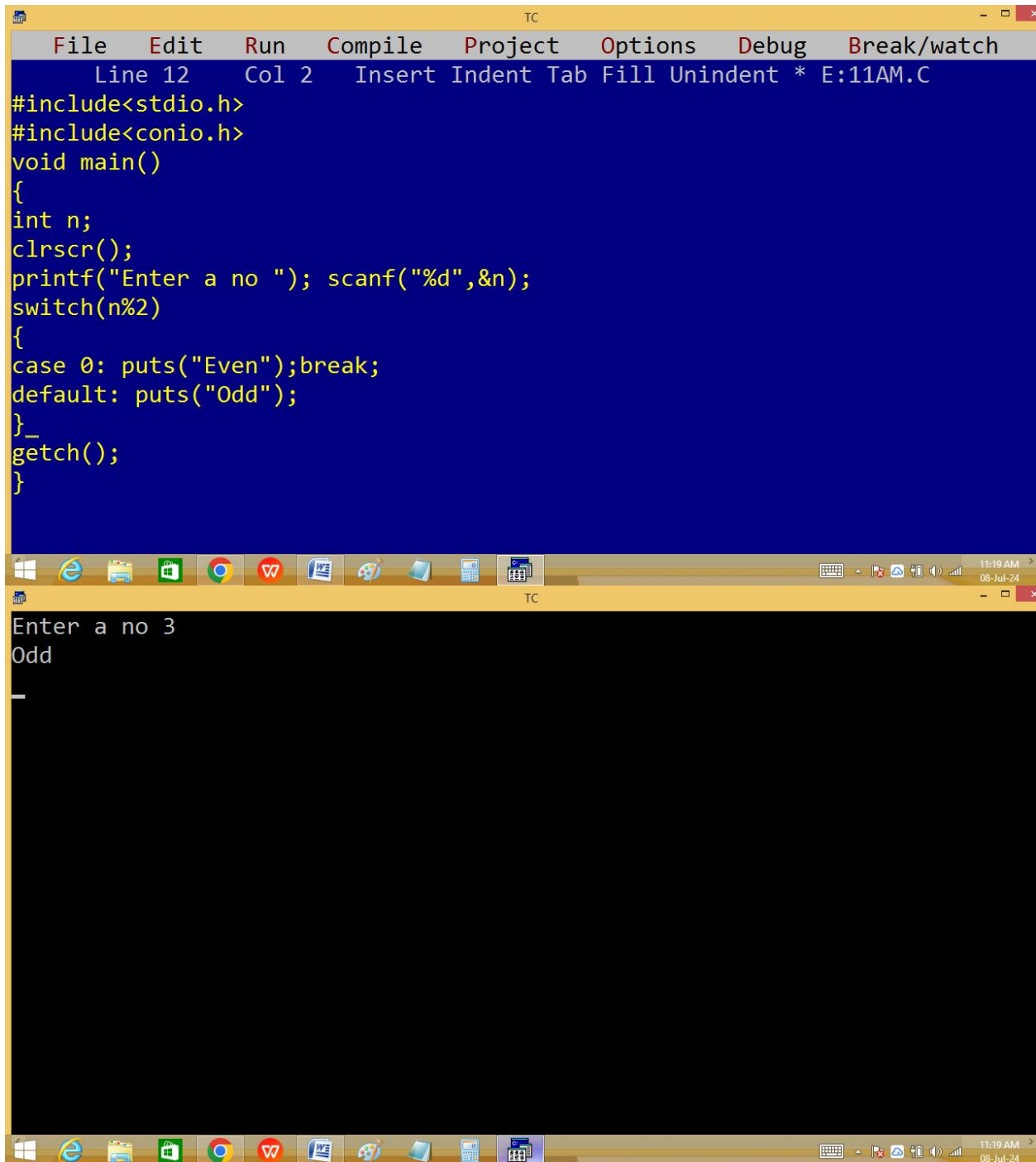


Finding even/odd using switch:



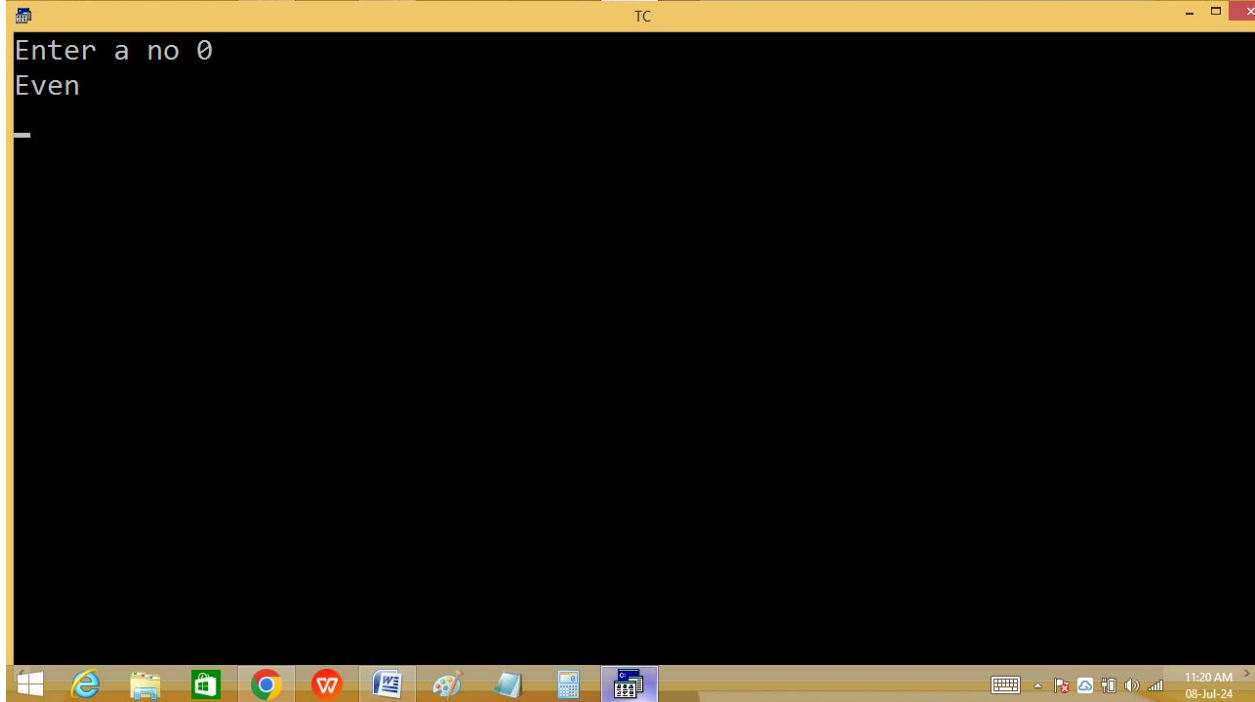
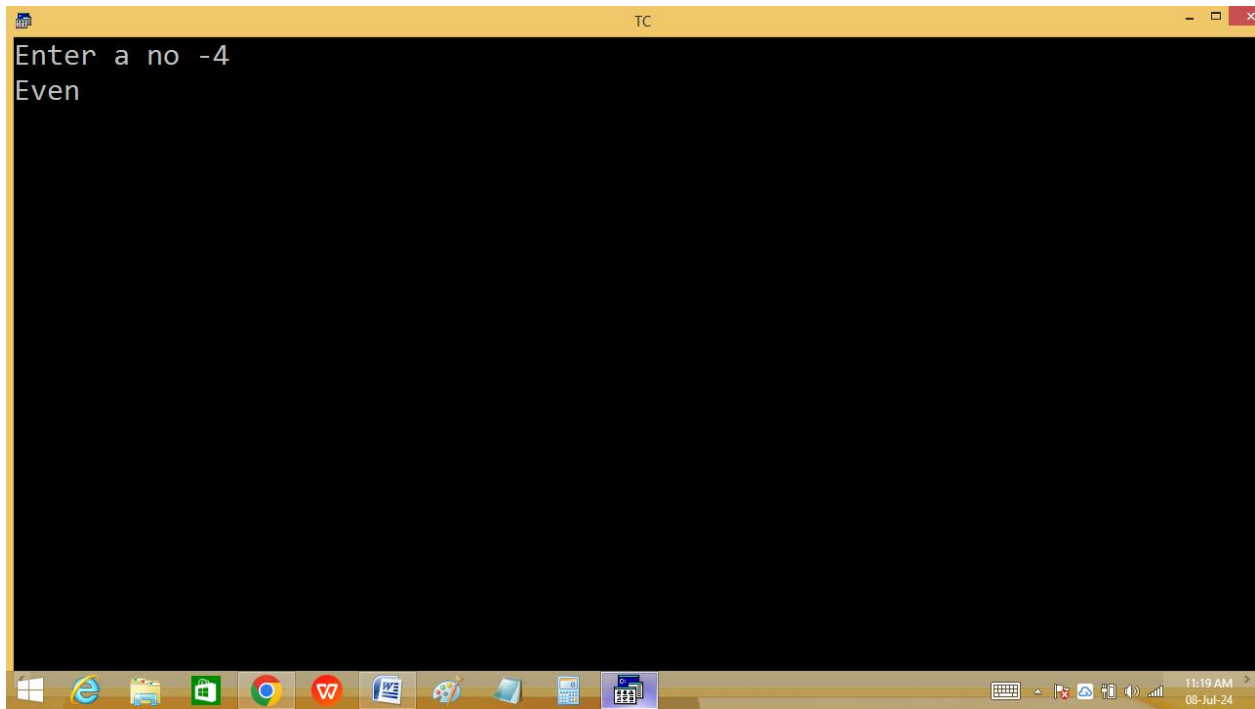
```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 12 Col 2 Insert Indent Tab Fill Unindent * E:11AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n;
clrscr();
printf("Enter a no "); scanf("%d",&n);
switch(n%2)
{
case 0: puts("Even");break;
default: puts("Odd");
}
getch();
}
```

