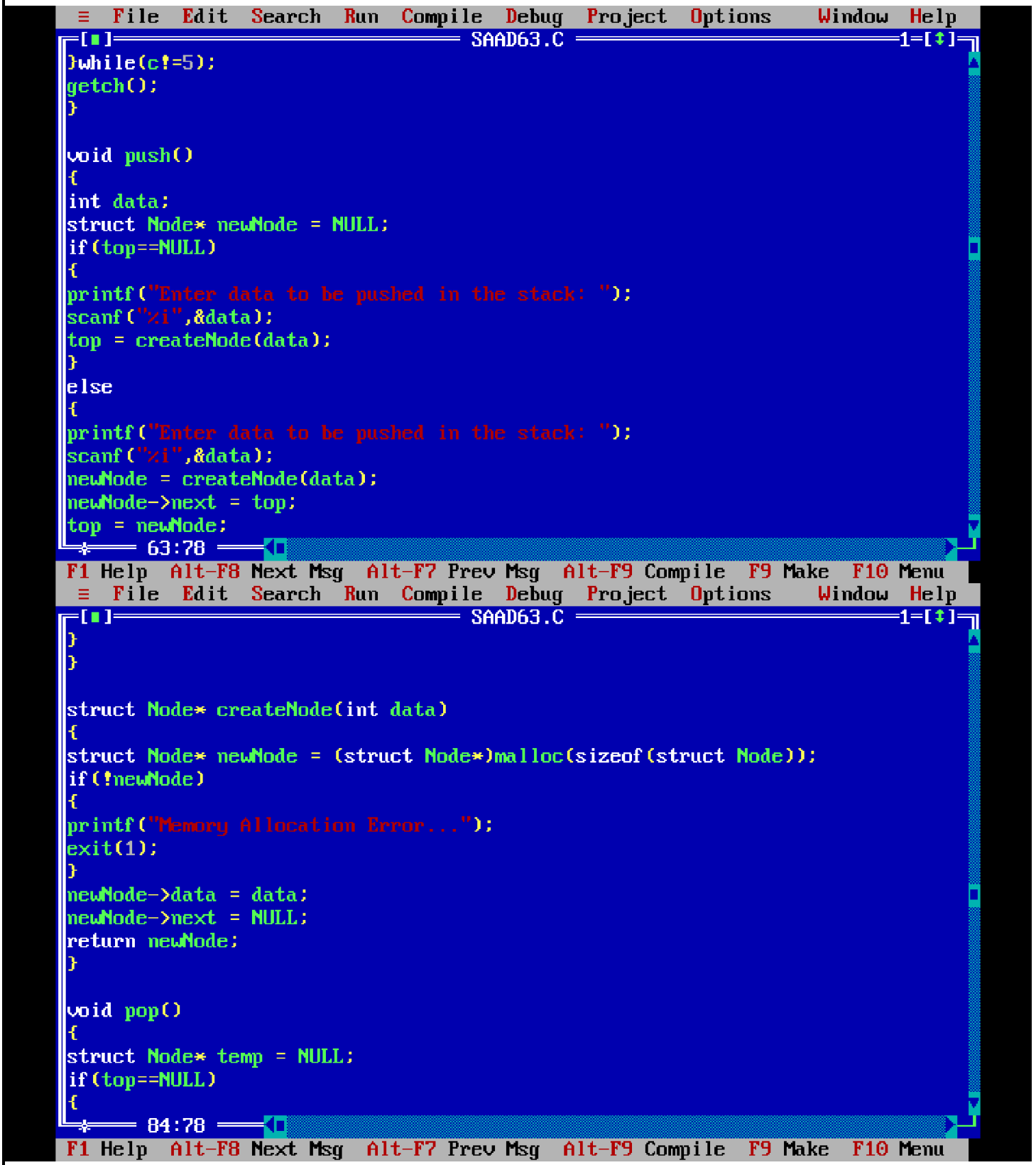
Code:

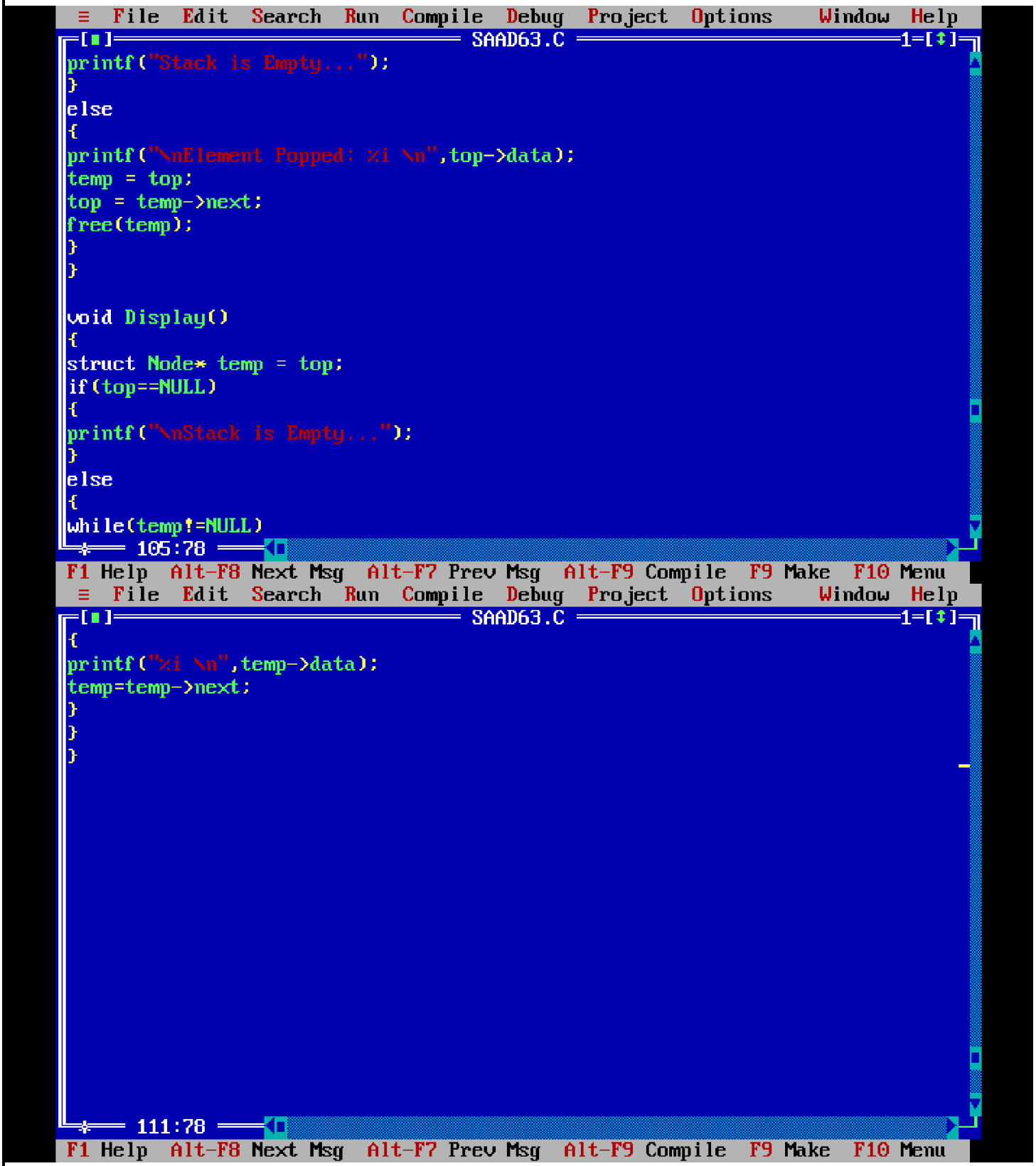
```
≡  File  Edit  Search  Run  Compile  Debug  Project  Options    Window  Help
[■]═══════════ SAAD63.C ═══════════1=[↕]
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct Node
{
int data;
struct Node* next;
};

struct Node* top=NULL;

struct Node* createNode(int);
void push();
void pop();
void Display();

void main()
{
int c;
clrscr();
do
 21:78
F1 Help  Alt-F8 Next Msg  Alt-F7 Prev Msg  Alt-F9 Compile  F9 Make  F10 Menu
≡  File  Edit  Search  Run  Compile  Debug  Project  Options    Window  Help
[■]═══════════ SAAD63.C ═══════════1=[↕]
{
printf("\n1.Push \n2.Pop \n3.Display \n4.Exit \nEnter your Choice: ");
scanf("%i",&c);
switch(c)
{
case 1:
push();
break;
case 2:
pop();
break;
case 3:
printf("\nThe Stack is: \n");
Display();
break;
case 4:
exit(0);
break;
default:
printf("Invalid Input...");
}
 42:78
F1 Help  Alt-F8 Next Msg  Alt-F7 Prev Msg  Alt-F9 Compile  F9 Make  F10 Menu
```


