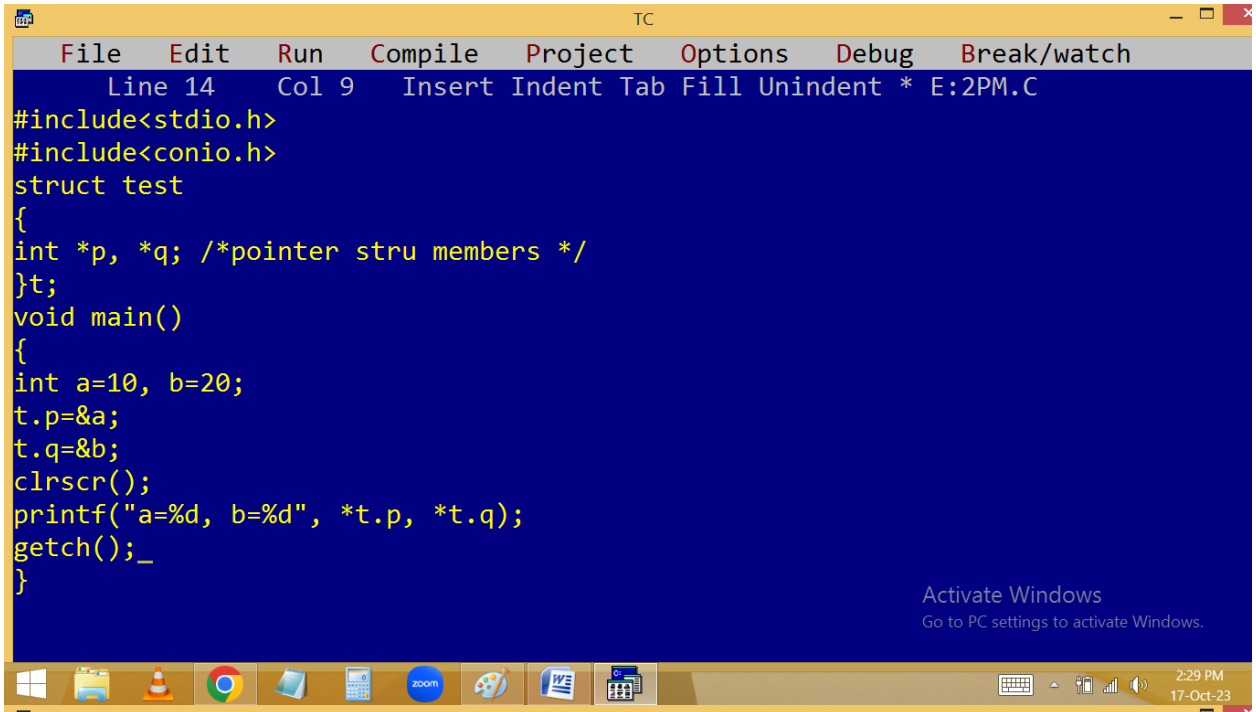
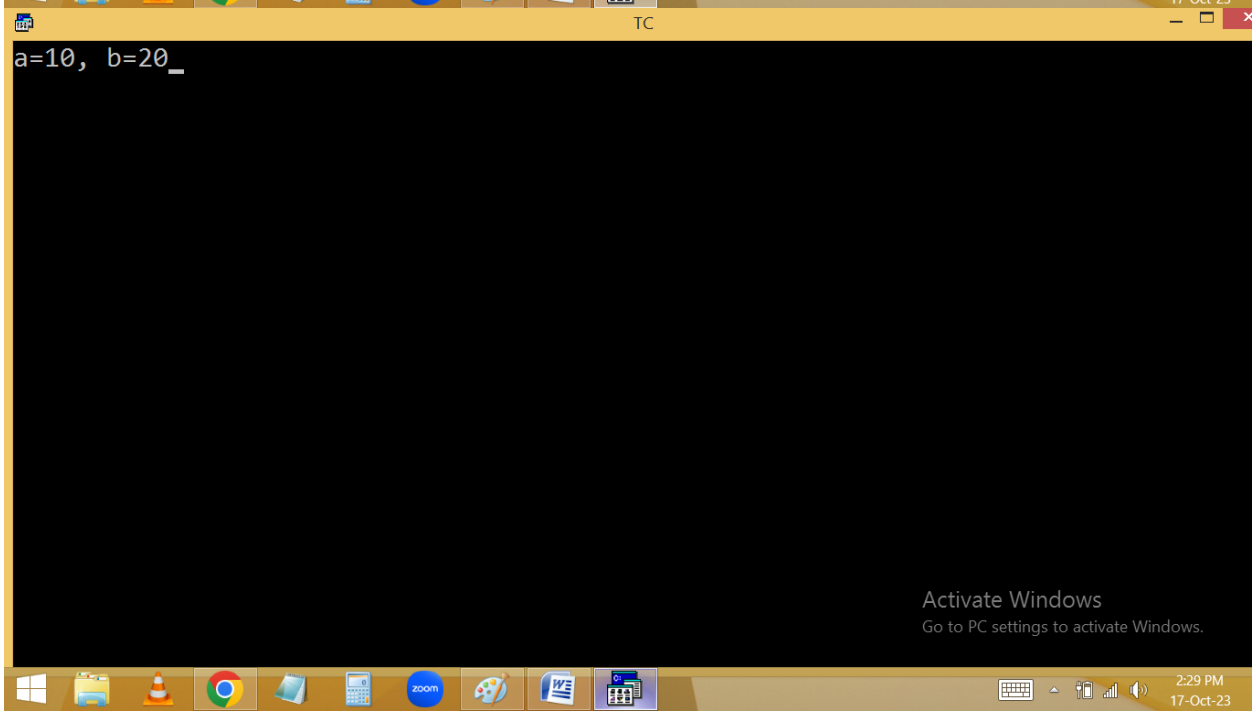
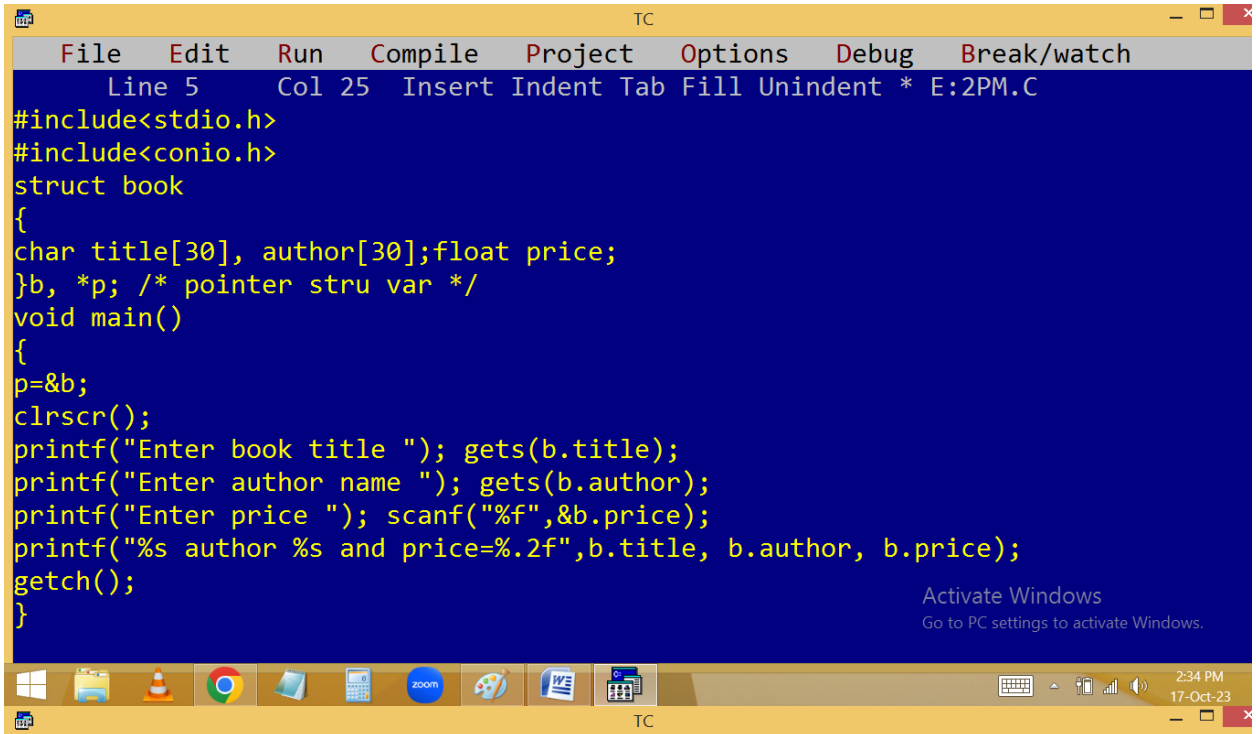
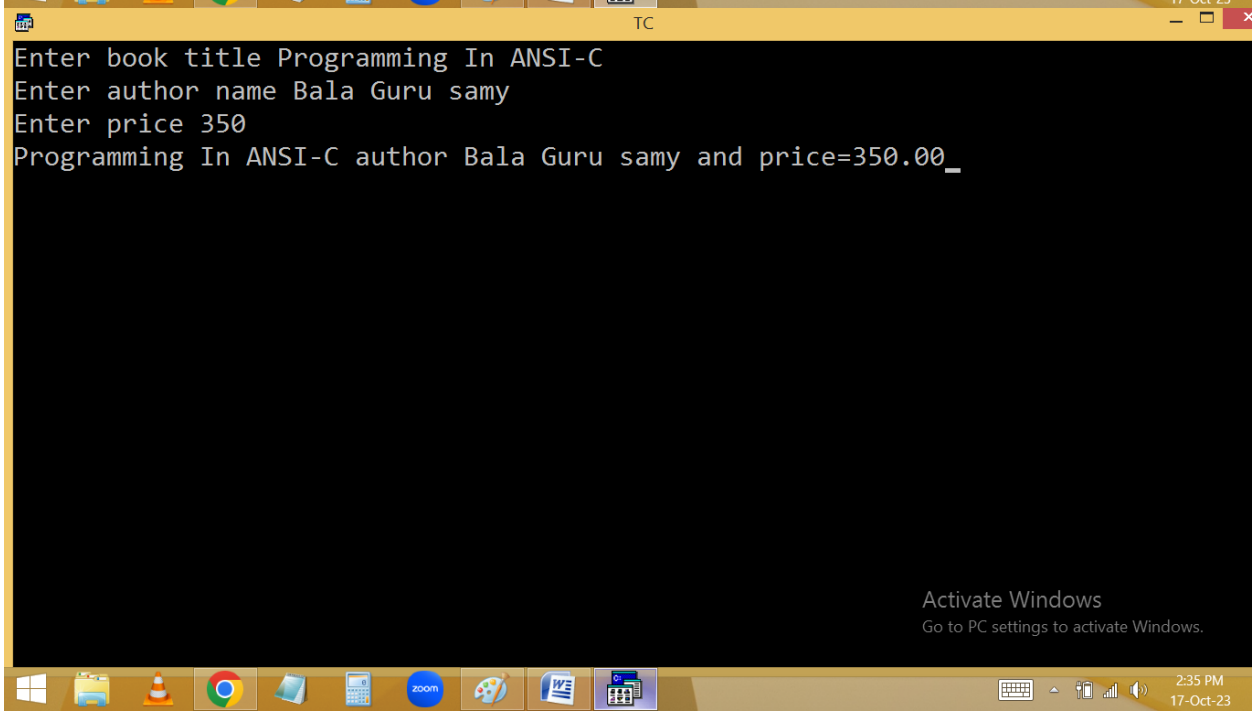
Pointer structure members:

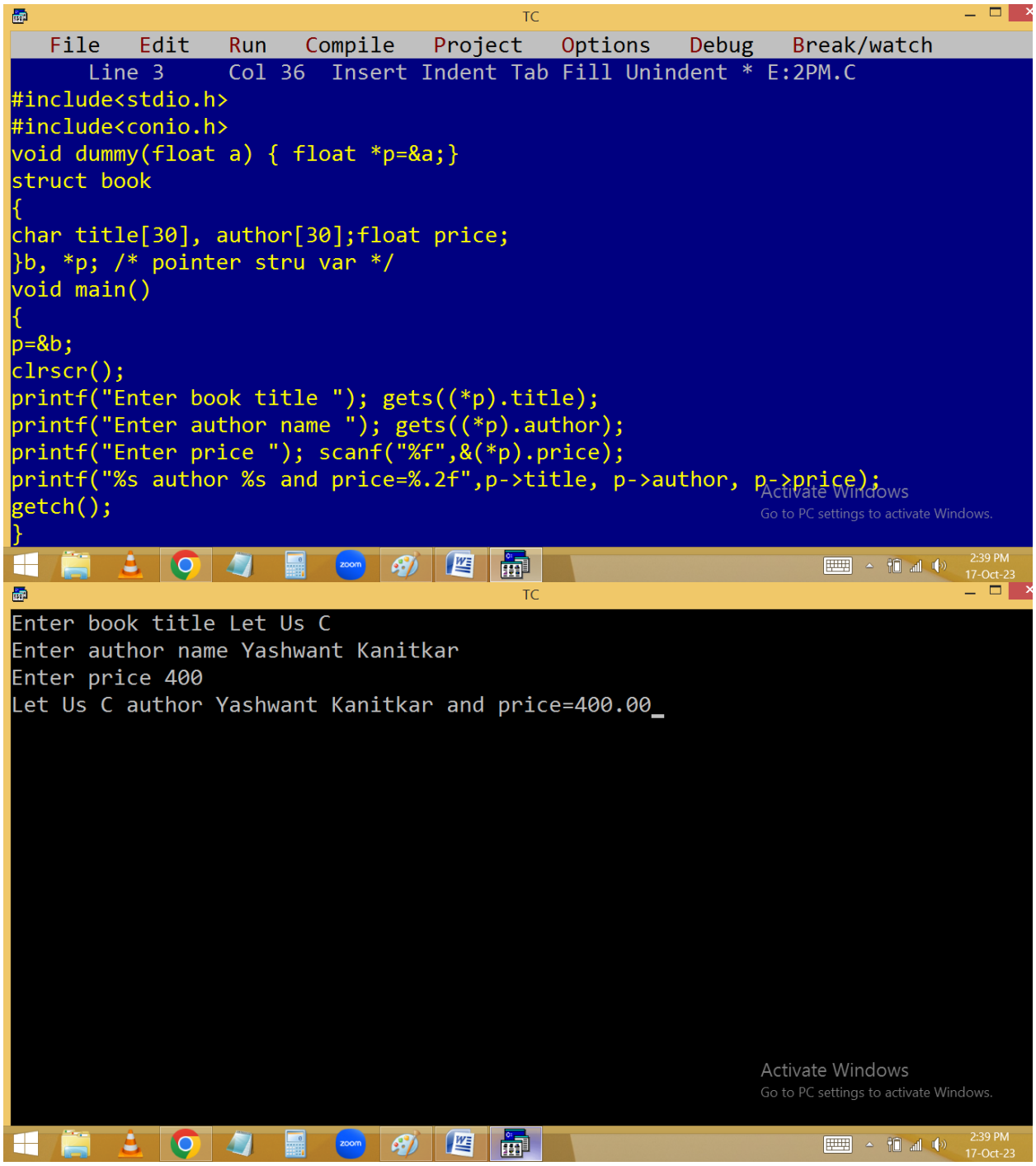