Enter a no 3
Odd

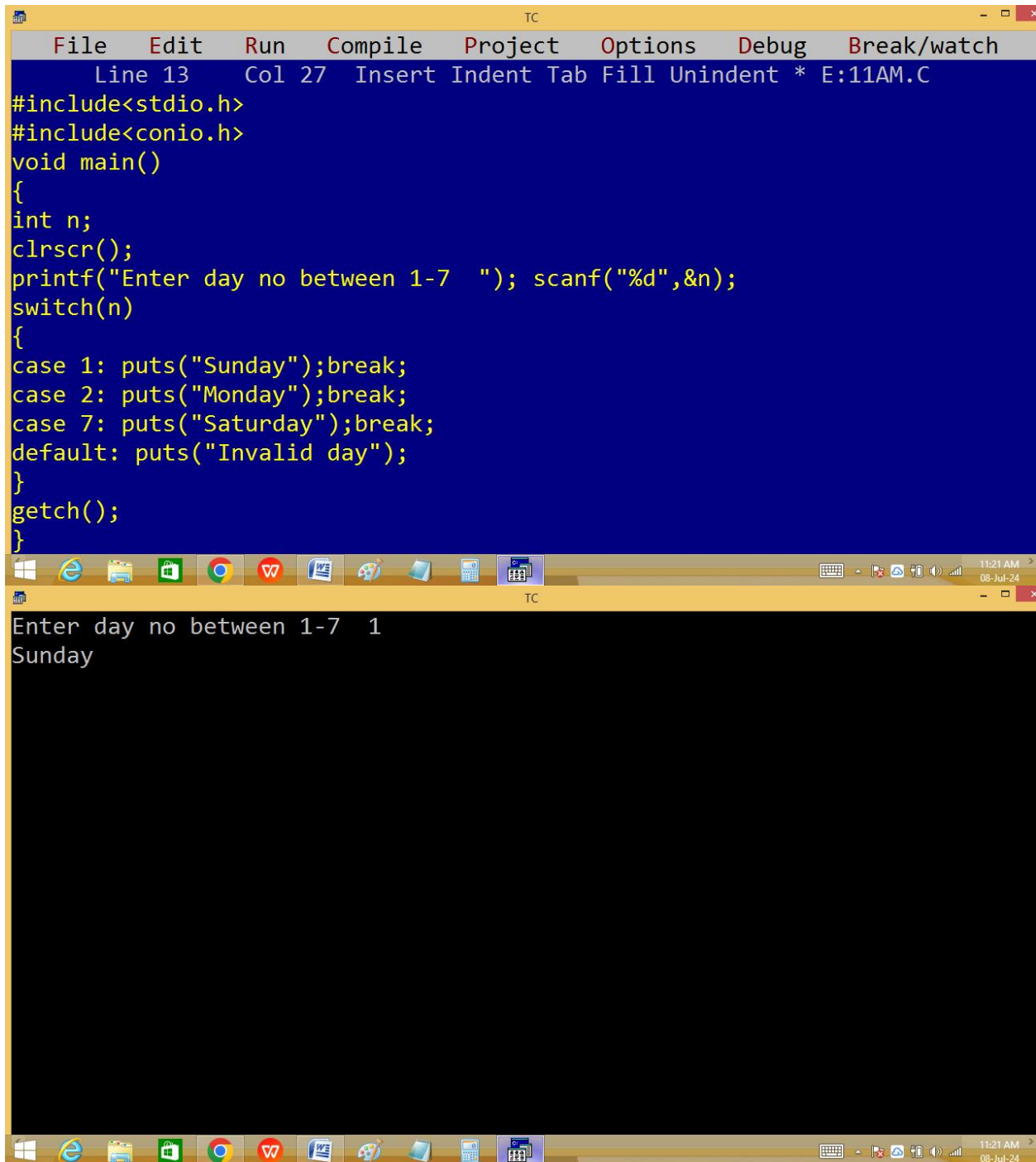
```
TC
Enter a no 4
Even
_
```



```
TC
Enter a no -3
Odd
```