File Edit Search Run Compile Debug Project Options Window Help
SAAD63.C
}while(c!=5);
getch();
}
void push()
{
int data;
struct Node* newNode = NULL;
if(top==NULL)
{
printf("Enter data to be pushed in the stack: ");
scanf("%i",&data);
top = createNode(data);
}
else
{
printf("Enter data to be pushed in the stack: ");
scanf("%i",&data);
newNode = createNode(data);
newNode->next = top;
top = newNode;
63:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
File Edit Search Run Compile Debug Project Options Window Help
SAAD63.C
}
}
struct Node* createNode(int data)
{
struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
if(!newNode)
{
printf("Memory Allocation Error...");
exit(1);
}
newNode->data = data;
newNode->next = NULL;
return newNode;
}
void pop()
{
struct Node* temp = NULL;
if(top==NULL)
{
84:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
SAAD63.C
printf("Stack is Empty...");
}
else
{
printf("\nElement Popped: %i \n",top->data);
temp = top;
top = temp->next;
free(temp);
}
}
void Display()
{
struct Node* temp = top;
if(top==NULL)
{
printf("\nStack is Empty...");
}
else
{
while(temp!=NULL)
105:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
File Edit Search Run Compile Debug Project Options Window Help
SAAD63.C
{
printf("%i \n",temp->data);
temp=temp->next;
}
}
}
111:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

Output: -

```
1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 12

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 23

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 34

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 45

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 56

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 3

The Stack is:
56
45
34
23
12
```

```
1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 2

Element Popped: 56

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 2

Element Popped: 45
```

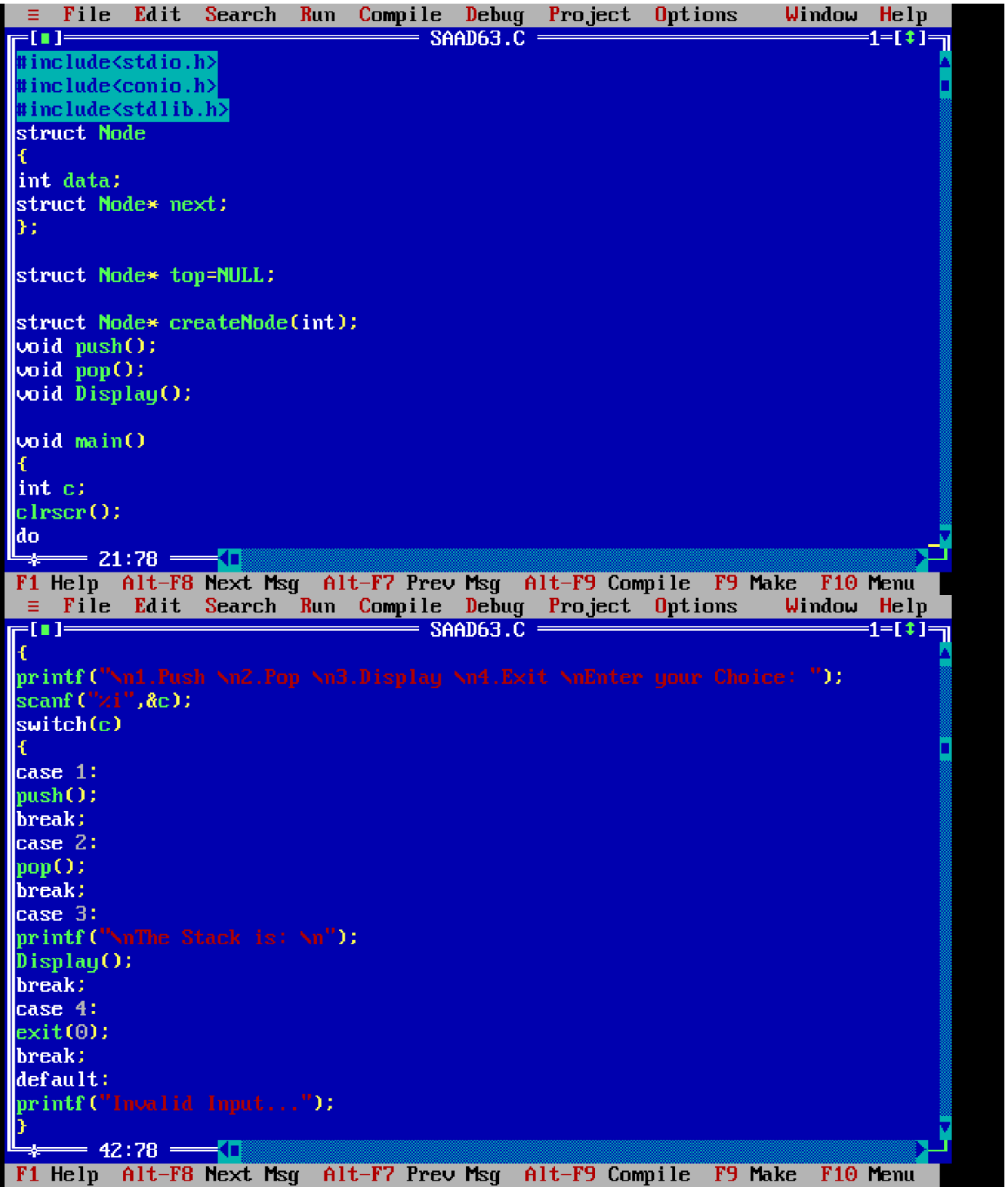
```
1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 3

The Stack is:
34
23
12

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 4
```

