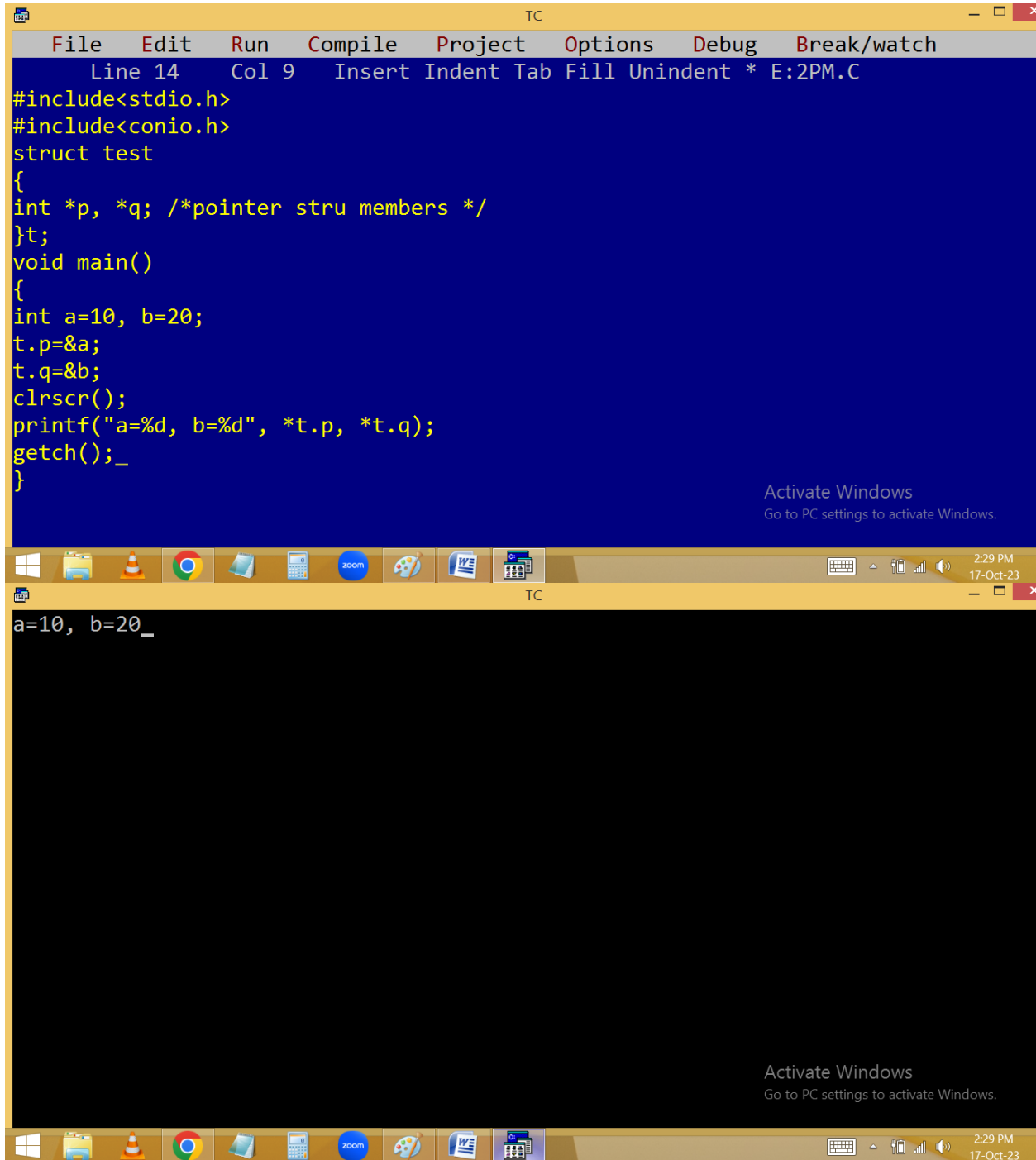


Pointer structure members:



The screenshot displays the Turbo C++ (TC) IDE interface. The top window shows the source code for a C program named E:2PM.C. The code defines a structure named 'test' with two pointer members, 'p' and 'q', and a 'main' function that initializes these pointers to point to variables 'a' and 'b' respectively. The bottom window shows the output of the program, which is 'a=10, b=20_'. The Windows taskbar at the bottom indicates the time is 2:29 PM on 17-Oct-23.

```
File Edit Run Compile Project Options Debug Break/watch
Line 14 Col 9 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
struct test
{
int *p, *q; /*pointer stru members */
}t;
void main()
{
int a=10, b=20;
t.p=&a;
t.q=&b;
clrscr();
printf("a=%d, b=%d", *t.p, *t.q);
getch();_
}
```

Activate Windows
Go to PC settings to activate Windows.

a=10, b=20_

Activate Windows
Go to PC settings to activate Windows.

