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:	16
Title of Experiment	* Write a 'C' Program to perform PUSH and POP Operations on Stack using an Array.

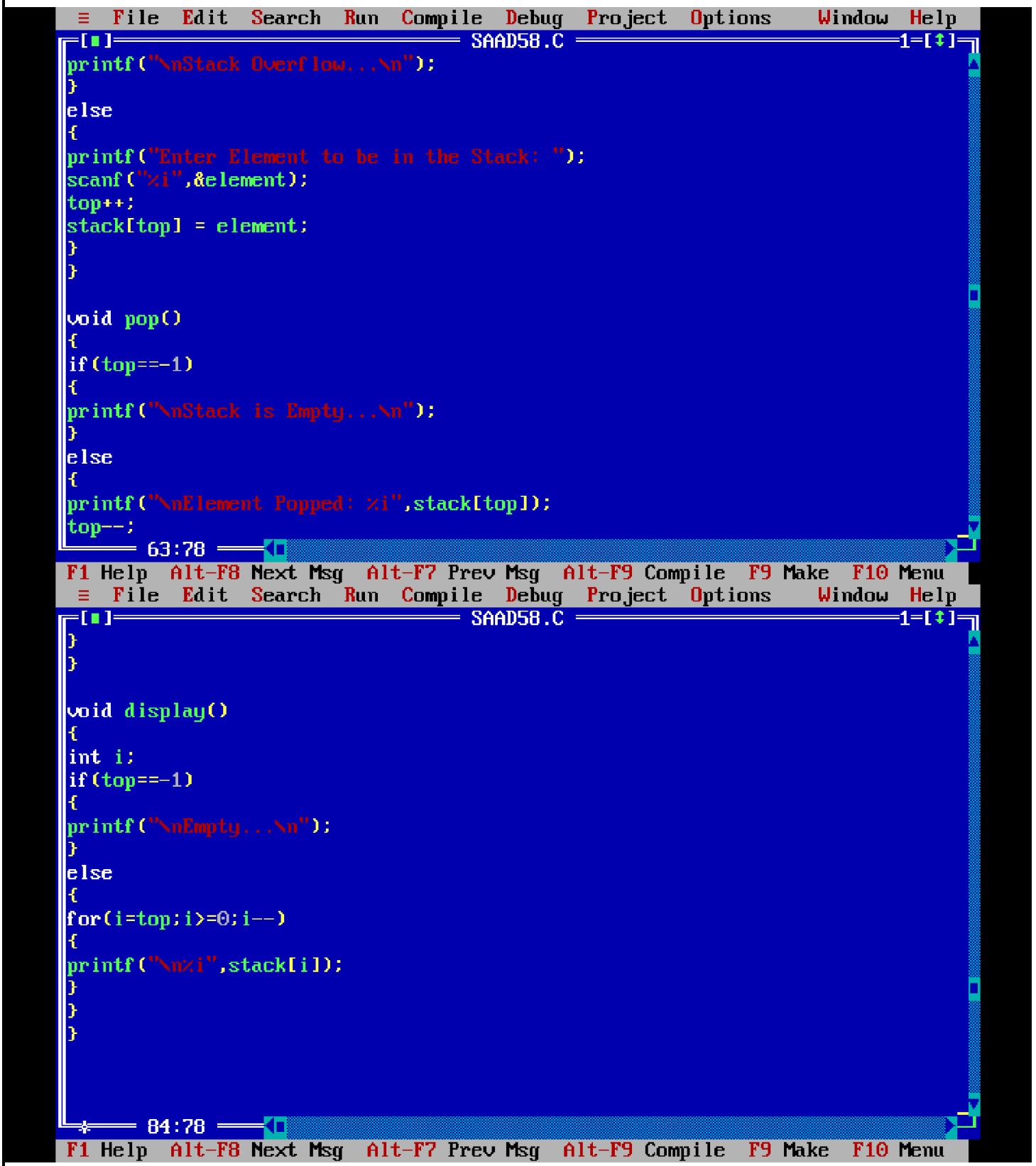
Aim: * Write a 'C' Program to perform PUSH and POP Operations on Stack using an Array.

