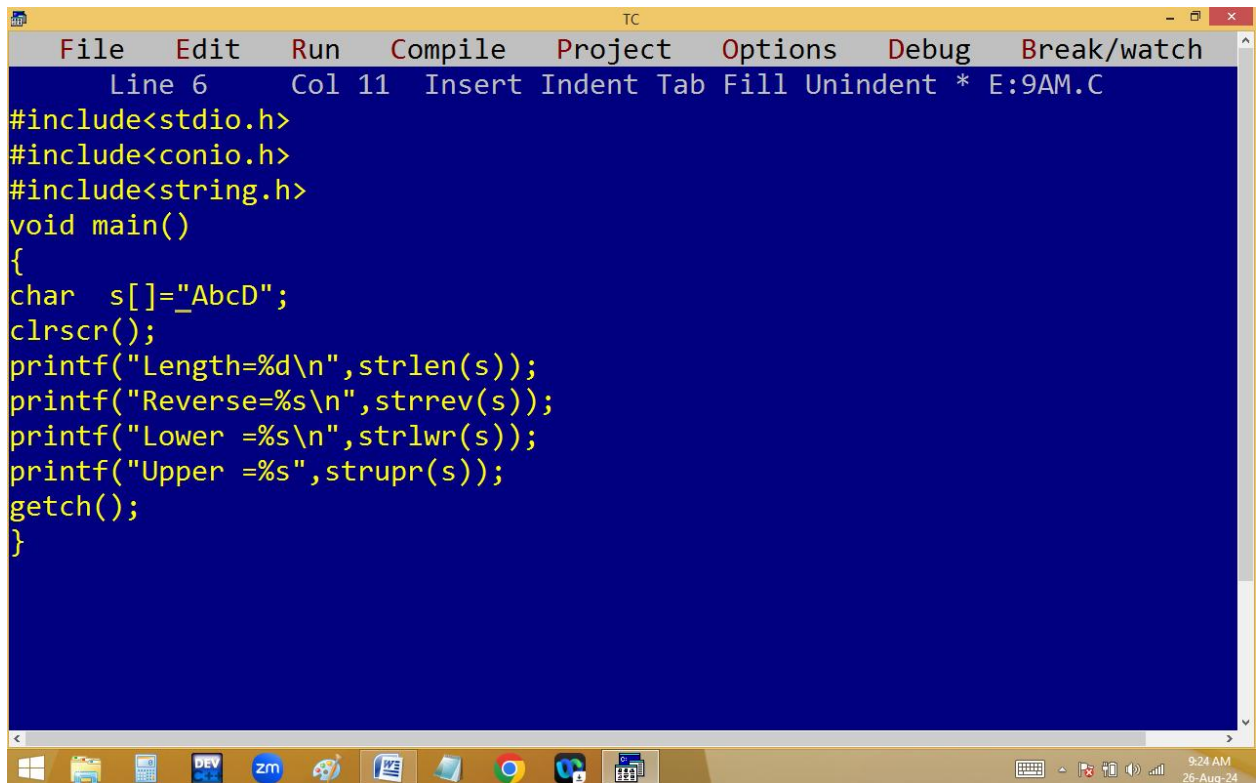


STRING LIBRARY FUNCTIONS

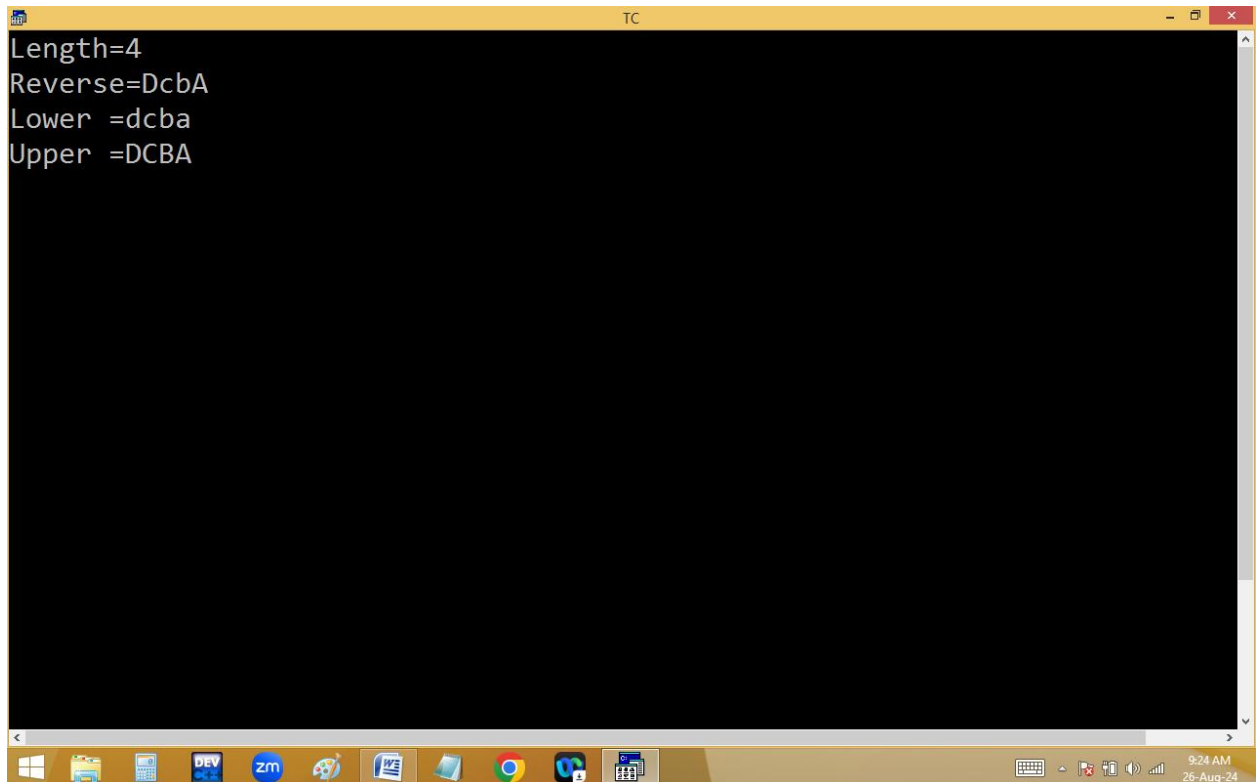
To manage string operations c language provides some of the predefined functions available in <string.h>

They are

1. **strlen():** It return string length.
Syntax: **strlen(string);**
2. **strrev():** it return reverse string.
Syntax: **strrev(string);**
3. **strlwr():** converts string into lower case.