Pointer to structure / pointer structure variables:


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();
}
Activate Windows
Go to PC settings to activate Windows.
2:34 PM
17-Oct-23


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
Activate Windows
Go to PC settings to activate Windows.
2:35 PM
17-Oct-23



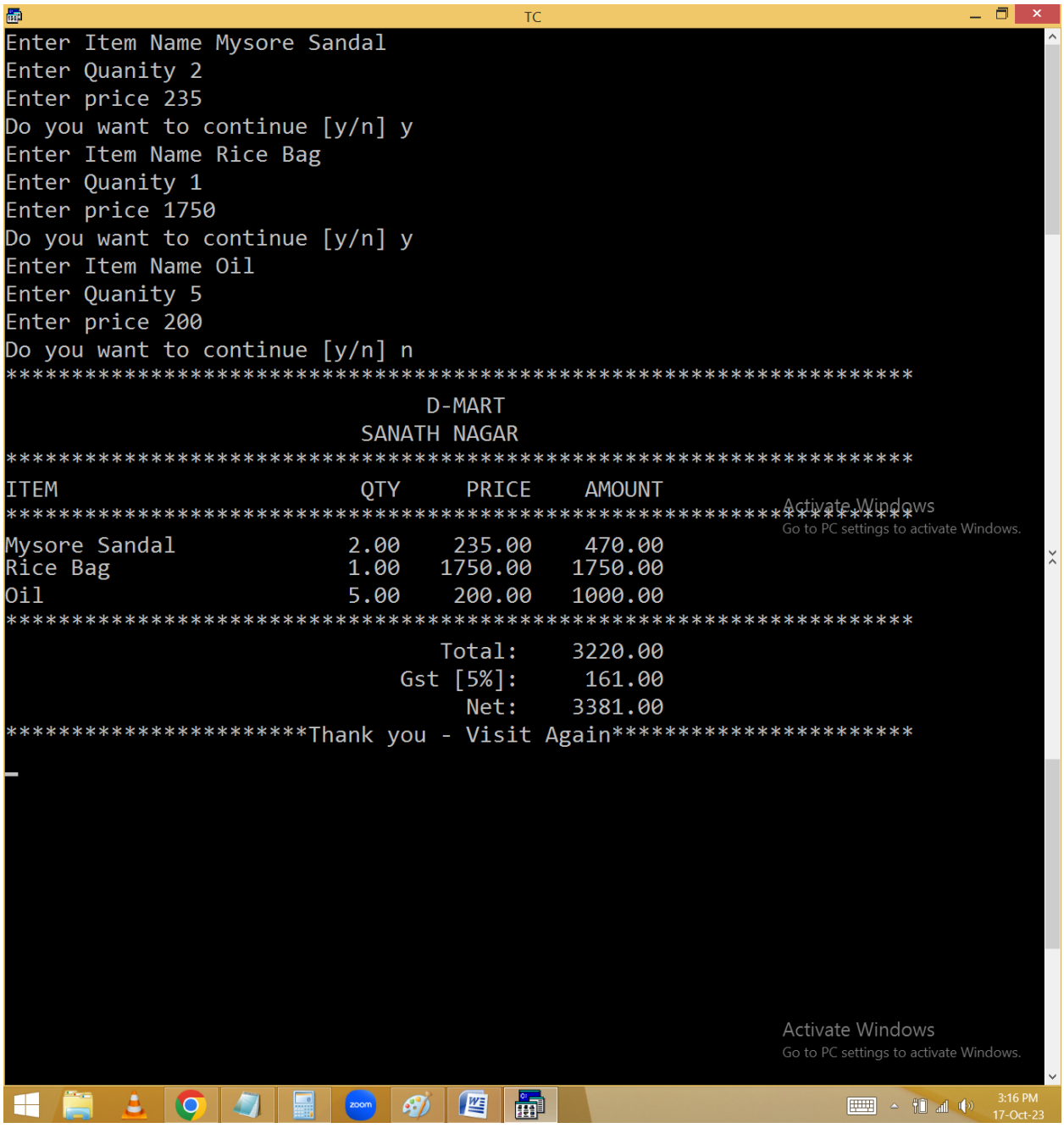
Array of structures:

#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 Quanity ");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\t D-MART");
puts("\t\t\t   SANATH NAGAR");
puts("**********************************************************************");
printf("%-20s%10s%10s%10s\n","ITEM","QTY","PRICE","AMOUNT");
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 Quanity 2
Enter price 235
Do you want to continue [y/n] y
Enter Item Name Rice Bag
Enter Quanity 1
Enter price 1750
Do you want to continue [y/n] y
Enter Item Name Oil
Enter Quanity 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:

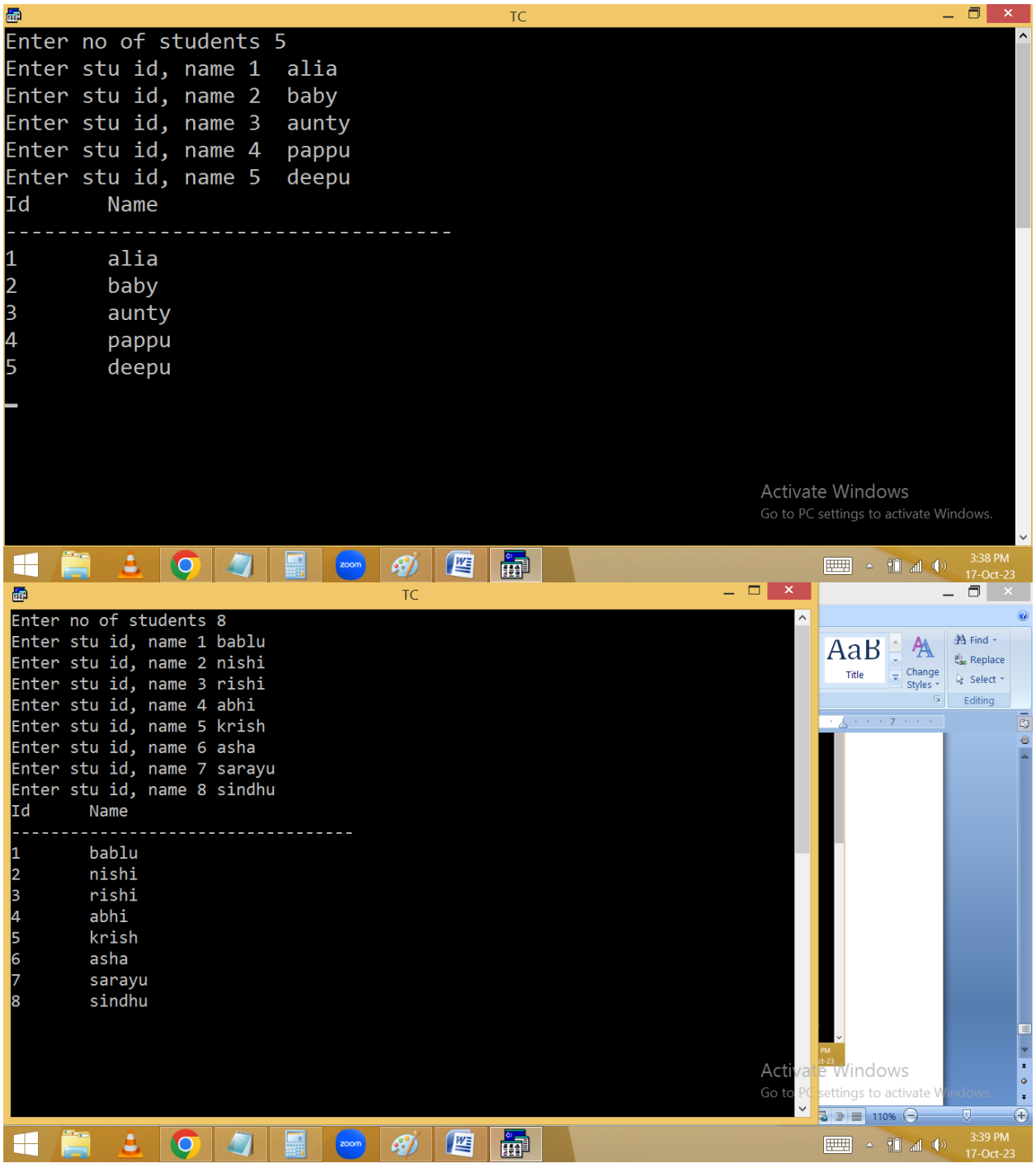
Allocating memory to structure variable at runtime.

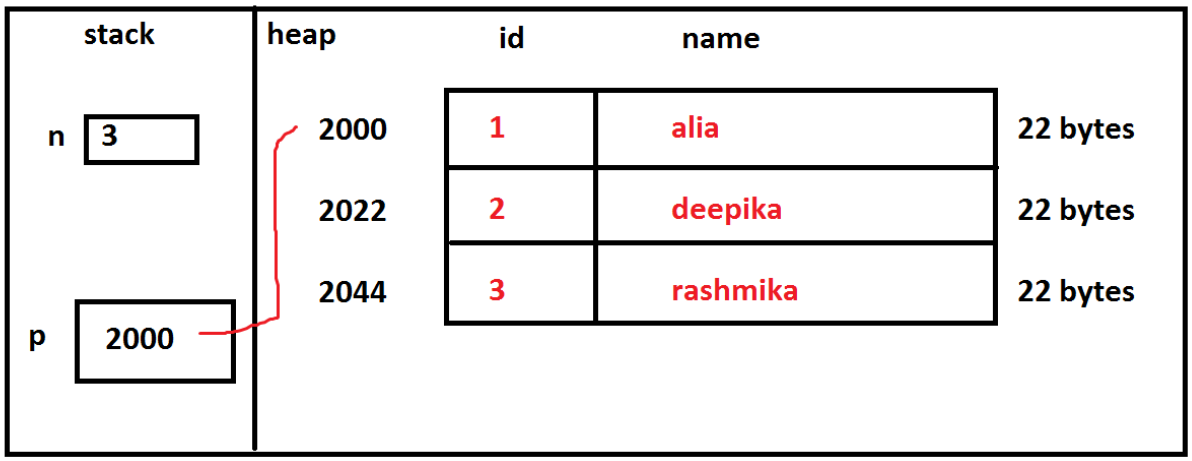
It prevents memory wastage.