Algorithm:

Step 1: Start
Step 2: Initialize stack[max] and set top = -1
Step 3: Display menu:
1 → Push
2 → Pop
3 → Display
4 → Exit
Step 4: Read user's choice
Step 5: If choice = 1, then call Push operation
  Step 5.1: Check if top == max-1
    If true, print "Stack Overflow"
    Else, read element, increment top, and store stack[top] = element
Step 6: If choice = 2, then call Pop operation
  Step 6.1: Check if top == -1
    If true, print "Stack is Empty"
    Else, print stack[top], then decrement top
Step 7: If choice = 3, then call Display operation
  Step 7.1: Check if top == -1
    If true, print "Stack is Empty"
    Else, print elements from stack[top] down to stack[0]
Step 8: If choice = 4, Exit the program
Step 9: Repeat Steps 3–8 until choice = 4
Step 10: Stop

Code:

```
 ≡  File  Edit  Search  Run  Compile  Debug  Project  Options    Window  Help
[■]═══════════════ SAAD58.C ═══════════════1=[↕]
#include<stdio.h>
#include<conio.h>
#define max 5
void push();
void pop();
void display();
int stack[max];
int top=-1;
void main()
{
int choice;
clrscr();
do
{
printf("\n1.Push \n2.Pop \n3.Display \n4.Exit \nEnter your choice: ");
scanf("%i",&choice);
switch(choice)
{
case 1:
push();
break;
═══ 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