Syntax: **strlwr(string);**
4. **strupr():** converts string into upper case.
Syntax: **strupr(string);**

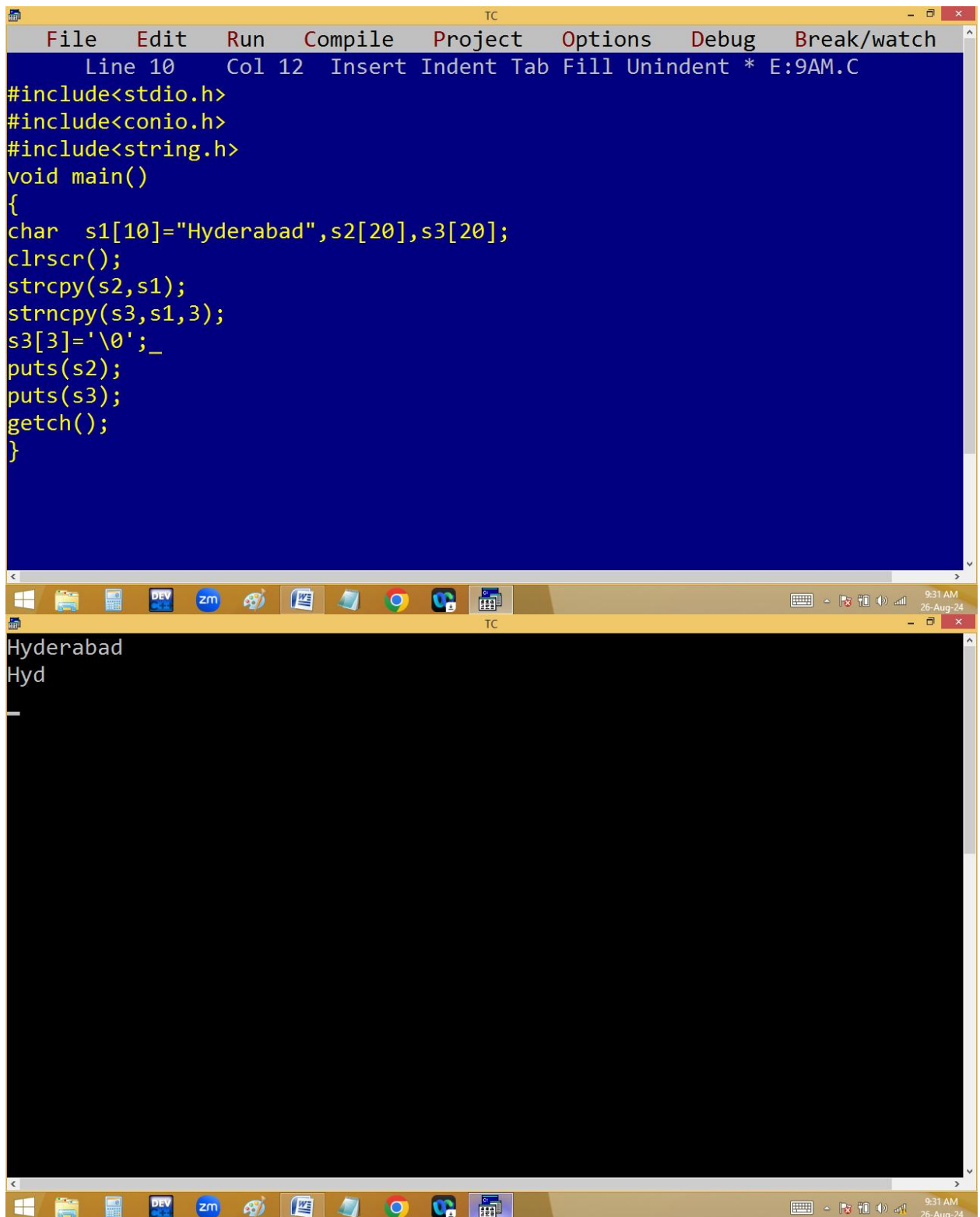


```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 6 Col 11 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[]="AbcD";
clrscr();
printf("Length=%d\n",strlen(s));
printf("Reverse=%s\n",strrev(s));
printf("Lower =%s\n",strlwr(s));
printf("Upper =%s",strupr(s));
getch();
}
```

A screenshot of a Turbo C++ (TC) window. The window has a yellow title bar with the text "TC" in the center. The main area is black with white text. The text is as follows:
Length=4
Reverse=DcbA
Lower =dcba
Upper =DCBA
At the bottom of the window is a Windows taskbar with various icons including Windows, File Explorer, DEV, zm, a folder icon, a document icon, Google Chrome, and a calendar icon. The system tray on the right shows the time as 9:24 AM and the date as 26-Aug-24.

```
Length=4
Reverse=DcbA
Lower =dcba
Upper =DCBA
```

- 5. **strcpy():** It copies the source string into destination string.
Syntax: **strcpy(destination string, source string);**
- 6. **strncpy():** It copies specified no of char into destination string.
Syntax: **strncpy(dest string, source string, no of char);**



The screenshot shows the Turbo C++ (TC) IDE. The top window is the code editor, which has a blue background and contains the following C code:

```
File Edit Run Compile Project Options Debug Break/watch
Line 10 Col 12 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s1[10]="Hyderabad",s2[20],s3[20];
clrscr();
strcpy(s2,s1);
strncpy(s3,s1,3);
s3[3]='\0';_
puts(s2);
puts(s3);
getch();
}
```

The bottom window is the output console, which has a black background and shows the output of the program:

```
Hyderabad
Hyd
```