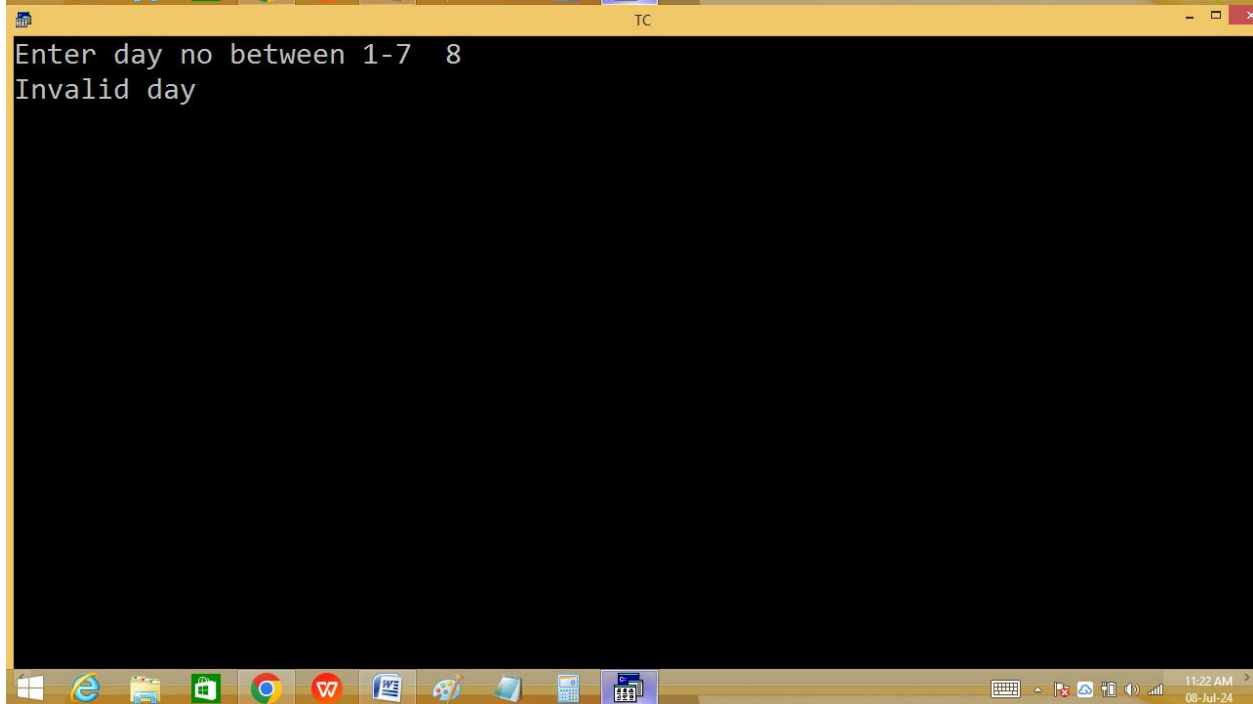
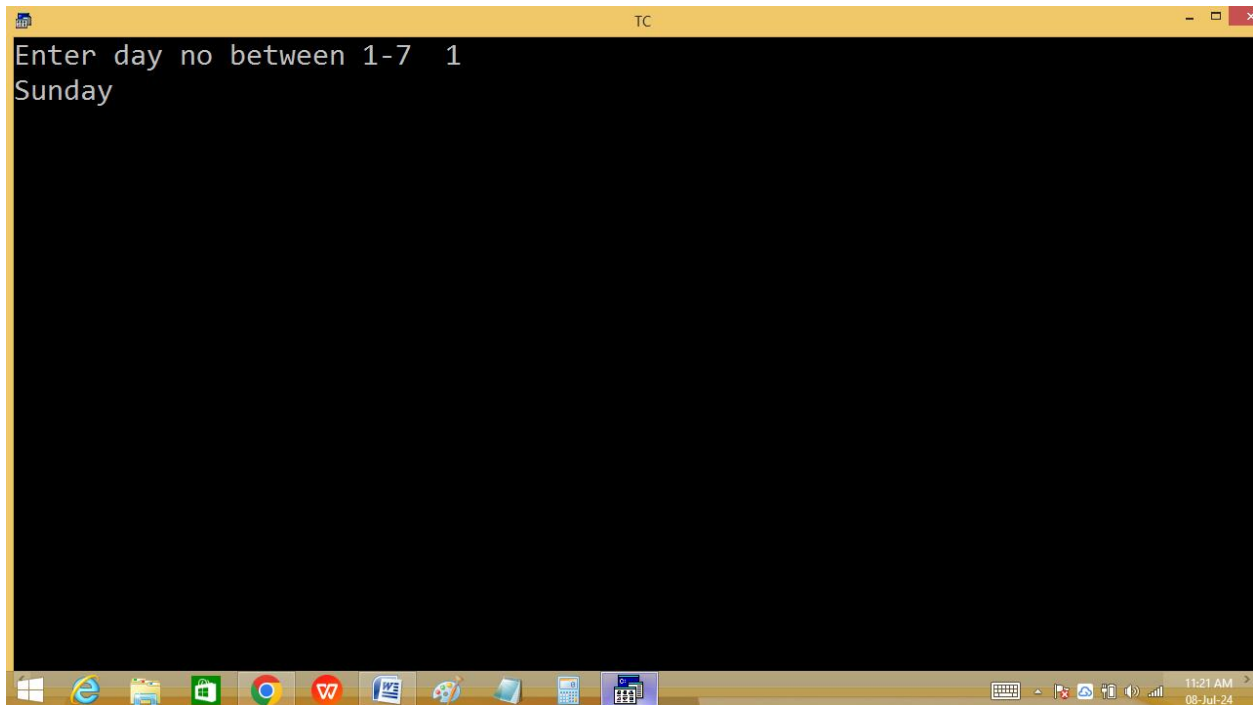


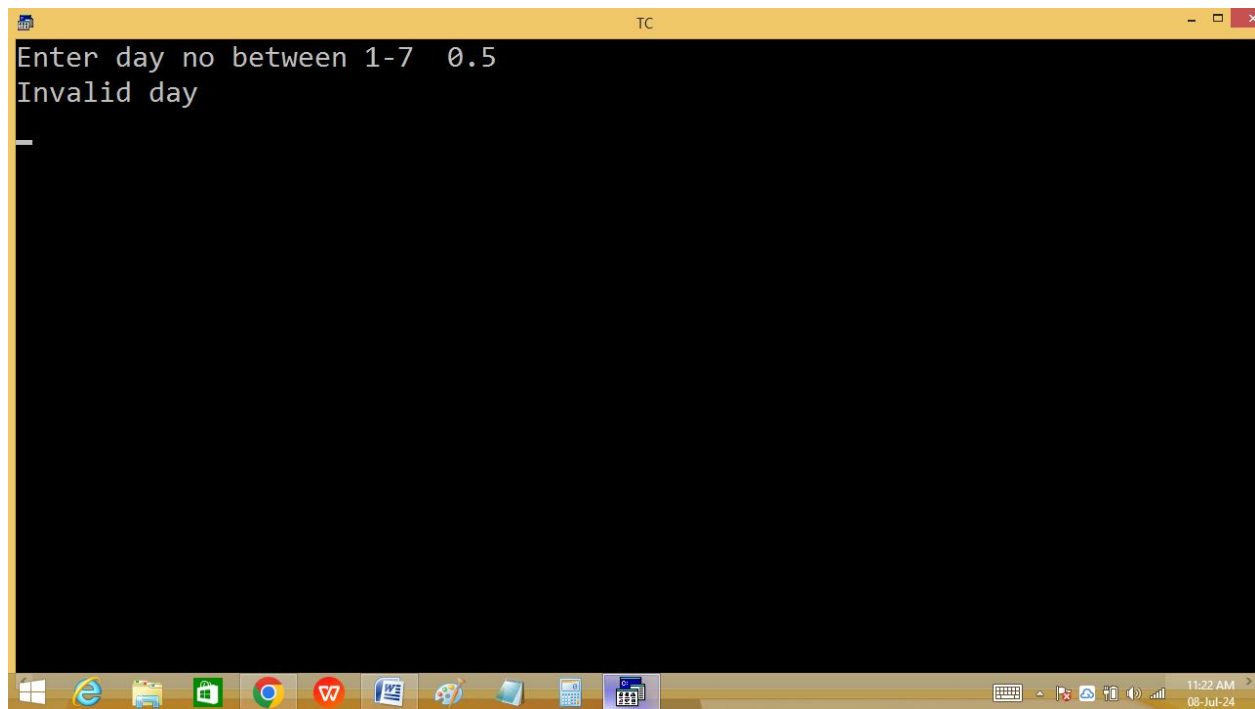
Eg: finding weekday name.



```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 13 Col 27 Insert Indent Tab Fill Unindent * E:11AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n;
clrscr();
printf("Enter day no between 1-7 "); scanf("%d",&n);
switch(n)
{
case 1: puts("Sunday");break;
case 2: puts("Monday");break;
case 7: puts("Saturday");break;
default: puts("Invalid day");
}
getch();
}

TC
Enter day no between 1-7 1
Sunday
```