```
#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();
}
```

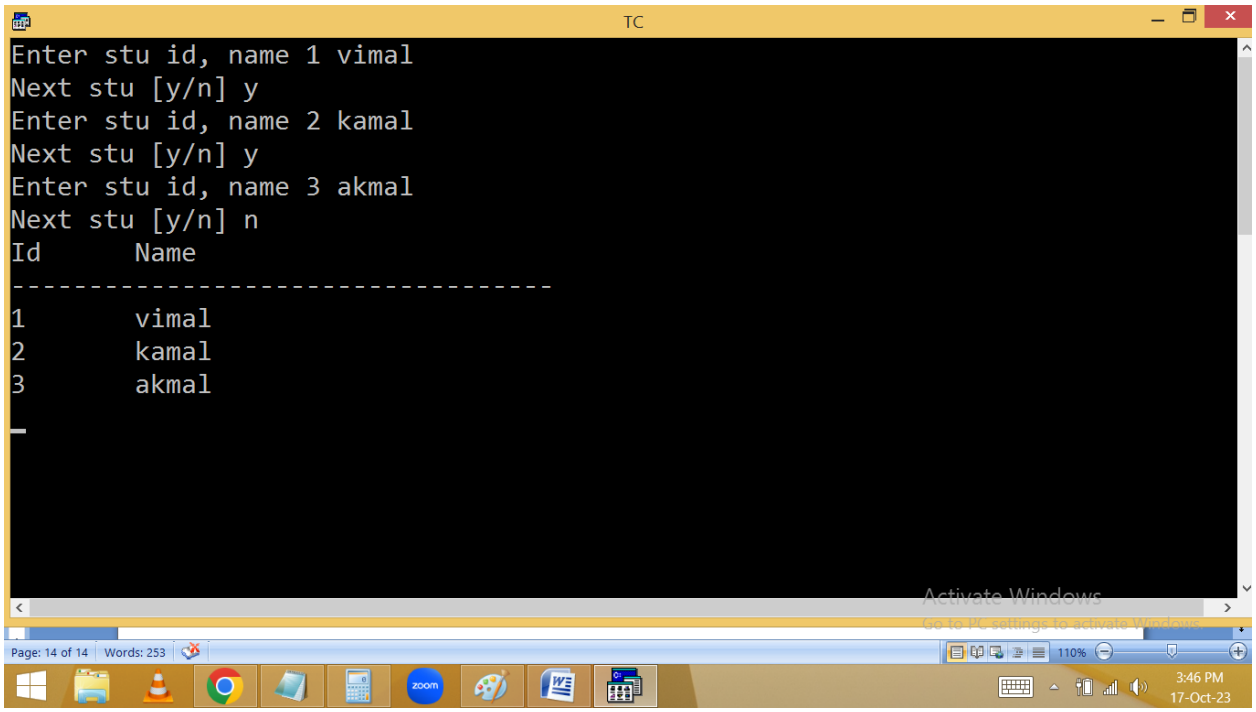

TC
Enter no of students 5
Enter stu id, name 1 alia
Enter stu id, name 2 baby
Enter stu id, name 3 aunty
Enter stu id, name 4 pappu
Enter stu id, name 5 deepu
Id Name
1 alia
2 baby
3 aunty
4 pappu
5 deepu
Activate Windows
Go to PC settings to activate Windows.
3:38 PM
17-Oct-23
TC
Enter no of students 8
Enter stu id, name 1 bablu
Enter stu id, name 2 nishi
Enter stu id, name 3 rishi
Enter stu id, name 4 abhi
Enter stu id, name 5 krish
Enter stu id, name 6 asha
Enter stu id, name 7 sarayu
Enter stu id, name 8 sindhu
Id Name
1 bablu
2 nishi
3 rishi
4 abhi
5 krish
6 asha
7 sarayu
8 sindhu
Find
Replace
Select
Editing
Change Styles
Title
Activate Windows
Go to PC settings to activate Windows.
110%
3:39 PM
17-Oct-23




```
#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] "); flushall(); 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();
}
```



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