Pointer to structure / pointer structure variables:

The image shows two screenshots of the Turbo C++ (TC) IDE. The top screenshot displays the source code of a C program. The bottom screenshot shows the program's execution output.

Top Screenshot (Source Code):

```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 5 Col 25 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
struct book
{
char title[30], author[30];float price;
}b, *p; /* pointer stru var */
void main()
{
p=&b;
clrscr();
printf("Enter book title "); gets(b.title);
printf("Enter author name "); gets(b.author);
printf("Enter price "); scanf("%f",&b.price);
printf("%s author %s and price=%.2f",b.title, b.author, b.price);
getch();
}
```

Bottom Screenshot (Execution Output):

```
TC
Enter book title Programming In ANSI-C
Enter author name Bala Guru samy
Enter price 350
Programming In ANSI-C author Bala Guru samy and price=350.00_
```

Both screenshots include a Windows taskbar at the bottom with icons for File Explorer, VLC, Chrome, and other applications. The system clock in the bottom right corner shows 2:34 PM and 2:35 PM on 17-Oct-23.

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays the source code for a program that uses an array of structures. The code includes `<stdio.h>` and `<conio.h>`. It defines a structure named `book` with fields `title`, `author`, and `price`. In the `main` function, a pointer `p` is assigned to the first element of the `book` array, and user input is used to populate the fields. The output window shows the program's execution, where the user enters 'Let Us C' for the title, 'Yashwant Kanitkar' for the author, and '400' for the price. The final output is 'Let Us C author Yashwant Kanitkar and price=400.00_'. The Windows taskbar at the bottom shows the date as 17-Oct-23 and the time as 2:39 PM.

```
File Edit Run Compile Project Options Debug Break/watch
Line 3 Col 36 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void dummy(float a) { float *p=&a;}
struct book
{
char title[30], author[30];float price;
}b, *p; /* pointer stru var */
void main()
{
p=&b;
clrscr();
printf("Enter book title "); gets((*p).title);
printf("Enter author name "); gets((*p).author);
printf("Enter price "); scanf("%f",&(*p).price);
printf("%s author %s and price=%.2f",p->title, p->author, p->price);
getch();
}
```

Enter book title Let Us C
Enter author name Yashwant Kanitkar
Enter price 400
Let Us C author Yashwant Kanitkar and price=400.00_

Array of structures:

```
#include<stdio.h>
```

```
#include<conio.h>

void dummy(float a) { float *p=&a;}

struct bill

{

char item[30];float qty, price;

}b[100]; /* array of structure */

void main()

{

int i=0, c=1; char ch; float amt,tot=0,gst, net;

clrscr();

while(1)

{

flushall();

printf("Enter Item Name "); gets(b[i].item);

printf("Enter Quantity ");scanf("%f",&b[i].qty);

printf("Enter price "); scanf("%f",&b[i].price);
```

```

flushall();

printf("Do you want to continue [y/n] ");
scanf("%c",&ch);

if(ch=='n' || ch=='N') break; else {i++;c++;}

}

puts("*****
*****");

puts("\t\t D-MART");

puts("\t\t SANATH NAGAR");

puts("*****
*****");

printf("%-
20s%10s%10s%10s\n","ITEM","QTY","PRICE","AMO
UNT");

puts("*****
*****");

for(i=0;i<c;i++)

```

```
{  
  
amt=b[i].qty*b[i].price; tot+=amt;  
  
printf("%-20s%10.2f%10.2f%10.2f\n",b[i].item,  
b[i].qty, b[i].price, amt);  
  
}  
  
gst=tot*5/100;  
  
net=tot+gst;  
  
puts("*****  
*****");  
  
printf("%40s%10.2f\n","Total: ",tot);  
  
printf("%40s%10.2f\n","Gst [5%]: ",gst);  
  
printf("%40s%10.2f\n","Net: ",net);  
  
puts("*****Thank you - Visit  
Again*****");  
  
getch();  
  
}
```

```
TC
Enter Item Name Mysore Sandal
Enter Quantity 2
Enter price 235
Do you want to continue [y/n] y
Enter Item Name Rice Bag
Enter Quantity 1
Enter price 1750
Do you want to continue [y/n] y
Enter Item Name Oil
Enter Quantity 5
Enter price 200
Do you want to continue [y/n] n
*****
                        D-MART
                        SANATH NAGAR
*****
ITEM                QTY    PRICE    AMOUNT
*****
Mysore Sandal        2.00    235.00    470.00
Rice Bag             1.00    1750.00   1750.00
Oil                  5.00     200.00   1000.00
*****
                        Total:    3220.00
                        Gst [5%]:   161.00
                        Net:      3381.00
*****Thank you - Visit Again*****

```

Activate Windows
Go to PC settings to activate Windows.