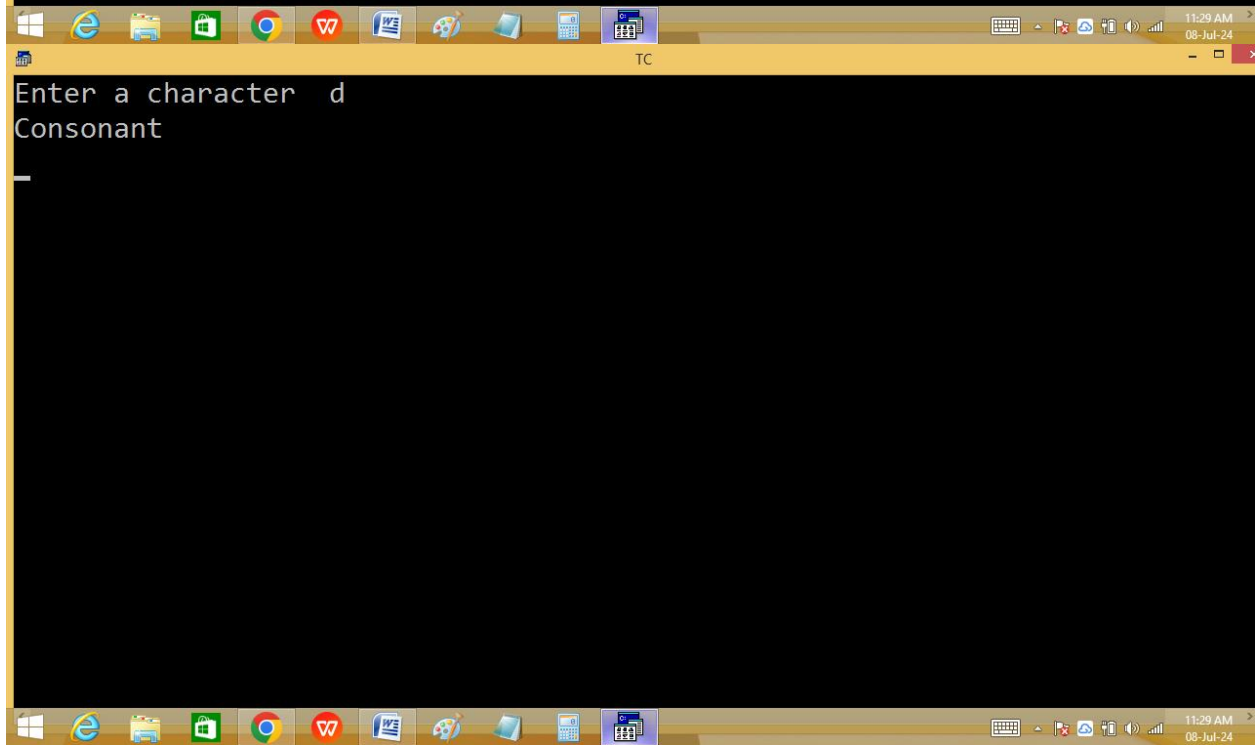
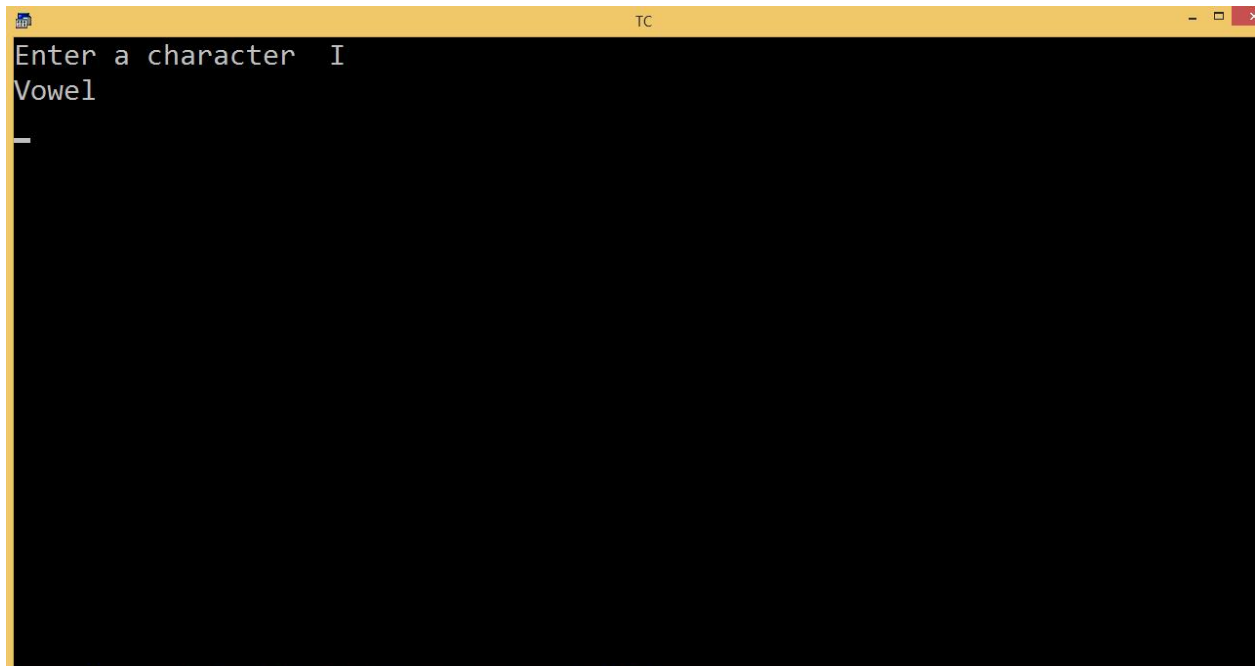
```
TC
Enter day no between 1-7 0.5
Invalid day
_
```