Practical Related Questions:

1. **Write a C program to perform following operations on Stack as Linked List.**
   PUSH (10), PUSH (20), POP, PUSH (10), PUSH (20), POP, PUSH (20), POP.
   Ans:

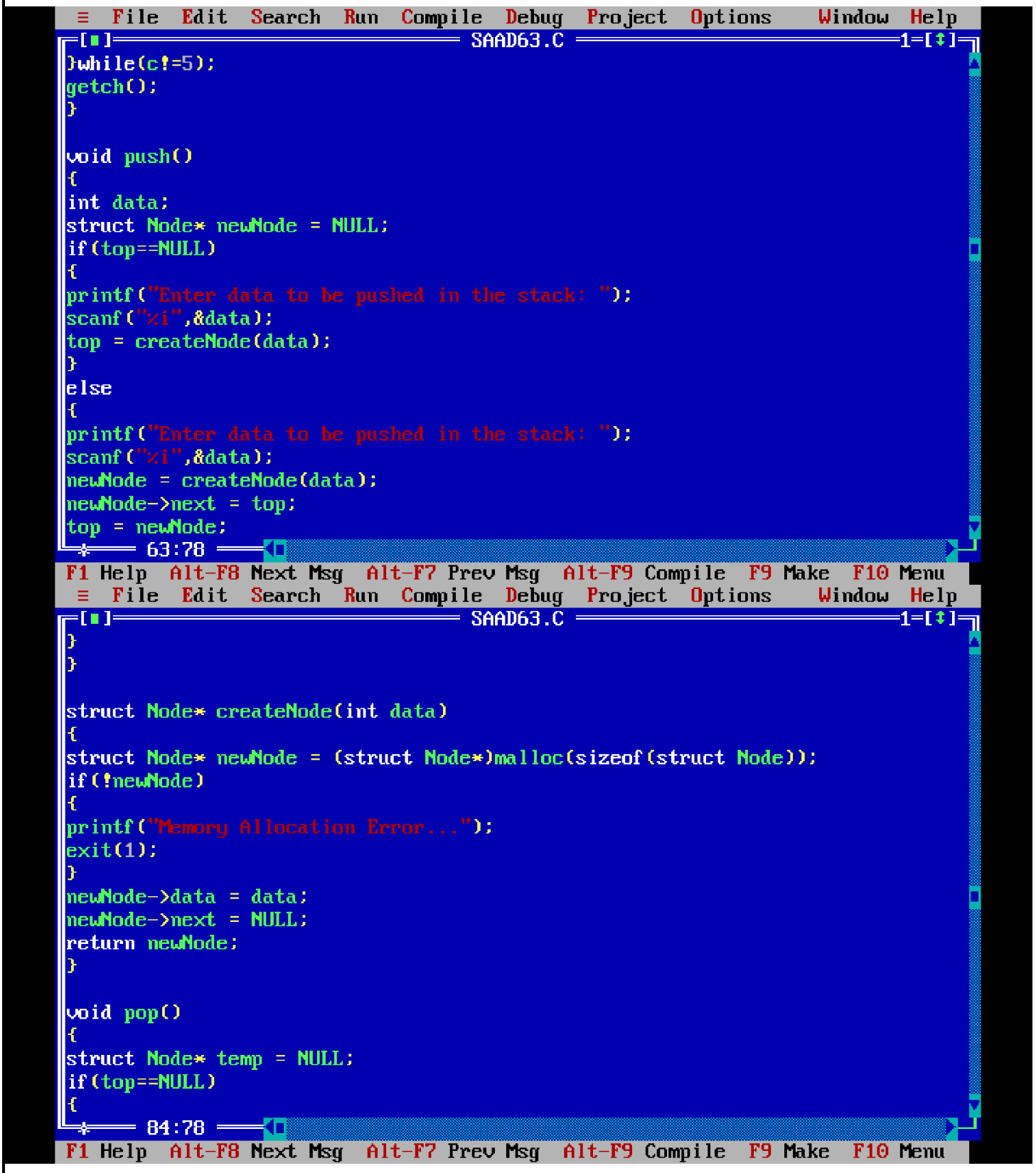


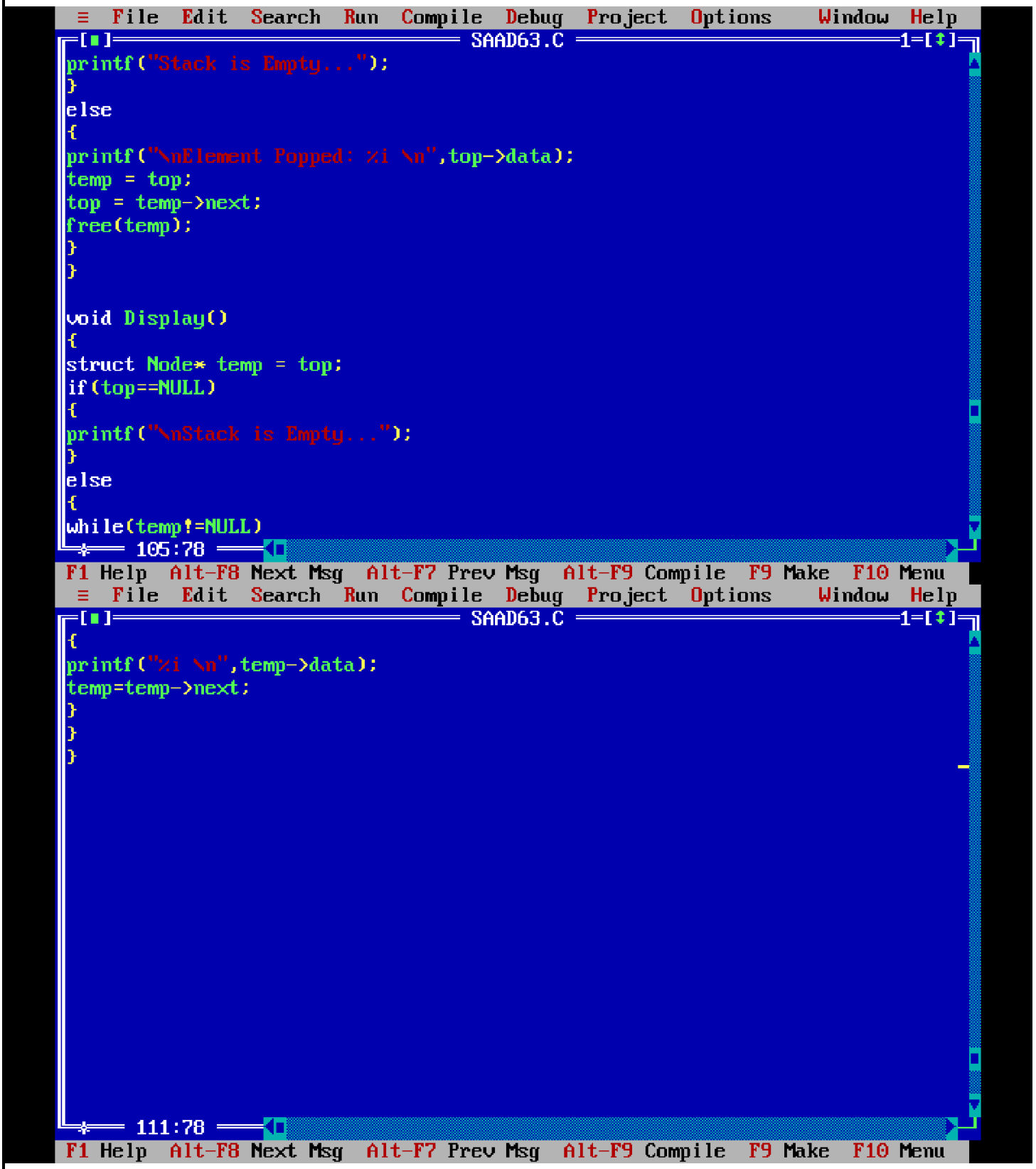
```
 ≡  File  Edit  Search  Run  Compile  Debug  Project  Options    Window  Help
[■]═══════════════ SAAD63.C ═══════════════1=[↕]
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct Node
{
int data;
struct Node* next;
};

struct Node* top=NULL;

struct Node* createNode(int);
void push();
void pop();
void Display();

void main()
{
int c;
clrscr();
do
 21:78
F1 Help  Alt-F8 Next Msg  Alt-F7 Prev Msg  Alt-F9 Compile  F9 Make  F10 Menu
 ≡  File  Edit  Search  Run  Compile  Debug  Project  Options    Window  Help
[■]═══════════════ SAAD63.C ═══════════════1=[↕]
{
printf("\n1.Push \n2.Pop \n3.Display \n4.Exit \nEnter your Choice: ");
scanf("%i",&c);
switch(c)
{
case 1:
push();
break;
case 2:
pop();
break;
case 3:
printf("\nThe Stack is: \n");
Display();
break;
case 4:
exit(0);
break;
default:
printf("Invalid Input...");
}
 42:78
F1 Help  Alt-F8 Next Msg  Alt-F7 Prev Msg  Alt-F9 Compile  F9 Make  F10 Menu
```


