LOOPS / ITERATIONS / REPETITITVE 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.

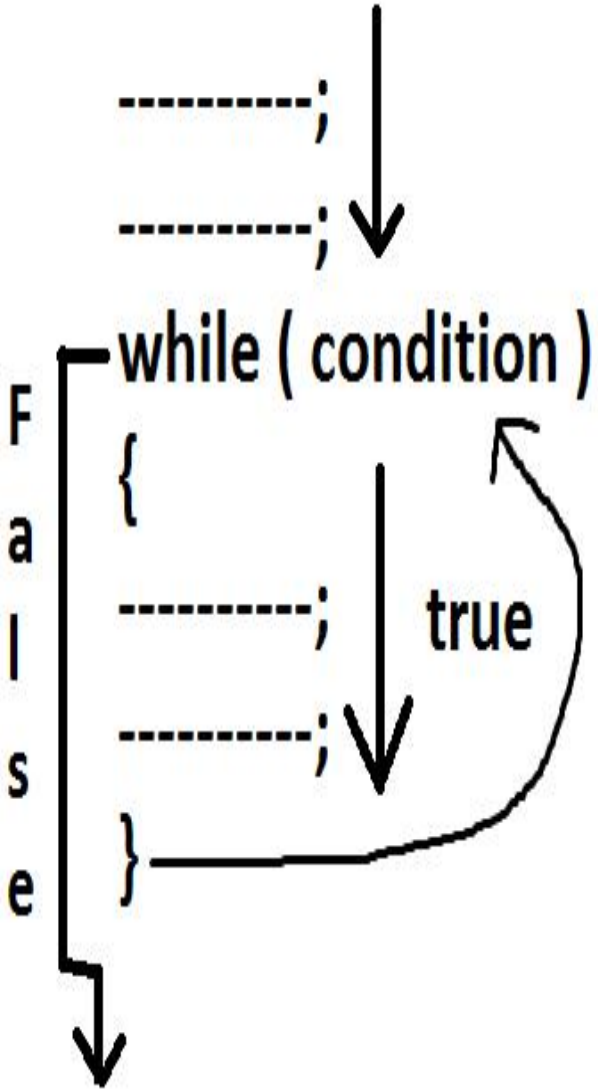
While loop:

- 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

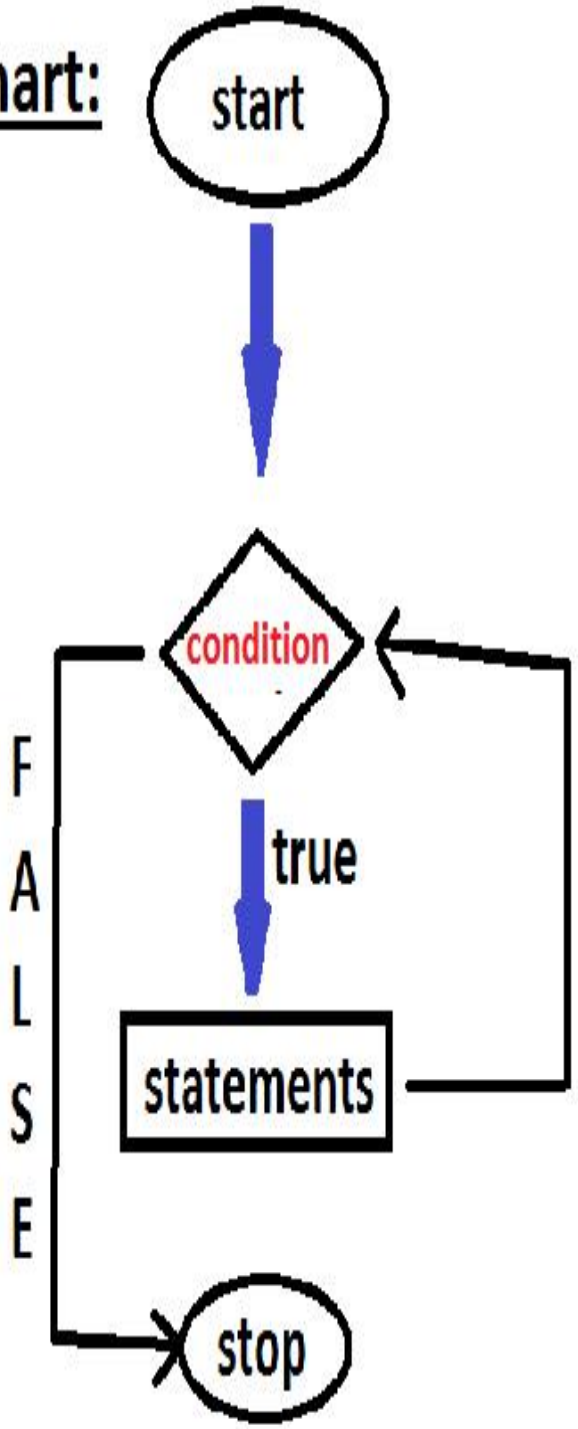
process is continued until while condition becomes false.

- **While is entry control loop.**

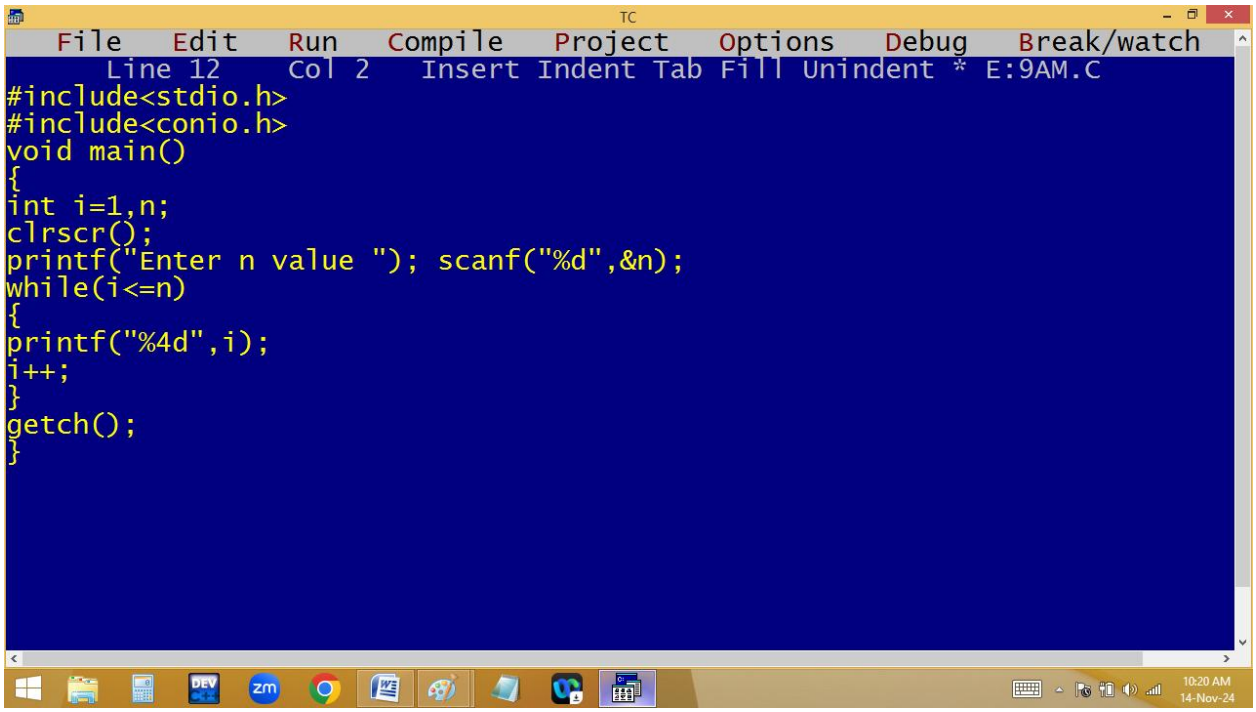
Syntax:

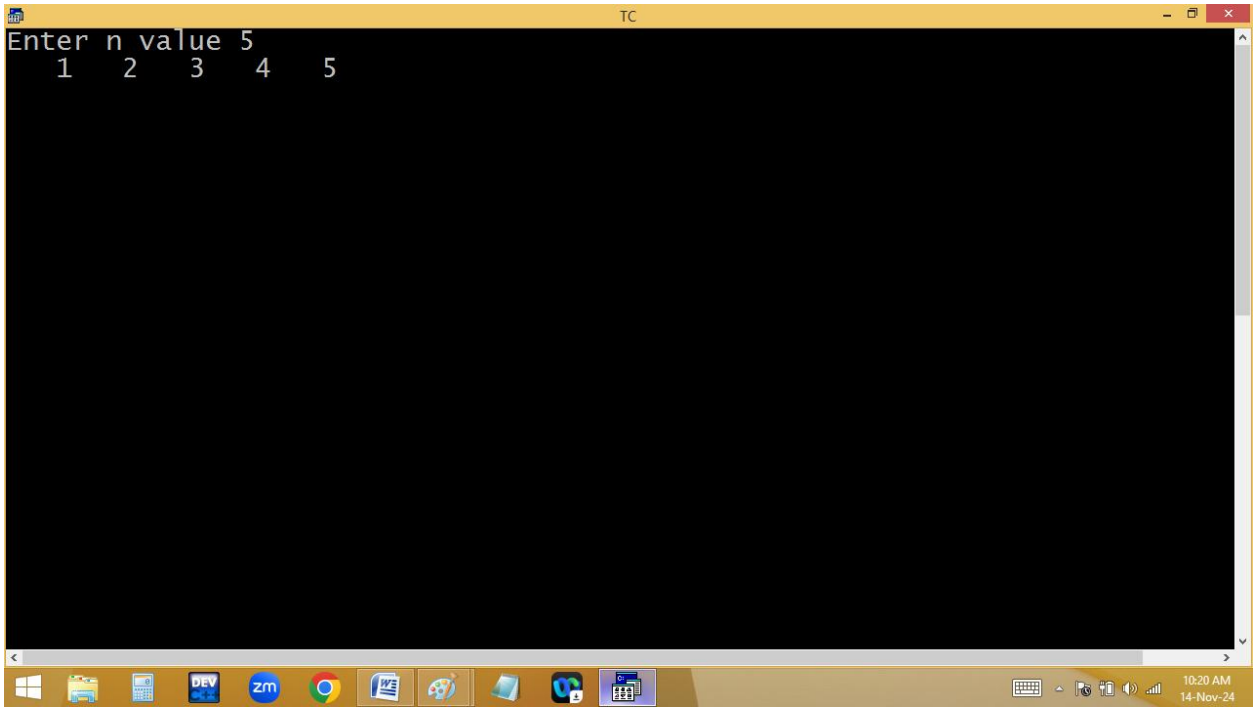


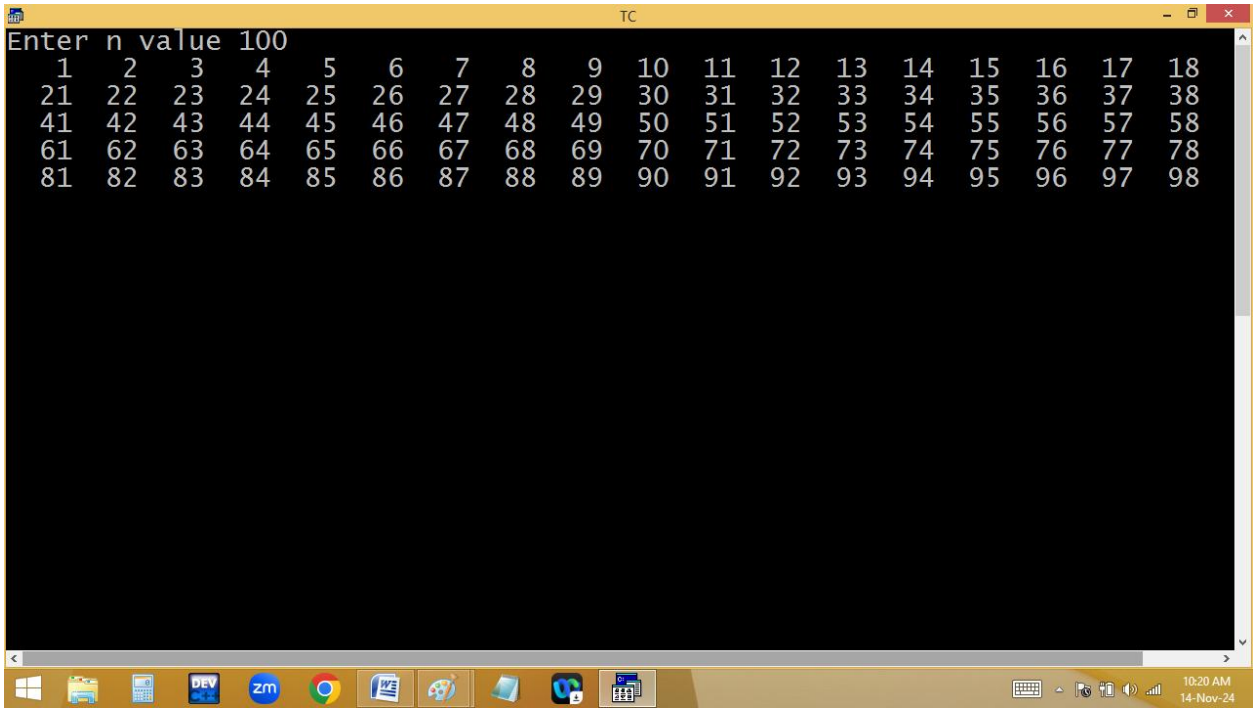
Flow chart:

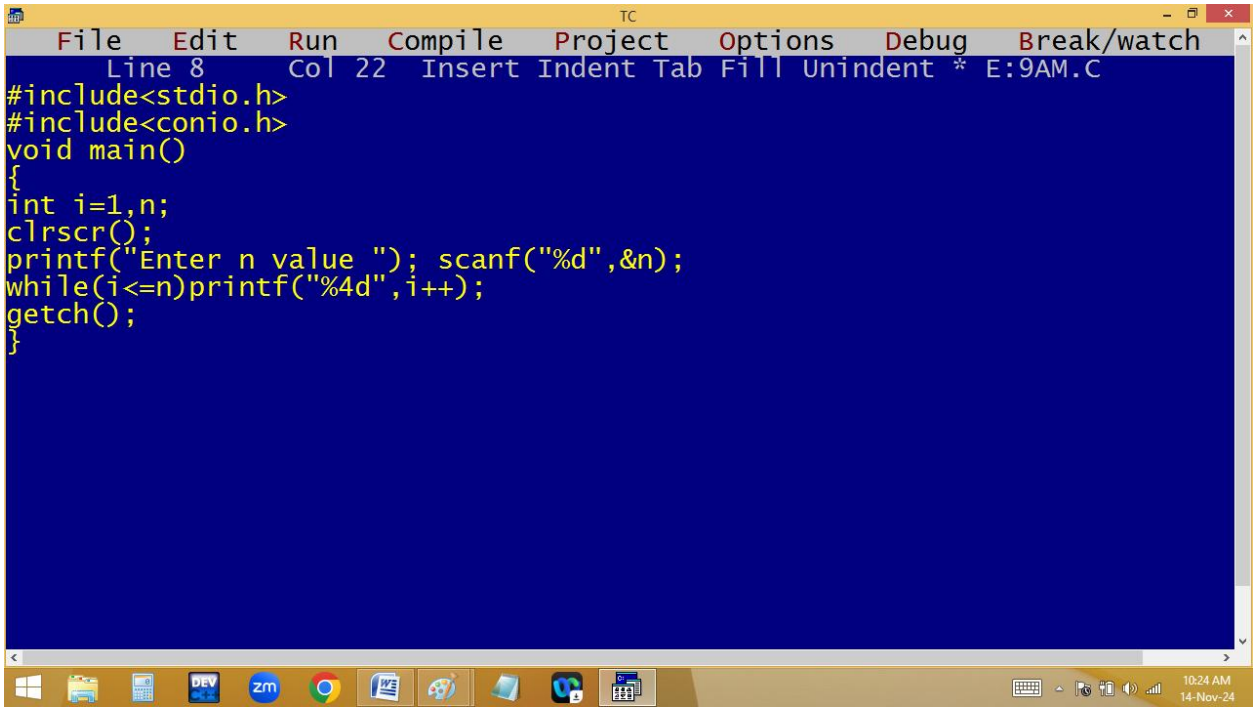


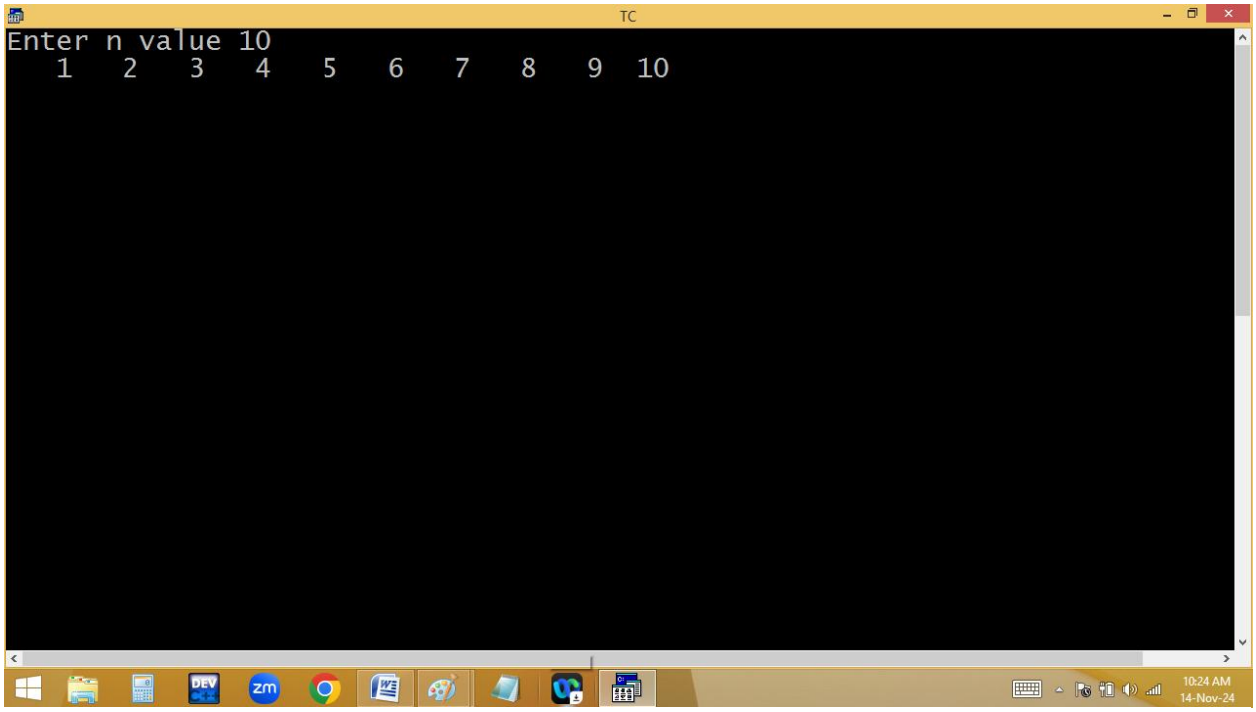
Printing 1..n natural numbers:

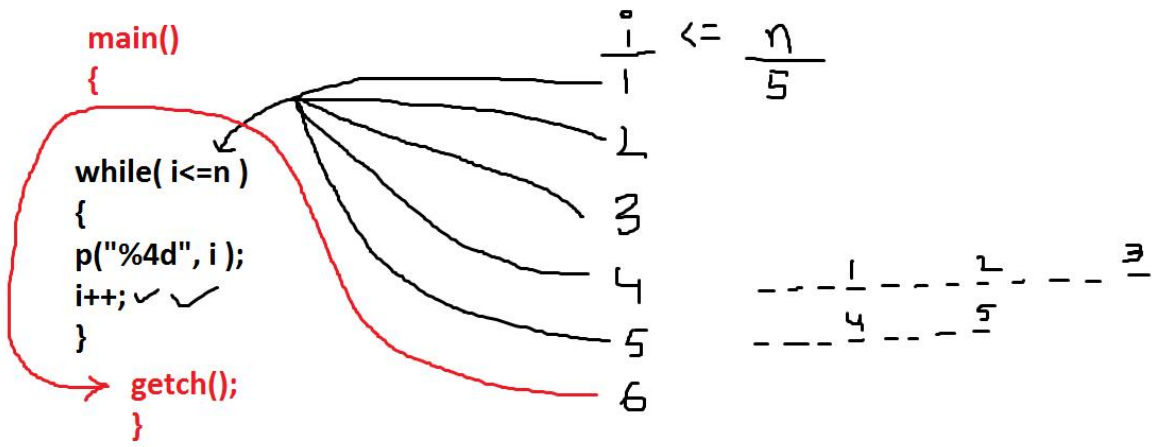





TC
Enter n value 100
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58
61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98
10:20 AM
14-Nov-24