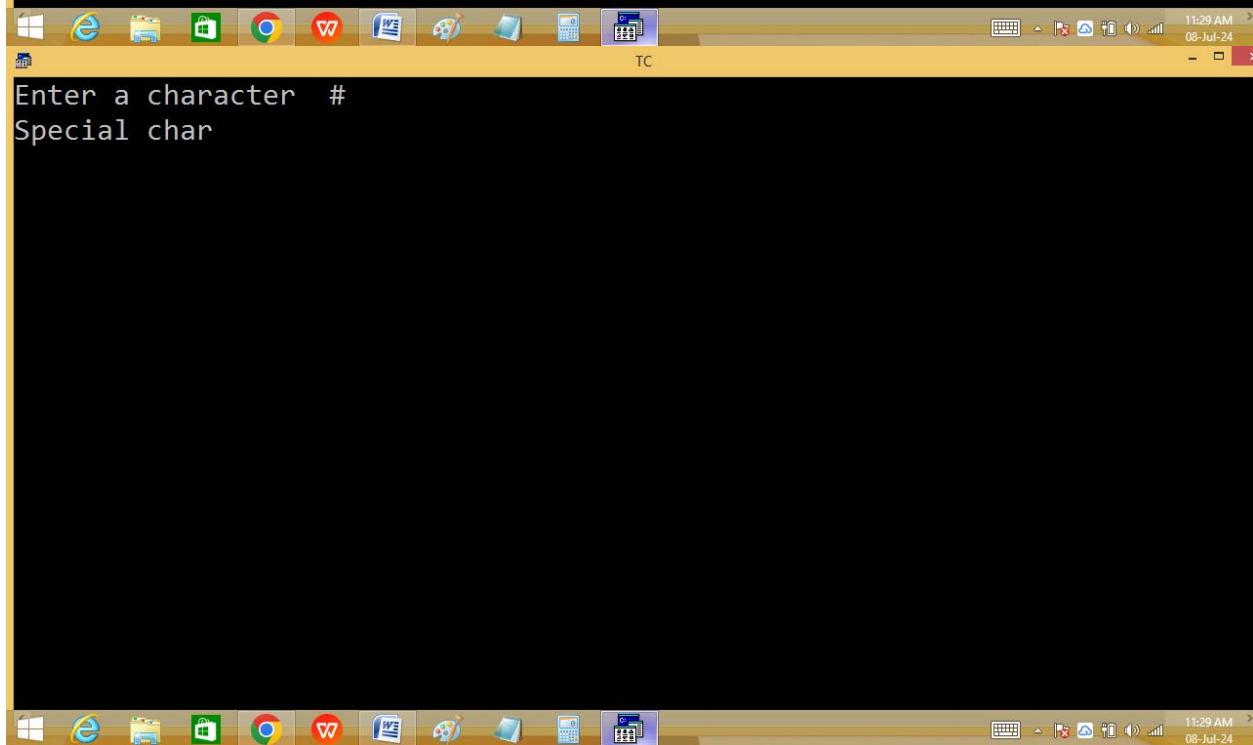
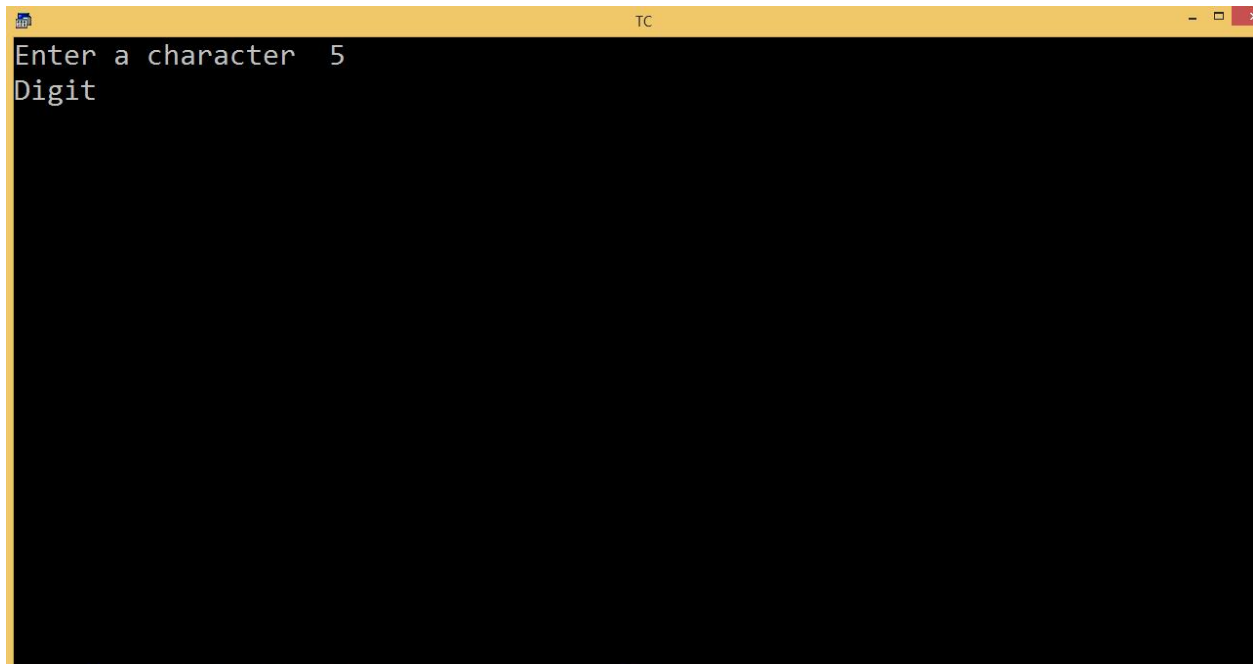
The screenshot shows a Windows 10 desktop environment. A terminal window titled 'TC' is open, displaying a C program that prompts the user to enter a day number between 1 and 7. The user has entered '0.5', and the program has outputted 'Invalid day'. The taskbar at the bottom shows various application icons, including the Start button, Edge browser, File Explorer, Word, and several utility programs. The system tray on the right indicates the time is 11:22 AM on 08-Jul-24.

Finding vowel / consonant using switch:

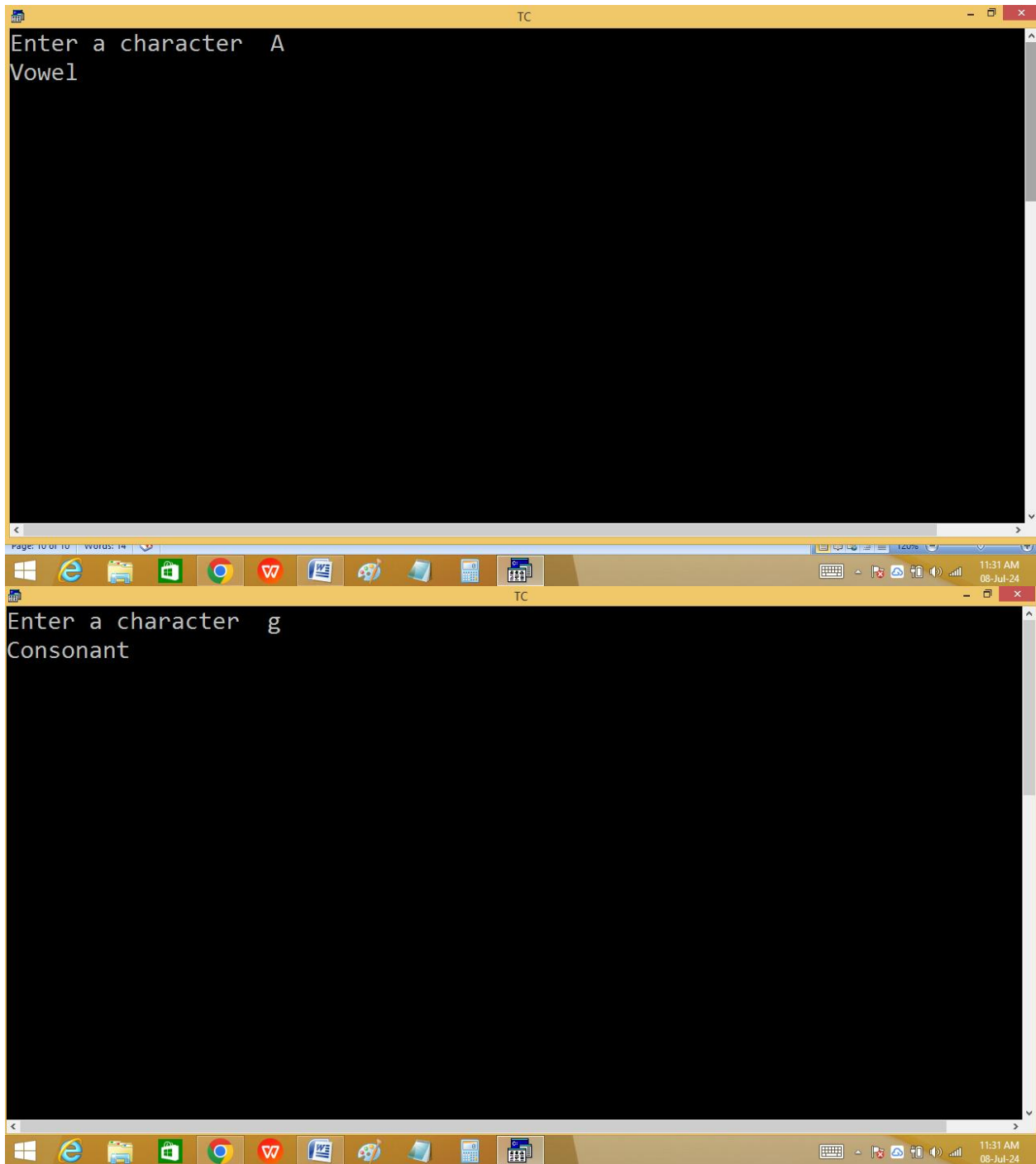
```
TC
#include<stdio.h>  #include<conio.h>
void main()
{
char ch; clrscr();
printf("Enter a character "); scanf("%c",&ch);
if(ch>='A'&&ch<='Z')ch+=32;
if(ch>='a'&&ch<='z')
{
switch(ch)
{
case 'a': case 'e': case 'i': case 'o': case 'u': puts("Vowel");break;
default: puts("Consonant");
}
}
else if(ch>='0' && ch<='9')puts("Digit");
else puts("Special char");
getch();
}

TC
Enter a character  e
Vowel
_
```



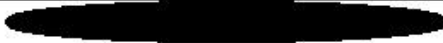


```
TC
#include<stdio.h> #include<conio.h>
void main()
{
char ch; clrscr();
printf("Enter a character "); scanf("%c",&ch);
if(ch>='A'&&ch<='Z')ch+=32;
if(ch>='a'&&ch<='z')
{
switch(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
{
case 1: puts("Vowel");break;
default: puts("Consonant");
}
}
else if(ch>='0' && ch<='9')puts("Digit");
else puts("Special char");
getch();
}
```



Read two numbers and perform arithmetic operation using a menu with switch.

Enter two numbers 7.7 2.2	
M E N U	
+. ADD -. SUB *. MUL %. MOD /. DIV E. EXIT	
Enter ur option[+]	sum= 9.9



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

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

```
#include<math.h>
```

```
void main()
```

```
{
```

```
float a,b; char op;
```

```
abc:
```

```
clrscr();
```

```
printf("Enter two numbers "); scanf("%f %f",&a, &b);
```

```
puts("-----");
puts("\t\t\t M   E   N   U");
puts("-----");
puts("\t\t\t +. Add");
puts("\t\t\t -. Sub");
puts("\t\t\t *. Mul");
puts("\t\t\t %. Mod");
puts("\t\t\t /. Div");
puts("\t\t\t E. Exit");
puts("-----");
flushall();
printf("\t\t\t Enter Ur option[ ]\b\b"); scanf("%c",&op);
gotoxy(60, 16);
switch(op)
{
case '+': printf("Sum=%.2f",a+b);break;
case '-': printf("Sub=%.2f",a-b);break;
case '*': printf("Mul=%.2f",a*b);break;
```

```
case '%': printf("Mod=%.2f",fmod(a,b));break;

case '/': printf("Div=%.2f",a/b);break;

case 'e':case 'E':return;

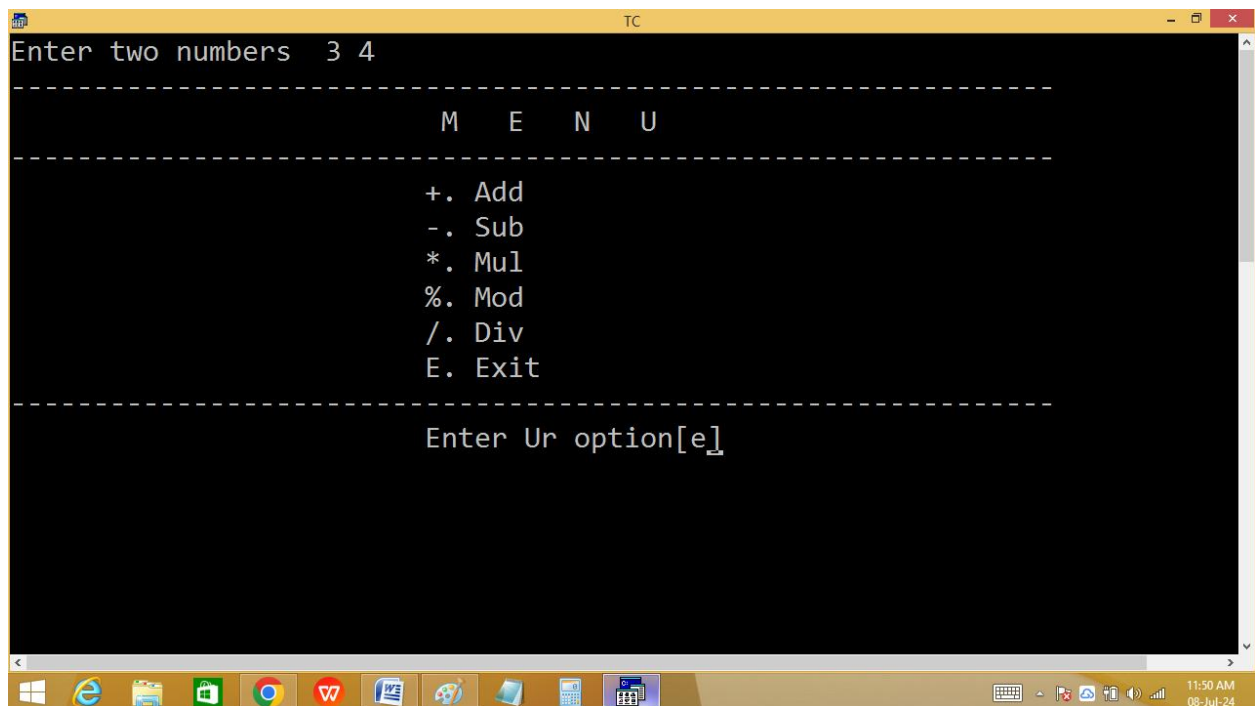
default: printf("Invalid selection");

}

getch();

goto abc;

}
```



```
TC
Enter two numbers 3 4
-----
                M   E   N   U
-----
+. Add
-. Sub
*. Mul
%. Mod
/. Div
E. Exit
-----
Enter Ur option[e]
```

```
TC
Enter two numbers 2 5
-----
M E N U
-----
+. Add
-. Sub
*. Mul
%. Mod
/. Div
E. Exit
-----
Enter Ur option[/]

Div=0.40_
```

```
TC
Enter two numbers 3.3 2.4
-----
M E N U
-----
+. Add
-. Sub
*. Mul
%. Mod
/. Div
E. Exit
-----
Enter Ur option[%]

Mod=0.90_
```