[■]═══════════════ SAAD58.C ═══════════════1=[↕]
case 2:
pop();
break;
case 3:
display();
break;
case 4:
choice = 4;
break;
default:
printf("Invalid Input...");
}
}while(choice!=4);
getch();
}

void push()
{
int element;
if(top==max-1)
{
═══ 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
SAAD58.C
printf("\nStack Overflow...\n");
}
else
{
printf("Enter Element to be in the Stack: ");
scanf("%i",&element);
top++;
stack[top] = element;
}
}
void pop()
{
if(top==-1)
{
printf("\nStack is Empty...\n");
}
else
{
printf("\nElement Popped: %i",stack[top]);
top--;
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
SAAD58.C
}
}
void display()
{
int i;
if(top==-1)
{
printf("\nEmpty...\n");
}
else
{
for(i=top;i>=0;i--)
{
printf("\n%i",stack[i]);
}
}
}
84:78
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 Element to be in the Stack: 12

1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 1
Enter Element to be in the Stack: 23

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

23
12
1.Push
2.Pop
3.Display
4.Exit
Enter your choice:
2

Element Popped: 23
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 3

12
```

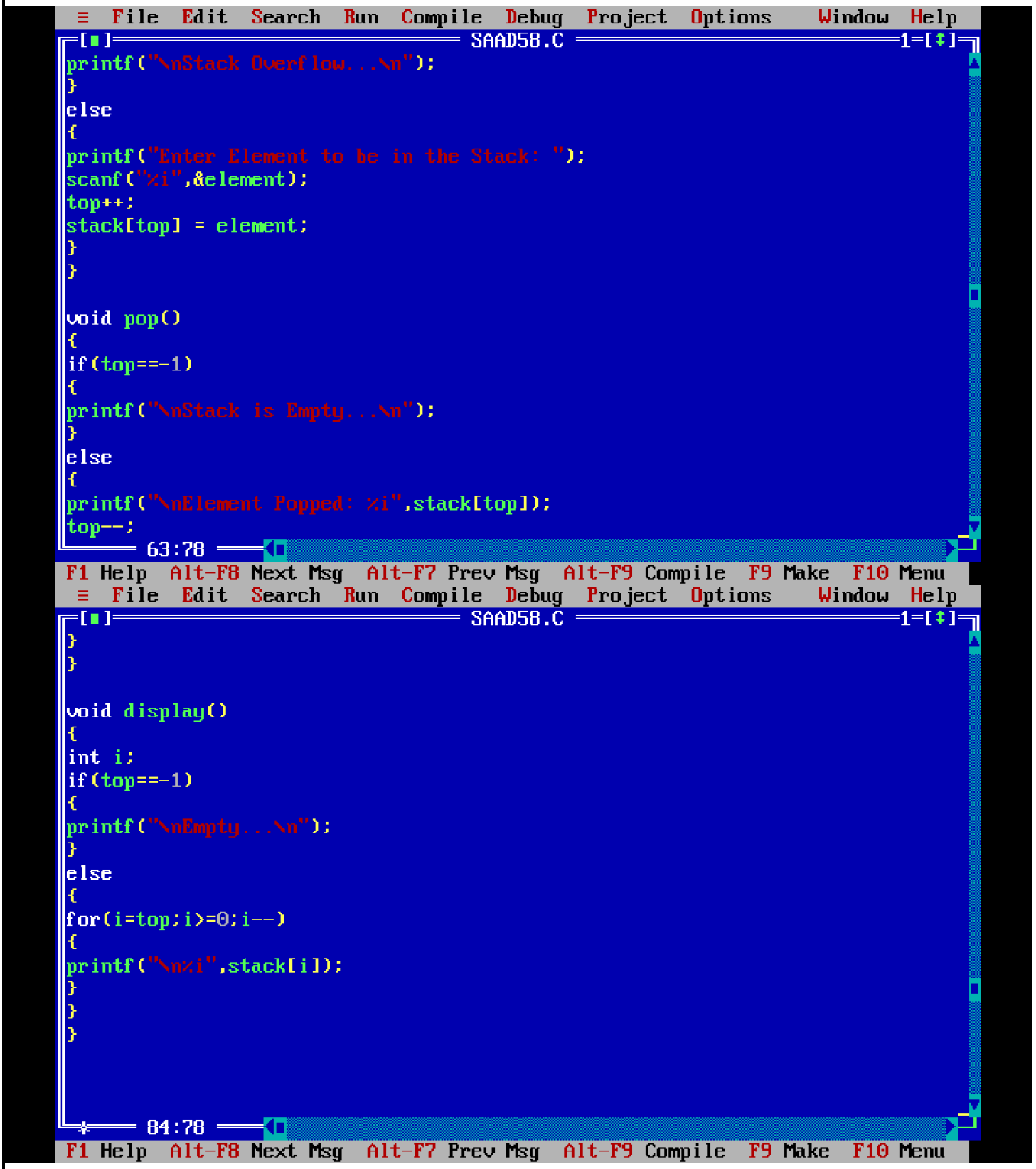
```
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 4
```