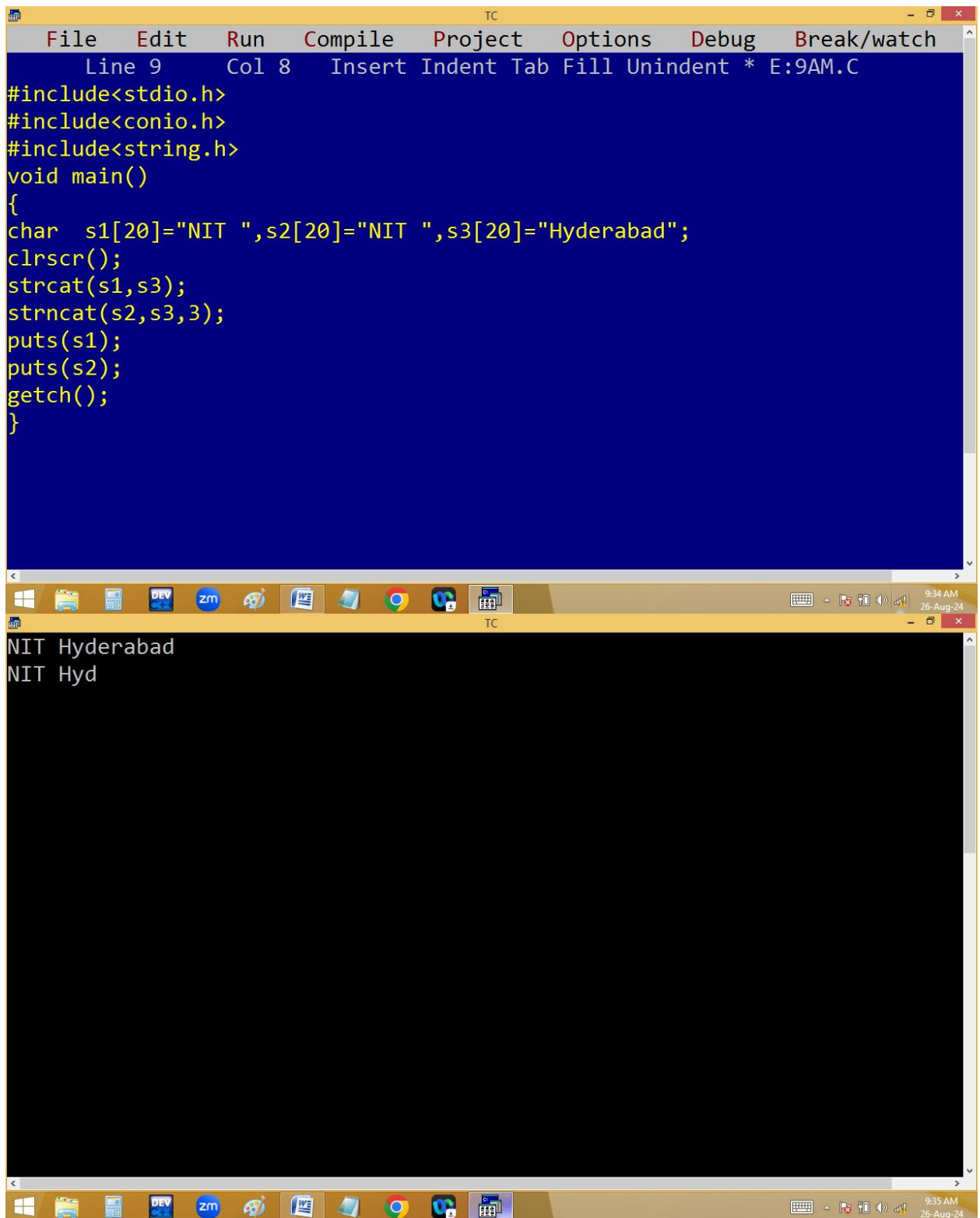
The Windows taskbar is visible at the bottom of the screen, showing various application icons and the system clock indicating 9:31 AM on 26-Aug-24.

7. strcat(): It adds string2 to string1.

Syntax: strcat(string1, string2);

8. **strncat()**: It adds specified no of char's to string1.

Syntax: **strncat(string1, string2, no of char);**



```
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 8 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s1[20]="NIT ",s2[20]="NIT ",s3[20]="Hyderabad";
clrscr();
strcat(s1,s3);
strncat(s2,s3,3);
puts(s1);
puts(s2);
getch();
}
```

NIT Hyderabad
NIT Hyd

9. strcmp(): It compare two strings until the first ascii value difference found.

Syntax: strcmp(string1, string2);

The image shows a screenshot of a Turbo C++ (TC) IDE. The top window displays a C program with the following code:

```
Line 13   Col 32   Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
clrscr();
printf("%d\n",strcmp("ab","ab"));
printf("%d\n",strcmp("ab","AB"));
printf("%d\n",strcmp("AB","ab"));
printf("%d\n",strcmp("abc","ab"));
printf("%d\n",strcmp("ab","abc"));
printf("%d\n",strcmp("ab","1234"));
getch();
}
```

The bottom window shows the output of the program, which consists of six integers printed on separate lines:

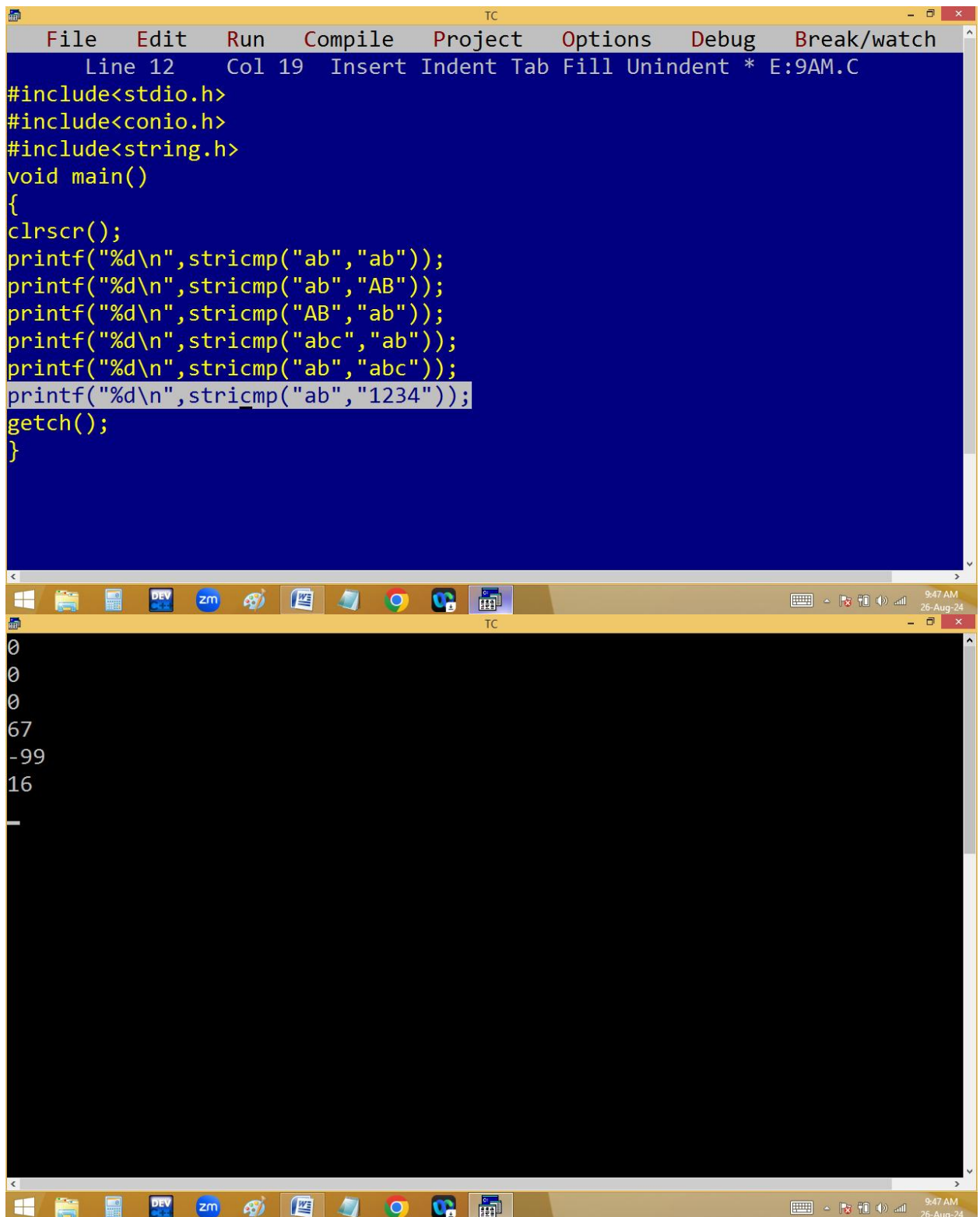
```
0
32
-32
99
-99
48
```

The IDE's taskbar at the bottom shows various icons including Windows, File Explorer, DEV, zm, and others. The system clock in the bottom right corner indicates 9:41 AM on 26-Aug-24.

10. **stricmp():** it checks the strings by ignoring case. i.e. in **stricmp()** lower and upper are same. If matching char not found

or different data type found in 2nd string, the first string char taken in upper case.

Syntax: `stricmp(string1, string2);`

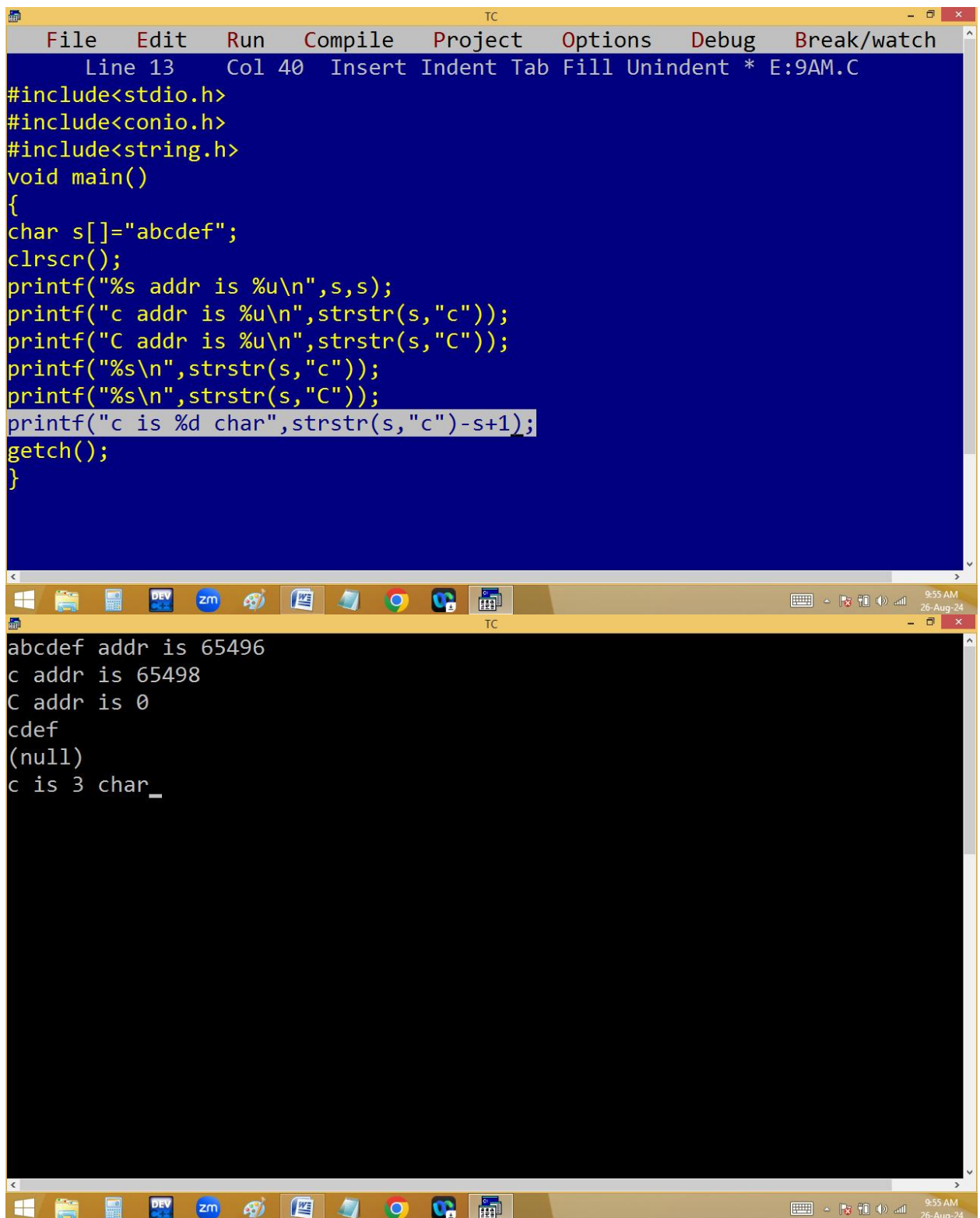


```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 12 Col 19 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
clrscr();
printf("%d\n",strcmp("ab","ab"));
printf("%d\n",strcmp("ab","AB"));
printf("%d\n",strcmp("AB","ab"));
printf("%d\n",strcmp("abc","ab"));
printf("%d\n",strcmp("ab","abc"));
printf("%d\n",strcmp("ab","1234"));
getch();
}
```

```
0
0
0
67
-99
16
_
```

11. strstr(): It searches the sub string in main string and if found, it return sub string address. If not found, it return 0.

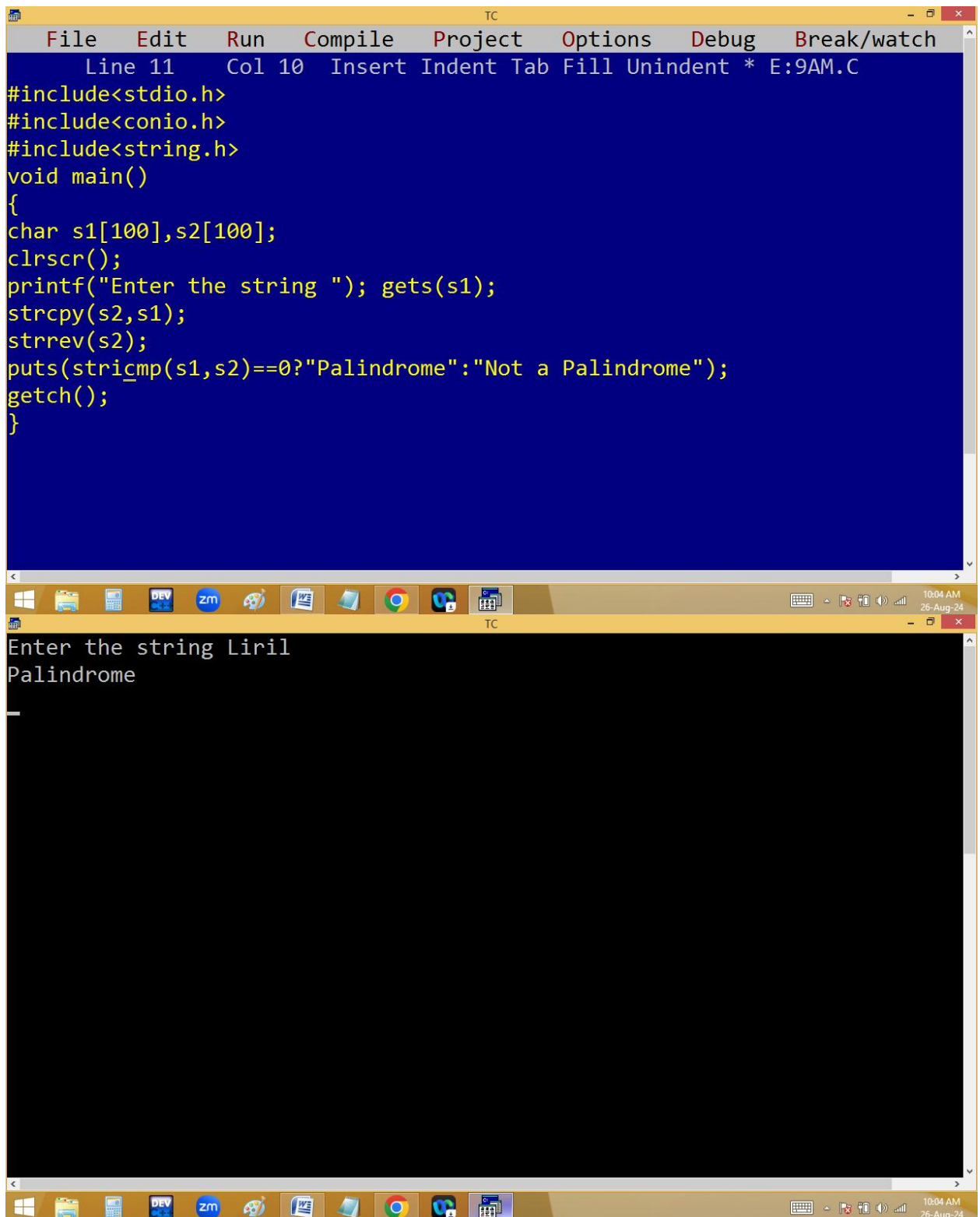
Syntax: strstr(main string, sub string);



```
File Edit Run Compile Project Options Debug Break/watch
Line 13 Col 40 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[]="abcdef";
clrscr();
printf("%s addr is %u\n",s,s);
printf("c addr is %u\n",strstr(s,"c"));
printf("C addr is %u\n",strstr(s,"C"));
printf("%s\n",strstr(s,"c"));
printf("%s\n",strstr(s,"C"));
printf("c is %d char",strstr(s,"c")-s+1);
getch();
}
```

abcdef addr is 65496
c addr is 65498
C addr is 0
cdef
(null)
c is 3 char_

Finding palindrome or not using predefined functions:

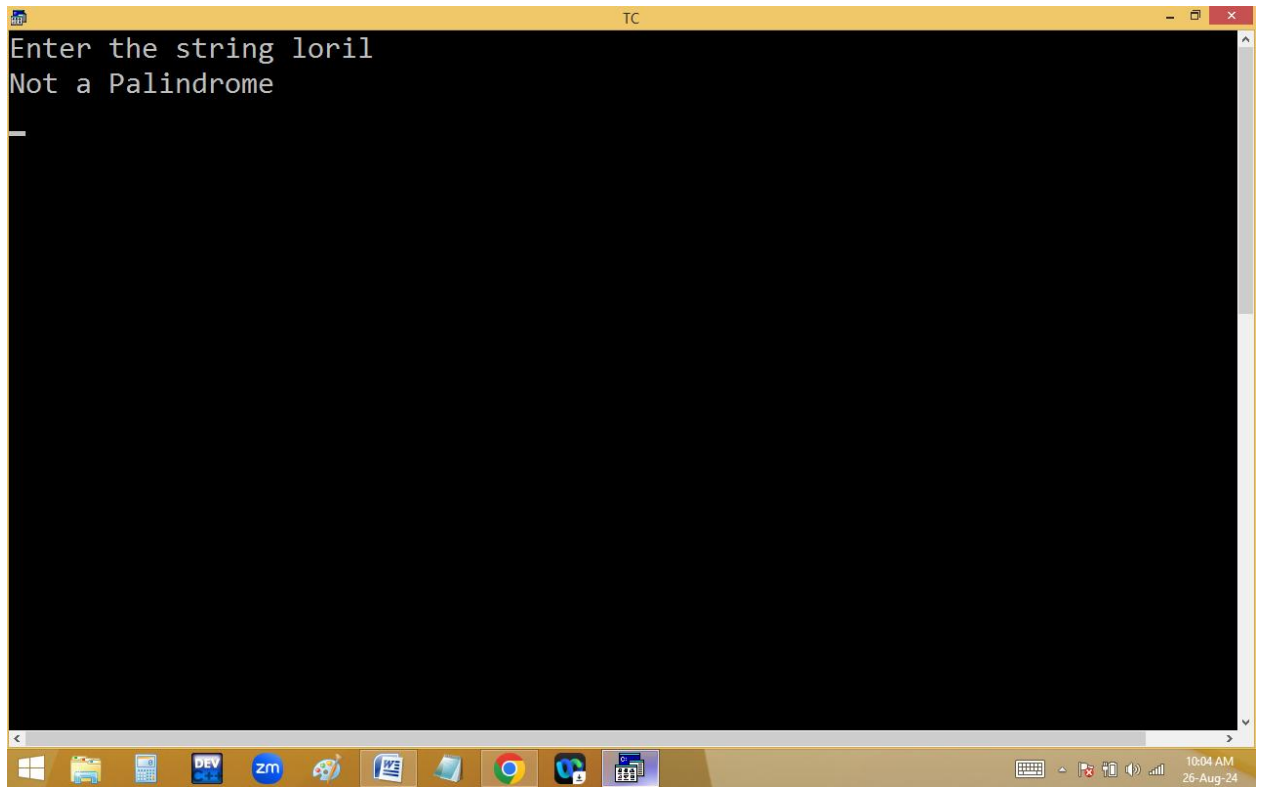


The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays the source code for a C program that checks if a string is a palindrome. The code includes headers for stdio.h, conio.h, and string.h. It defines a main function where two character arrays, s1 and s2, are declared and initialized to 100. The program prompts the user to enter a string, reads it into s1, copies it to s2, reverses s2 using strrev, and then compares s1 and s2 using strcmp. If they are equal, it prints "Palindrome"; otherwise, it prints "Not a Palindrome".

```
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 10 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s1[100],s2[100];
clrscr();
printf("Enter the string "); gets(s1);
strcpy(s2,s1);
strrev(s2);
puts(strcmp(s1,s2)==0?"Palindrome":"Not a Palindrome");
getch();
}
```

The bottom window shows the program's execution. It prompts "Enter the string" and the user has entered "Liril". The program then outputs "Palindrome".

```
Enter the string Liril
Palindrome
```



Sorting of strings:

The image shows a screenshot of a Turbo C++ (TC) IDE. The top window displays the source code of a program that searches for a string within an array of strings. The code includes headers for `stdio.h`, `conio.h`, and `string.h`. It defines a `main` function that declares an array `s1` of 5 strings, each up to 100 characters long, and a search string `s2`. It prompts the user to enter 5 strings and a search string, then prints the names and searches for the input string. The bottom window shows the program's execution. The user entered the strings "rohith sharma", "jack", "akash", "bablu", and "shradda kapoor". The search string entered was "sh". The program printed "NAMES" followed by a separator line, and then listed the strings "rohith sharma", "akash", and "shradda kapoor" which contain the search string.

```
File Edit Run Compile Project Options Debug Break/watch
Line 14 Col 1 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s1[5][100],s2[100]; int i;
clrscr();
printf("Enter 5 strngs\n");for(i=0;i<5;i++)gets(s1[i]);
printf("Enter search string "); gets(s2);
puts("NAMES");
puts("*****");
for(i=0;i<5;i++)
if(strstr(s1[i],s2)!=0)puts(s1[i]);
getch();
}
```

Enter 5 strngs
rohith sharma
jack
akash
bablu
shradda kapoor
Enter search string sh
NAMES

rohith sharma
akash
shradda kapoor

Sorting of strings:

s[0]	vijay gopi anu
s[1]	gopi vijay gopi
s[2]	anu gopi vijay vicky
s[3]	vicky vijay

selection sort

```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 2 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[7][100],t[100]; int i,j; clrscr();
printf("Enter 7 strngs\n");for(i=0;i<7;i++)gets(s[i]);
for(i=0;i<=5;i++)
{
for(j=i+1;j<=6;j++)
{if(stricmp(s[i],s[j])>0)
{strcpy(t,s[i]);strcpy(s[i],s[j]);strcpy(s[j],t);}
} }
puts("NAMES");
puts("*****");
for(i=0;i<7;i++)puts(s[i]);
getch();
}

Enter 7 strngs
Virat
gill
Sachin
dhoni
surya
hardik
Jadeja
NAMES
*****
dhoni
gill
hardik
Jadeja
Sachin
surya
Virat
_
```

POINTERS:

Pointer is a variable which stores the address of another variable of same type.

Pointer is a variable which stores the address of memory [bytes] at runtime [dynamic].

```
char name[20];
```