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

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

```
void main()
{
float amt=0,qty; int op;
abc:
clrscr();
puts("-----");
puts("\t\t HOTEL SAI KRISHNA");
puts("-----");
puts("\t\t 1. Tea - 10/-");
puts("\t\t 2. Coffee/Milk/Boost/Lemon Tea/Water bottle - 20/-");
puts("\t\t 3. Idly / Bonda / Poha / Upma / Plain Dosa - 30/-");
puts("\t\t 4. Masala / Onion / Karam dosa / wada / sambar
idly/poori - 40/-");
puts("\t\t 5. Ghee dosa / Paneer dosa / Curd rice/ Sambar wada
50/- ");
puts("\t\t 6. Veg biryani - 60/-");
puts("\t\t 7. Bill");
puts("\t\t 8. Cancel");
```

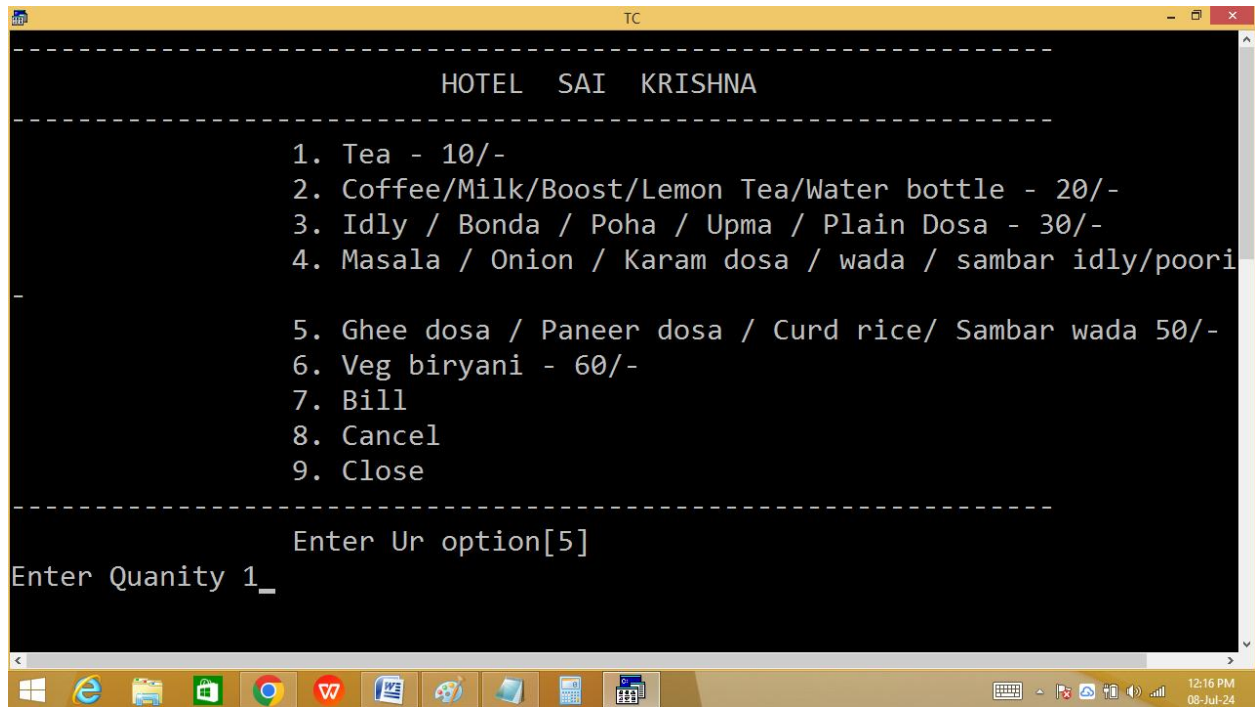


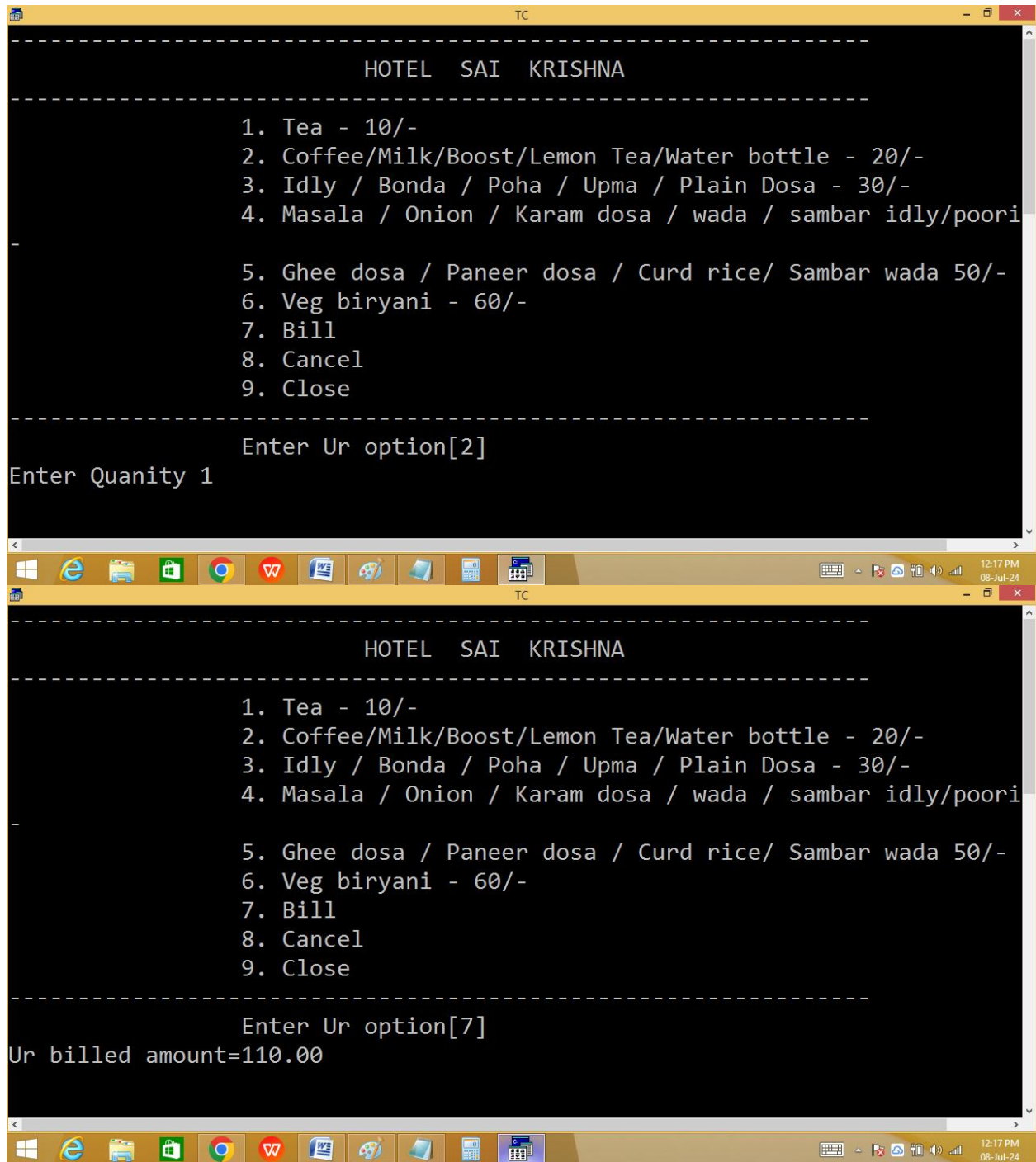
```
puts("\t\t 9. Close");  
puts("-----");  
printf("\t\t Enter Ur option[ ]\b\b"); scanf("%d",&op);  
if(op<=6){printf("Enter Quantity "); scanf("%f",&qty);}  
switch(op)  
{  
case 1: amt+=qty*10;break;  
case 2: amt+=qty*20;break;  
case 3: amt+=qty*30;break;  
case 4: amt+=qty*40;break;  
case 5: amt+=qty*50;break;  
case 6: amt+=qty*60;break;  
case 7: printf("Ur billed amount=%.2f",amt);  
amt=0;getch();break;  
case 8: printf("Ur order cancelled");amt=0;getch();break;  
case 9: return;  
default: printf("Invalid selection");getch();  
}  
goto abc;
```

