

Sorting of strings:

s[0]	s anju mi l ky chandu
s[1]	mi l ky s anju mi l ky jaanu
s[2]	cha n du mi l ky sa n ju milky
s[3]	ja a nu mi l ky sa n ju saajan
s[4]	sa a jan sanju

```
TC
Line 18 Col 8 Insert Indent Tab Fill Unindent * C:NONAME.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[5][40],temp[40]; int i,j; clrscr();
puts("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(stricmp(s[i],s[j])>0)
{ strcpy(temp,s[i]); strcpy(s[i],s[j]); strcpy(s[j],temp);}
}
}
puts("Names"); puts("-----");for(i=0;i<5;i++)puts(s[i]);
getch();
}

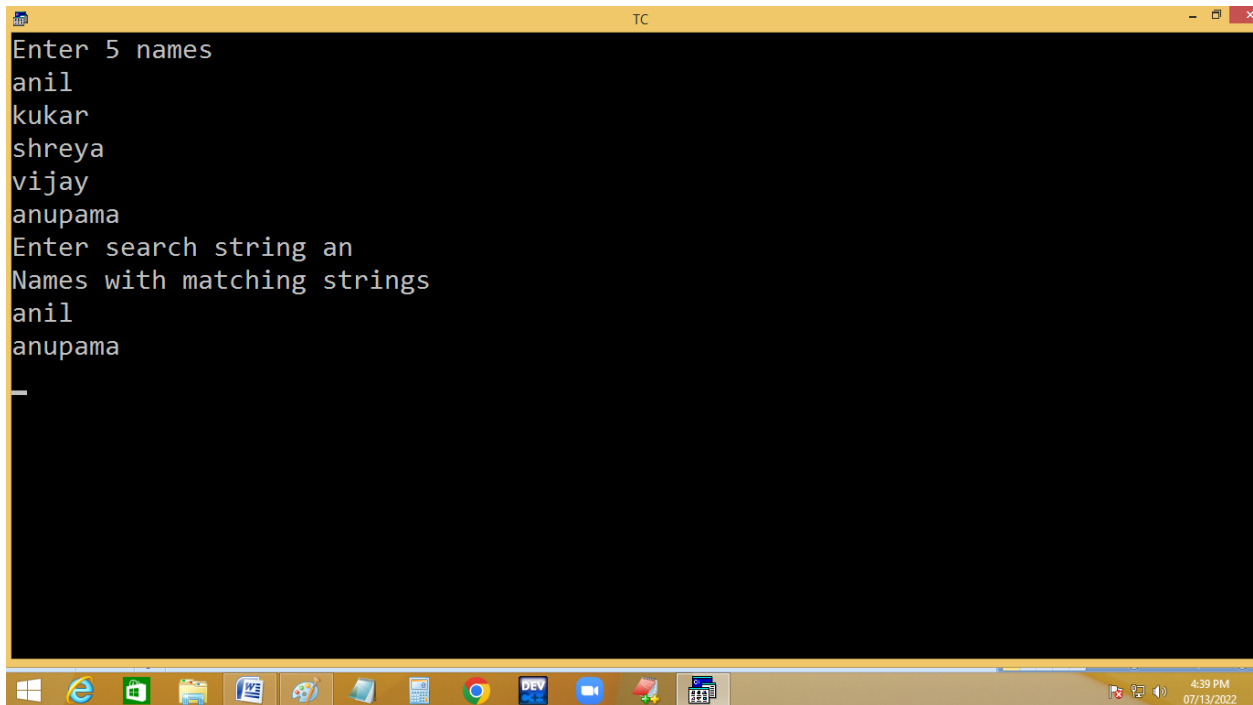
Enter 5 names
james
CHINNU
munni
CHOTU
banti
Names
-----
banti
CHINNU
CHOTU
james
munni
```

Searching for matching string:

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays the source code of a C program. The code includes headers for `stdio.h`, `conio.h`, and `string.h`. The `main` function prompts the user to enter 5 names and a search string. It then iterates through the names and prints those that contain the search string using the `strstr` function. The bottom window shows the program's execution output, where the user has entered the names 'janaki', 'sarayu', 'amith', 'vishal', and 'santhosh', and the search string 'u'. The output shows 'Names with matching strings' followed by 'sarayu'.

```
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 37 Indent Tab Fill Unindent * C:NONAME.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[5][40],temp[40]; int i,j; clrscr();
puts("Enter 5 names "); for(i=0;i<5;i++)gets(s[i]);
printf("Enter search string "); gets(temp);
puts("Names with matching strings");
for(i=0;i<5;i++)
{
if(strstr(s[i],temp)!=0)puts(s[i]);
}
getch();
}
```

Enter 5 names
janaki
sarayu
amith
vishal
santhosh
Enter search string u
Names with matching strings
sarayu



```
TC
Enter 5 names
anil
kukar
shreya
vijay
anupama
Enter search string an
Names with matching strings
anil
anupama
_
```

POINTERS

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

Pointer is a memory location, which holds the address of another memory location.

Pointer is a derived data type.

Advantages:

1. Dynamic memory allocation.

2. Program performance is increased due to preventing memory wastage.
3. They are very much used in System programming.
4. They are very much used in dynamic linked list & Stacks [data structures].
5. It allows to access a local variable outside the function i.e. data sharing between functions. [call by address].
6. To handle strings, arrays etc in functions we need pointers.
7. To handle data files we are using pointers.
8. They directly works on variable address. Due to this search time is reduced and execution speed is increased.

Disadvantage:

They are not secured and make programmer complex.