File Edit Search Run Compile Debug Project Options Window Help
SAAD63.C
}while(c!=5);
getch();
}
void push()
{
int data;
struct Node* newNode = NULL;
if(top==NULL)
{
printf("Enter data to be pushed in the stack: ");
scanf("%i",&data);
top = createNode(data);
}
else
{
printf("Enter data to be pushed in the stack: ");
scanf("%i",&data);
newNode = createNode(data);
newNode->next = top;
top = newNode;
63:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
File Edit Search Run Compile Debug Project Options Window Help
SAAD63.C
}
}
struct Node* createNode(int data)
{
struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
if(!newNode)
{
printf("Memory Allocation Error...");
exit(1);
}
newNode->data = data;
newNode->next = NULL;
return newNode;
}
void pop()
{
struct Node* temp = NULL;
if(top==NULL)
{
84:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
SAAD63.C
printf("Stack is Empty...");
}
else
{
printf("\nElement Popped: %i \n",top->data);
temp = top;
top = temp->next;
free(temp);
}
}
void Display()
{
struct Node* temp = top;
if(top==NULL)
{
printf("\nStack is Empty...");
}
else
{
while(temp!=NULL)
105:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
File Edit Search Run Compile Debug Project Options Window Help
SAAD63.C
{
printf("%i \n",temp->data);
temp=temp->next;
}
}
}
111:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

Output: -

```
1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 10

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 20

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 2

Element Popped: 20

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 10

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 20

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 2

Element Popped: 20
```

```
1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 1
Enter data to be pushed in the stack: 20

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 2

Element Popped: 20

1.Push
2.Pop
3.Display
4.Exit
Enter your Choice: 4
```

2. **What is the time complexity of the push, pop, and peek operations in your implementation?**
   Ans:
   In a stack, the push, pop, and peek operations all work on the top element only.
   - Push adds a new element at the top.
   - Pop removes the top element.
   - Peek just retrieves the top element without removing it.

   Since none of these operations require traversing the stack or shifting elements, each of them takes constant time.
   Therefore, the time complexity of push, pop, and peek is O(1).

Marks Obtained			**Dated signature of Teacher**
Process Related (35)	**Product Related (15)**	**Total (50)**	