Practical Related Questions:

1. **Write a C program to perform following operations on Empty stack:**
   a. **PUSH (10), PUSH (20), POP, PUSH (10), PUSH (20), POP, POP, POP, PUSH (20), POP.**

Ans:

```
 ≡  File  Edit  Search  Run  Compile  Debug  Project  Options    Window  Help
[■]═══════════ SAAD58.C ═══════════1=[↕]
#include<stdio.h>
#include<conio.h>
#define max 5
void push();
void pop();
void display();
int stack[max];
int top=-1;
void main()
{
int choice;
clrscr();
do
{
printf("\n1.Push \n2.Pop \n3.Display \n4.Exit \nEnter your choice: ");
scanf("%i",&choice);
switch(choice)
{
case 1:
push();
break;
═══ 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
[■]═══════════ SAAD58.C ═══════════1=[↕]
case 2:
pop();
break;
case 3:
display();
break;
case 4:
choice = 4;
break;
default:
printf("Invalid Input...");
}
}while(choice!=4);
getch();
}

void push()
{
int element;
if(top==max-1)
{
═══ 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
SAAD58.C
printf("\nStack Overflow...\n");
}
else
{
printf("Enter Element to be in the Stack: ");
scanf("%i",&element);
top++;
stack[top] = element;
}
}
void pop()
{
if(top==-1)
{
printf("\nStack is Empty...\n");
}
else
{
printf("\nElement Popped: %i",stack[top]);
top--;
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
SAAD58.C
}
}
void display()
{
int i;
if(top==-1)
{
printf("\nEmpty...\n");
}
else
{
for(i=top;i>=0;i--)
{
printf("\n%i",stack[i]);
}
}
}
84:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

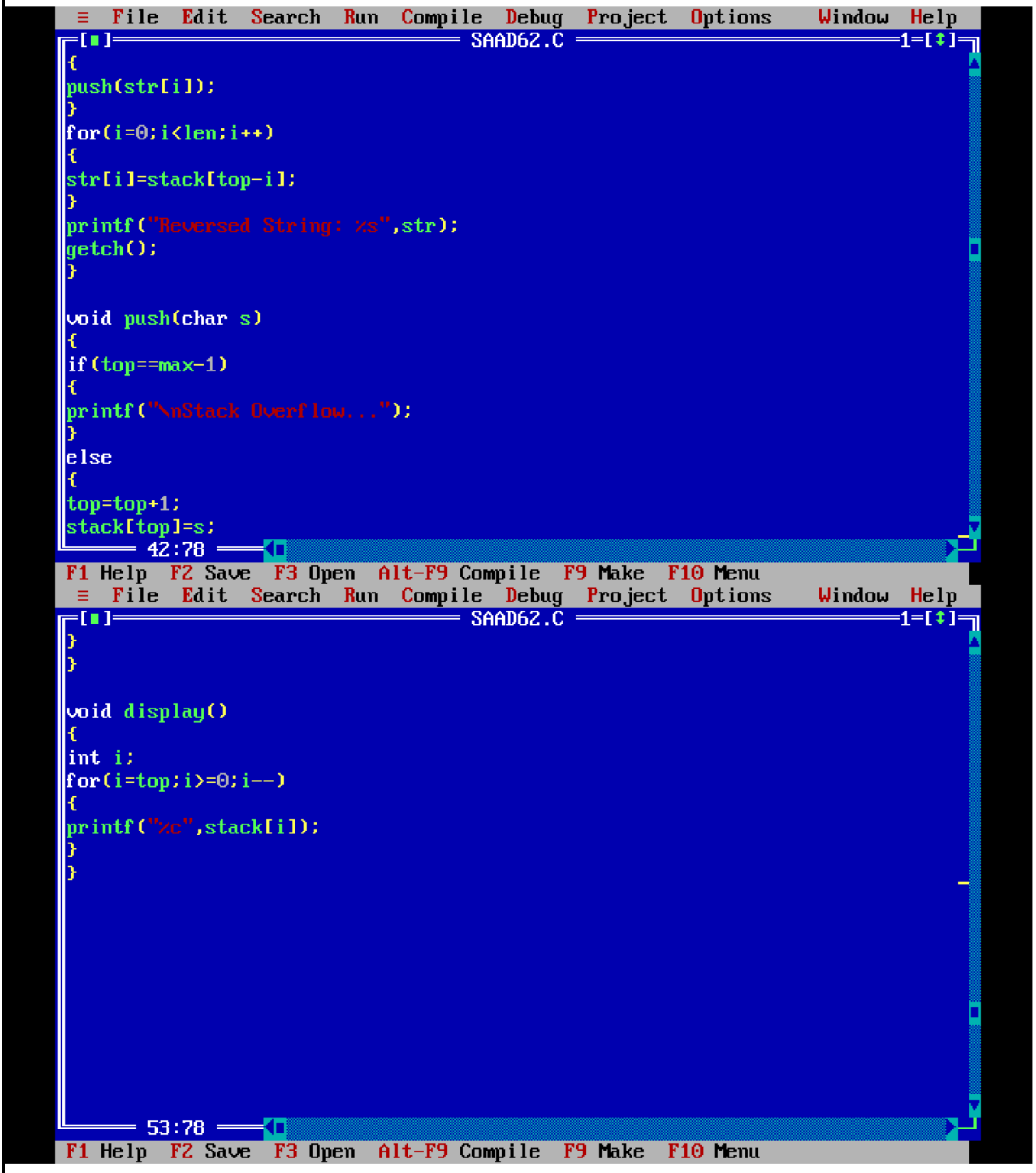
Output: -

```
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 1
Enter Element to be in the Stack: 10

1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 1
Enter Element to be 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 Element to be in the Stack: 10

1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 1
Enter Element to be 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: 2

Element Popped: 10
```

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

Element Popped: 10
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 1
Enter Element to be in the Stack: 20

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

Element Popped: 20
```

2. **Write a C program to reverse a String using Stack.**
   Ans:

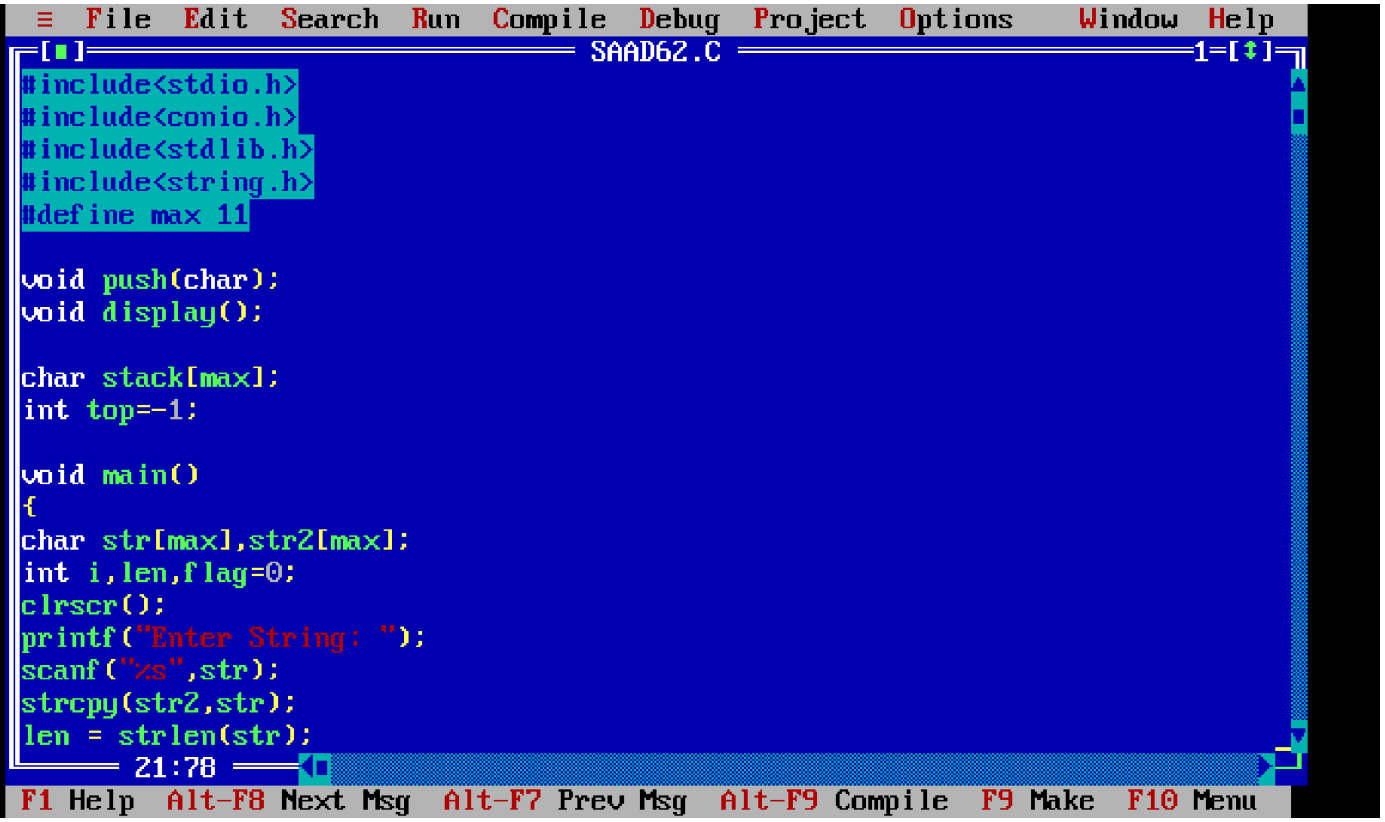
```
≡ File  Edit  Search  Run  Compile  Debug  Project  Options  Window  Help
[■]═══ SAAD62.C ═══1=[↕]
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#define max 11

void push(char);
void display();

char stack[max];
int top=-1;

void main()
{
char str[max];
int i,len;
clrscr();
printf("Enter String: ");
scanf("%s",str);
len = strlen(str);
for(i=0;i<len;i++)
═══ 21:78 ═══
F1 Help | File-management commands (Open, Save, Print, etc.)
```