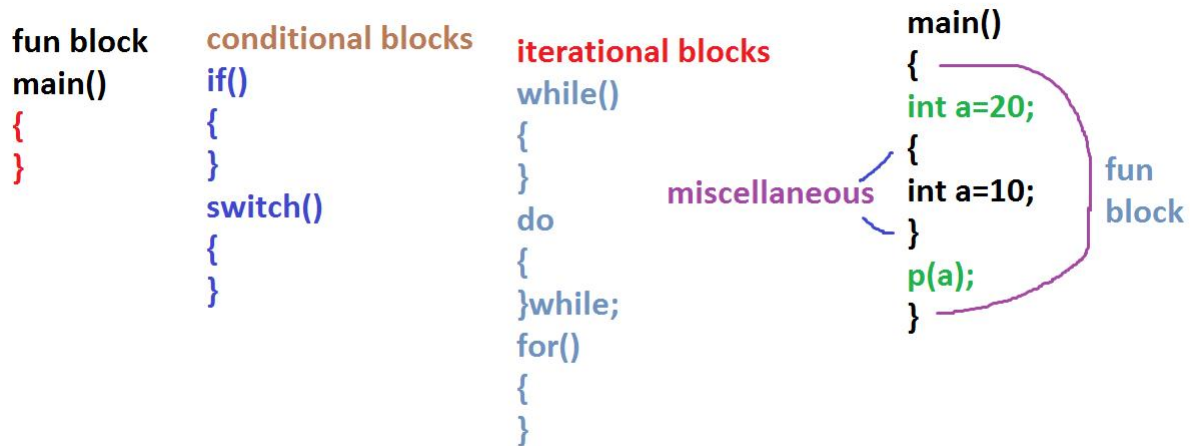
}

```
-----  
HOTEL SAI KRISHNA  
-----  
1. Tea - 10/-  
2. Coffee/Milk/Boost/Lemon Tea/Water bottle - 20/-  
3. Idly / Bonda / Poha / Upma / Plain Dosa - 30/-  
4. Masala / Onion / Karam dosa / wada / sambar idly/poori  
-  
5. Ghee dosa / Paneer dosa / Curd rice/ Sambar wada 50/-  
6. Veg biryani - 60/-  
7. Bill  
8. Cancel  
9. Close  
-----  
Enter Ur option[_]
```

```
-----  
HOTEL SAI KRISHNA  
-----  
1. Tea - 10/-  
2. Coffee/Milk/Boost/Lemon Tea/Water bottle - 20/-  
3. Idly / Bonda / Poha / Upma / Plain Dosa - 30/-  
4. Masala / Onion / Karam dosa / wada / sambar idly/poori  
-  
5. Ghee dosa / Paneer dosa / Curd rice/ Sambar wada 50/-  
6. Veg biryani - 60/-  
7. Bill  
8. Cancel  
9. Close  
-----  
Enter Ur option[4]  
Enter Quantity 1
```







Loops / Iterations / Repetitive statements

Loops are used to repeat a block/group of statements continuously until the given condition becomes false.

Loops reduce program size and improves performance.

In loops beginning and ending points are same.

C-Language supports basically 2 types of loops.

1. Entry/pre controlled loops.

2. Exit/post controlled loops.

In entry control loops, condition is tested first and it is true then only statements block is executed.

Under entry control loops we are having

- i. While loop**
- ii. For loop**

In exit control loop, the statements are executed first and later condition is tested.

Under exit control loop we are having

- i. do while.**

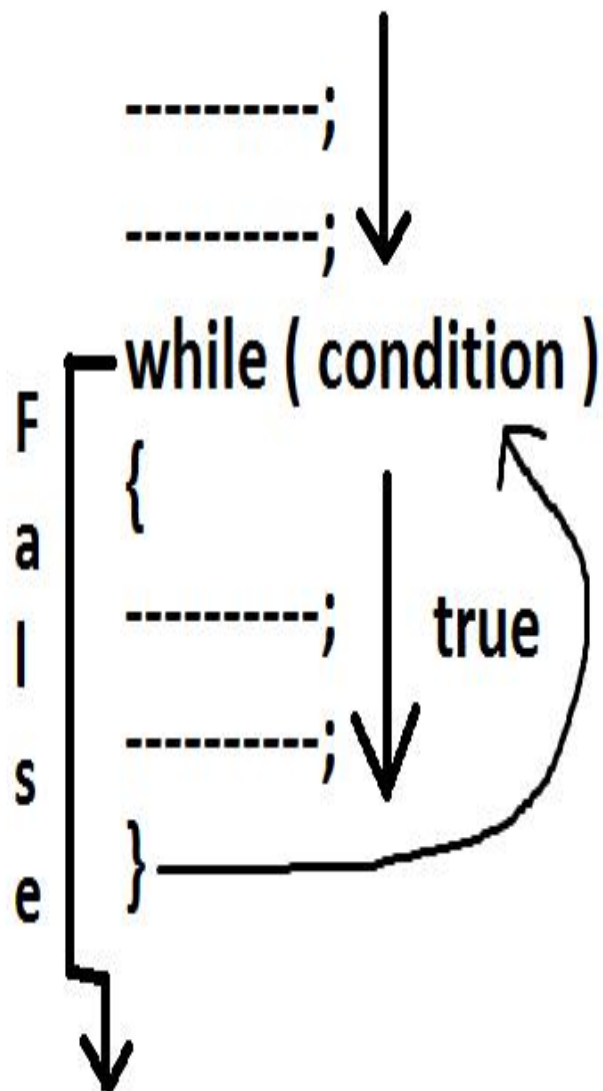
While loop:

- **while is a keyword.**
- **In while loop condition is tested first and it is true then only while block statements are executed. After executing**

while block statements, the program execution automatically shifted/jumped to while condition at the beginning. If it is true then once again the while block statements are repeated. Like this the process is continued until while condition becomes false.

➤ **While is entry control loop.**

Syntax:



Flow chart:

