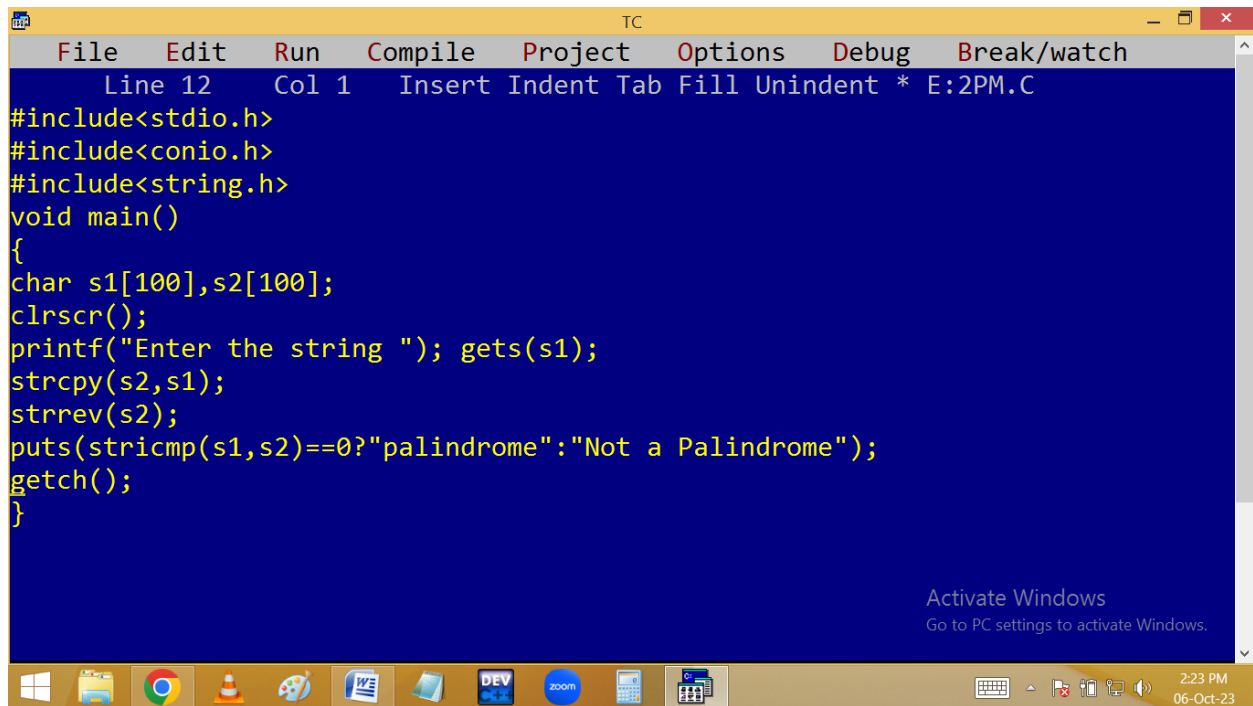


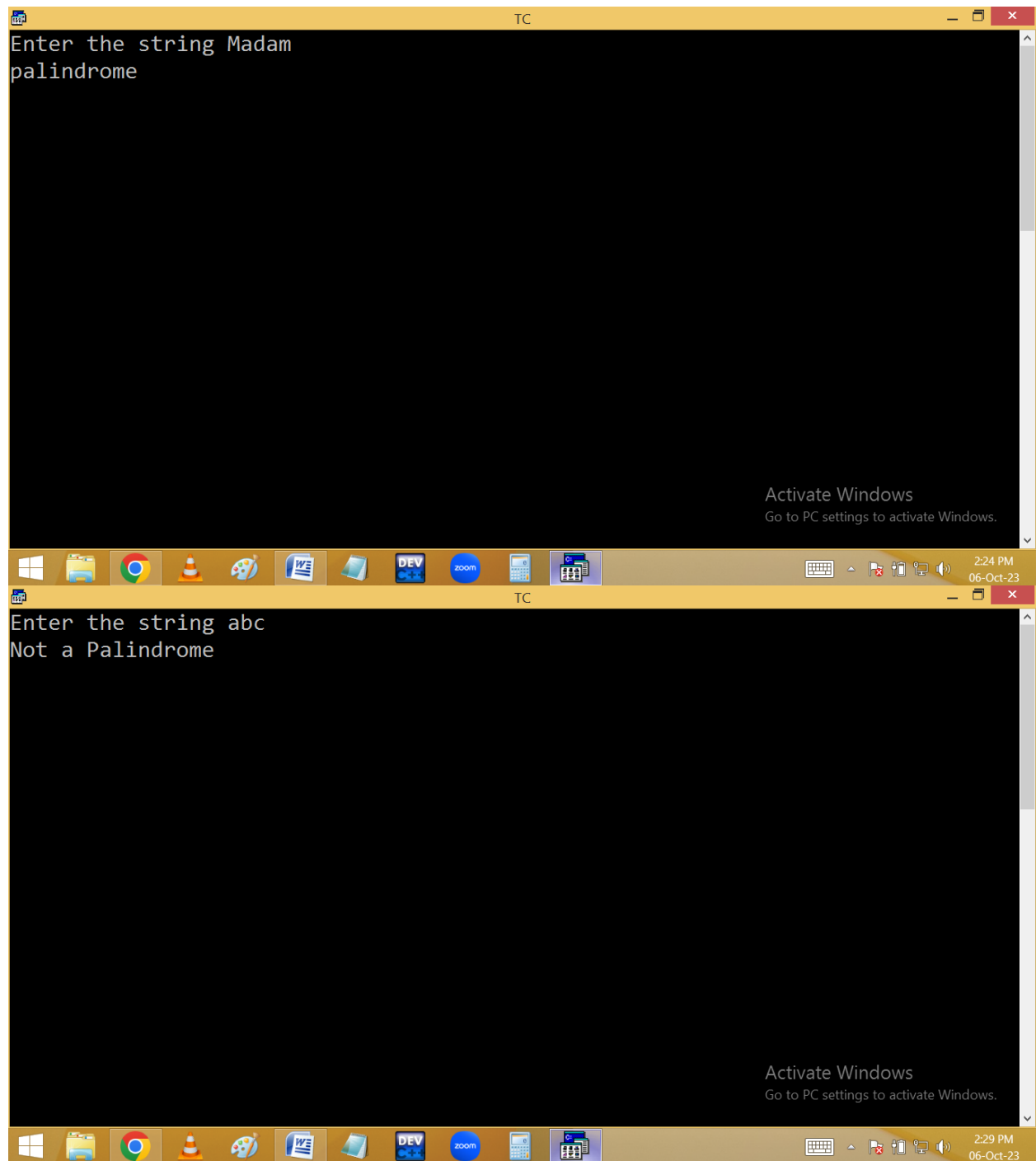
Finding palindrome using library functions:



The screenshot shows the Turbo C++ (TC) IDE interface. The menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. The status bar at the top indicates 'Line 12 Col 1' and 'Insert Indent Tab Fill Unindent * E:2PM.C'. The main editing area has a blue background and contains the following C code:

```
#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(stricmp(s1,s2)==0?"palindrome":"Not a Palindrome");
getch();
}
```

An 'Activate Windows' watermark is visible in the bottom right corner of the IDE window, with the text 'Go to PC settings to activate Windows.' The Windows taskbar at the bottom shows various application icons, including the Start button, File Explorer, Google Chrome, VLC media player, Paint, Word, and several development-related icons like DEV, Zoom, and a calculator. The system clock in the bottom right corner shows '2:23 PM' and '06-Oct-23'.



Sorting of strings:

s[0] ~~nitin~~ ~~gopi~~ aruna

s[1] ~~gopi~~ nitin ~~gopi~~ gopal

s[2] ~~aruna~~ ~~gopi~~ nitin gopi

s[3] ~~gopal~~ gopi nitin

```
TC
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[5][100],t[100];int i,j;
clrscr();