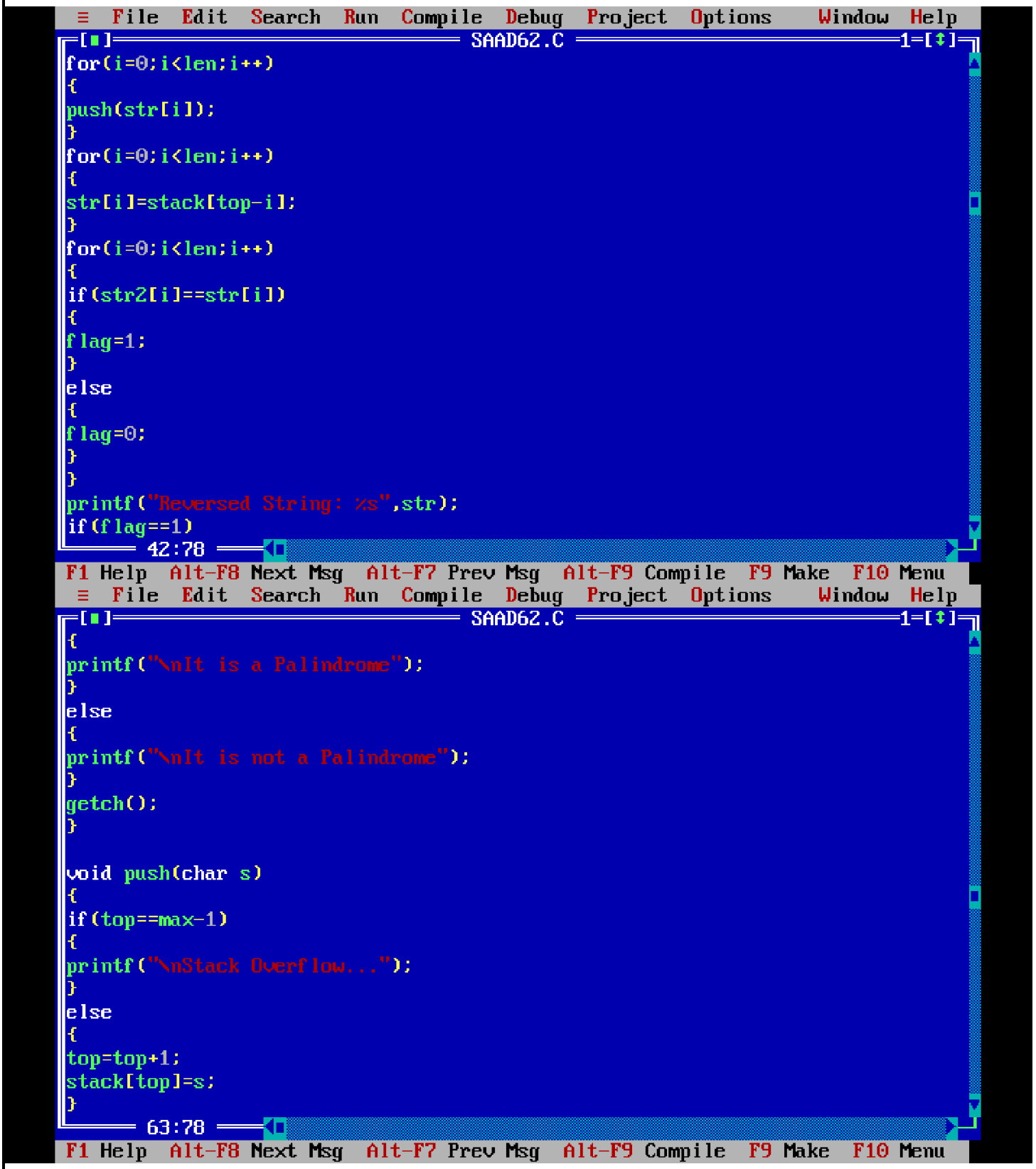

File Edit Search Run Compile Debug Project Options Window Help
SAAD62.C
1=[↕]
{
push(str[i]);
}
for(i=0;i<len;i++)
{
str[i]=stack[top-i];
}
printf("Reversed String: %s",str);
getch();
}
void push(char s)
{
if(top==max-1)
{
printf("\nStack Overflow...");
}
else
{
top=top+1;
stack[top]=s;
42:78
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
File Edit Search Run Compile Debug Project Options Window Help
SAAD62.C
1=[↕]
}
}
void display()
{
int i;
for(i=top;i>=0;i--)
{
printf("%c",stack[i]);
}
}
53:78
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

Output:

```
Enter String: Hello
Reversed String: olleH_
```

3. **Write a C program to check whether the given string is palindrome or not using stack.**
   Ans:




File Edit Search Run Compile Debug Project Options Window Help
SAAD62.C
1=[↕]
for(i=0;i<len;i++)
{
push(str[i]);
}
for(i=0;i<len;i++)
{
str[i]=stack[top-i];
}
for(i=0;i<len;i++)
{
if(str2[i]==str[i])
{
flag=1;
}
else
{
flag=0;
}
}
printf("Reversed String: %s",str);
if(flag==1)
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
SAAD62.C
1=[↕]
{
printf("\nIt is a Palindrome");
}
else
{
printf("\nIt is not a Palindrome");
}
getch();
}
void push(char s)
{
if(top==max-1)
{
printf("\nStack Overflow...");
}
else
{
top=top+1;
stack[top]=s;
}
63:78
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
}

void display()
{
int i;
for(i=top;i>=0;i--)
{
printf("%c",stack[i]);
}
}
```

Output:

```
Enter String: amima
Reversed String: amima
It is a Palindrome_
```

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