Activate Windows
Go to PC settings to activate Windows.

3:16 PM
17-Oct-23

D-MART SANATH NAGAR				
	Item	Qty	Price	Amount
b[0]	M.S	2	235	470
b[1]	R.bag	1	1750	1750
b[2]	oil	5	200	1000
total:				3220
GST [5%]:				161
net:				3381

Dynamic structures:

Allocating memory to structure variable at runtime.
It prevents memory wastage.

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<alloc.h>
```

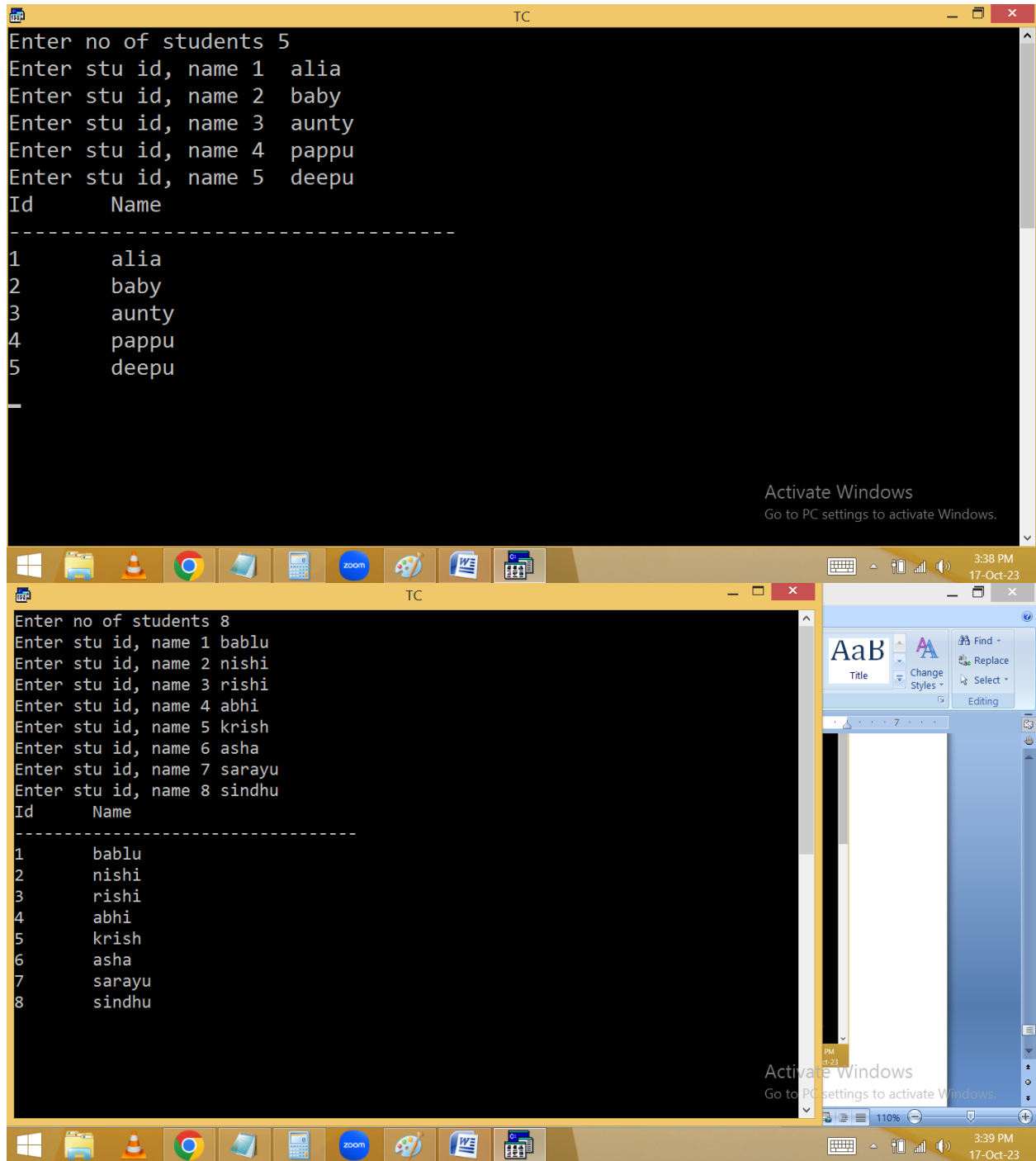
```
void dummy(float a) { float *p=&a;}
```

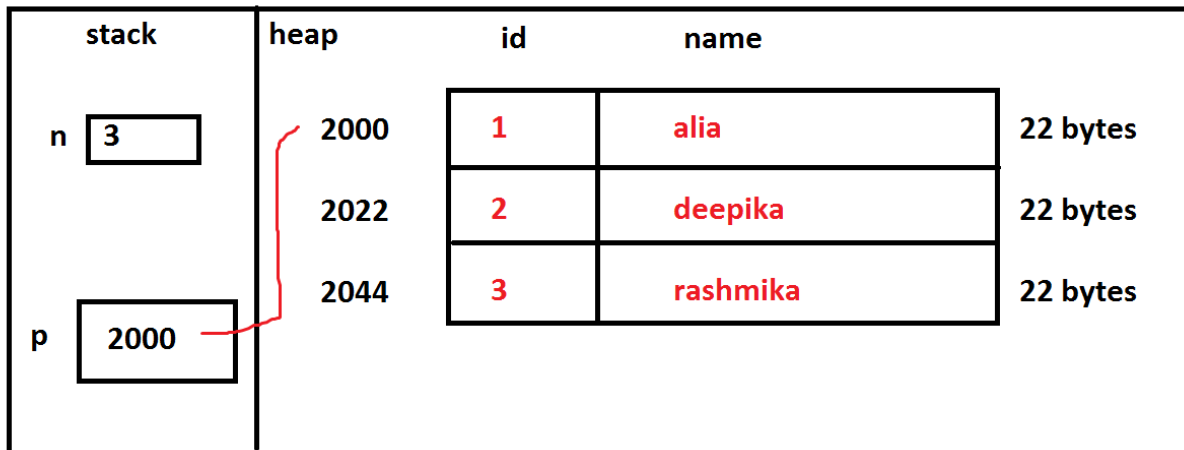
```
struct stu
```