printf("Enter 5 names "); for(i=0;i<5;i++)gets(s[i]);
for(i=0;i<=3;i++)
{
for(j=i+1;j<=4;j++)
{if(strcmp(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<5;i++)puts(s[i]);
getch();
}

```

Page: 3 of 3 Words: 8 100% 2:42 PM 06-Oct-23

TC

```
Enter 5 names chinnu
pandu
MUNNA
bunny
PAPPU
NAMES
-----
bunny
chinnu
MUNNA
pandu
PAPPU

```

Activate Windows
Go to PC settings to activate Windows.

2:42 PM
06-Oct-23

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 [byte] at **runtime** [dynamic].

int a[10]={4, 9}; static memory[compile time]

derived datatype

advantages:

1. Dma – run time mem alloc
2. Mem wastage prevented.
3. Performance is high
4. System software development
5. Data structures
6. It allows to access local variable outside function. i.e. sharing data between functions [call by addr/reference]
7. File handling.

8. Strings / array controlled with pointers.

9. speed

disadvantage:

they are not secured.

Syntax:

```
Datatype * variable; /* declaration */
```

Here * indicates it is a pointer variable

Eg:

```
int * p; /* pointer var declaration */
```

```
int a=100;
```

```
p = &a;
```

to access the pointer we are using below

syntax:

```
*var;
```

Here * is called indirection op

Dereferencing op

Redirection op

Value at that address

Value of pointer