```
{  
int id;  
char name[20];  
}s, *p;  
void main()  
{  
int n,i;  
clrscr();  
printf("Enter no of students "); scanf("%d",&n);  
p = ( struct stu * )calloc(n,sizeof(s));  
for(i=0;i<n;i++)  
{  
printf("Enter stu id, name ");  
scanf("%d %s", &(p+i)->id,(p+i)->name);  
}  
puts("Id\tName");
```

```
puts("-----");  
for(i=0;i<n;i++) printf("%d\t%s\n",(p+i)->id, (p+i)-  
>name);  
free(p); p=NULL; getch();  
}
```





```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<alloc.h>
```

```
void dummy(float a) { float *p=&a;}
```

```
struct stu
```

```
{
```

```
int id;
```

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

```
}s, *p;
```

```
void main()
```

```
{
```

```
int i=0,c=1;char ch;

clrscr();

p = ( struct stu * )calloc(c,sizeof(s));

while(1)

{

printf("Enter stu id, name ");

scanf("%d %s", &(p+i)->id,(p+i)->name);

printf("Next stu [y/n] "); fflush(); scanf("%c",&ch);

if(ch=='y' || ch=='Y')

{

i++; c++; p = (struct stu *)realloc(p, c*sizeof(s));

}

else break;

}


puts("Id\tName");
```

```

puts("-----");

for(i=0;i<c;i++) printf("%d\t%s\n",(p+i)->id, (p+i)-
>name);

free(p); p=NULL; getch();

}

```

The screenshot shows a Turbo C++ (TC) window with the following text:

```

Enter stu id, name 1 vimal
Next stu [y/n] y
Enter stu id, name 2 kamal
Next stu [y/n] y
Enter stu id, name 3 akmal
Next stu [y/n] n
Id      Name
-----
1       vimal
2       kamal
3       akmal

```

The window title is "TC". The status bar at the bottom indicates "Page: 14 of 14" and "Words: 253". The taskbar shows various icons including Windows, File Explorer, VLC, Chrome, and others. The system clock shows "3:46 PM" and "17-Oct-23".

OPERATOR	NAME	GROUPING
()	function call	left-to-right
[]	array element	
.	structure, union member	
->	structure, union member with pointer	
!	logical not	right-to-left
~	one's complement	

-	minus	
++	increment	
--	decrement	
&	address	
*	indirection	
(type)	type cast	
sizeof	size in bytes	
*	multiplication	left-to-right
/	division	
%	remainder	
+	addition	left-to-right
-	subtraction	
<<	shift left	left-to-right
>>	shift right	
<	less than	left-to-right
<=	less than or equal	
>	greater than	
>=	greater than or equal	
==	equal	left-to-right
!=	not equal	
&	bitwise and	left-to-right
^	bitwise exclusive or	left-to-right
	bitwise or	left-to-right
&&	logical and	left-to-right
	logical or	left-to-right
?:	conditional	right-to-left

=	assignment operator	right-to-left
+=	assignment replace add	
-=	assignment replace subtract	
*=	assignment replace multiply	
/=	assignment replace divide	
%=	assignment replace remainder	
<<=	assignment replace shift left	
>>=	assignment replace shift right	
&=	assignment replace and	
^=	assignment replace exclusive or	
=	assignment replace or	
,	comma	left-to-right