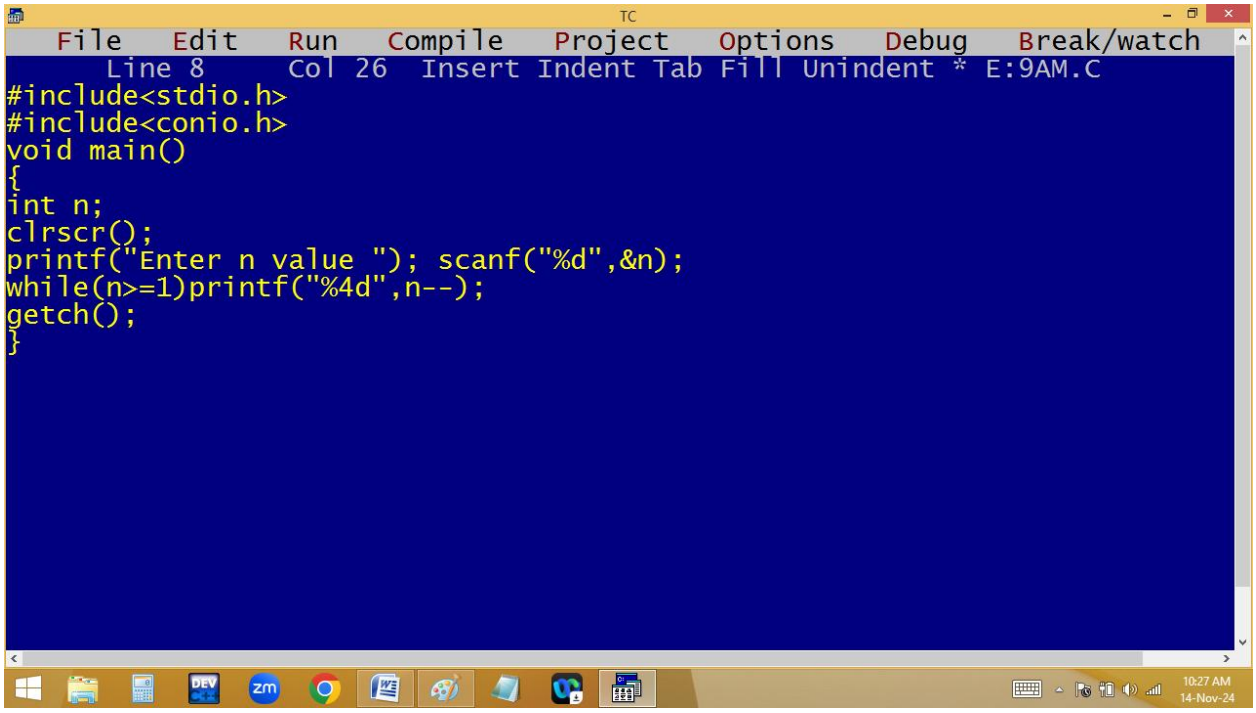

TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 22 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int i=1,n;
clrscr();
printf("Enter n value "); scanf("%d",&n);
while(i<=n)printf("%4d",i++);
getch();
}
10:24 AM
14-Nov-24

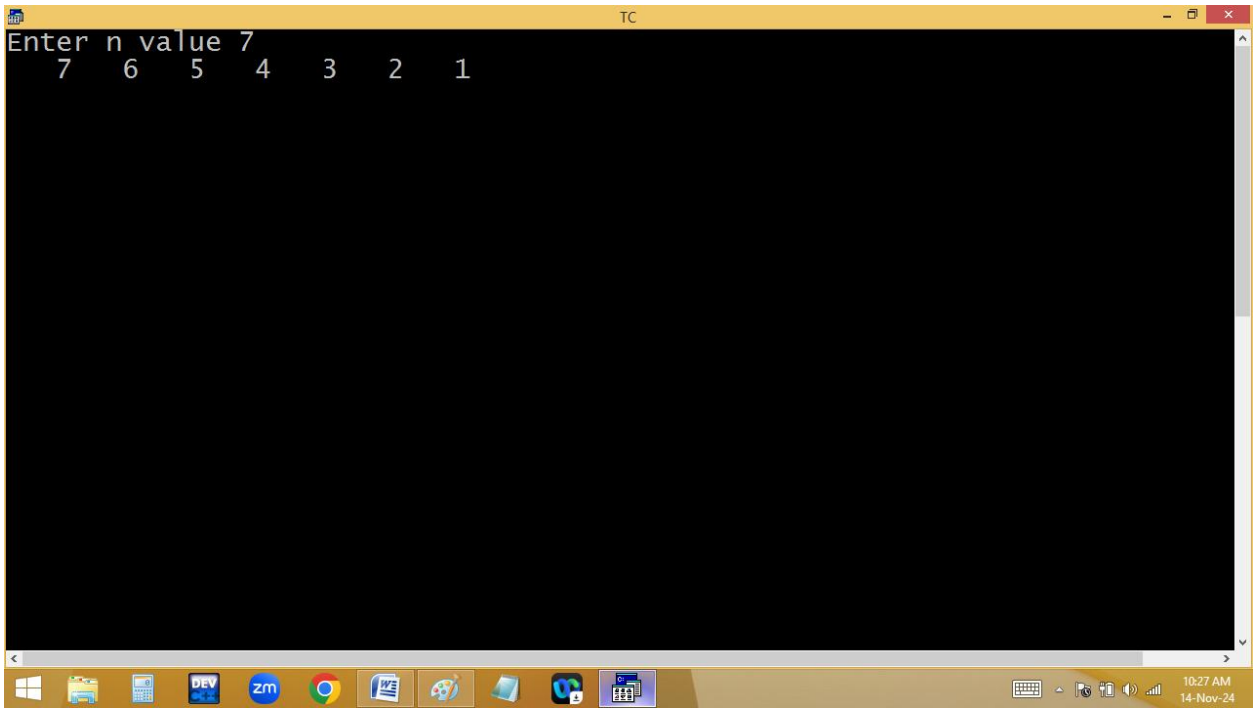




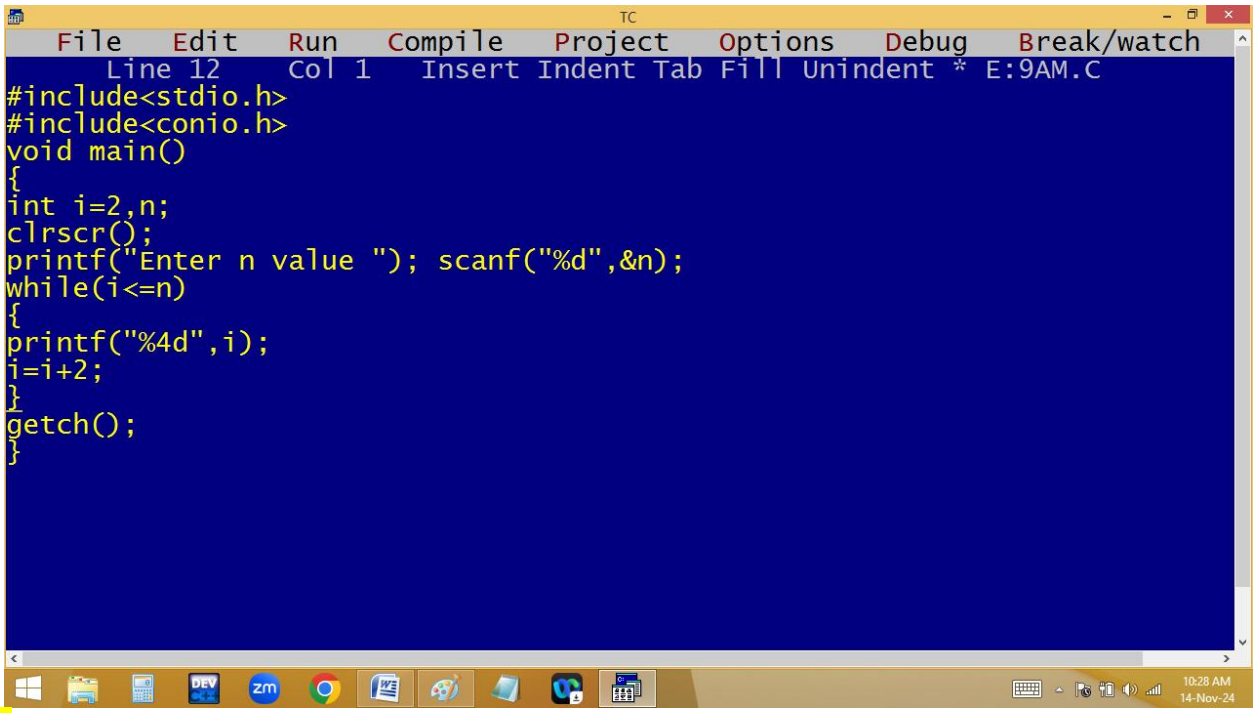
1..n in reverse order:

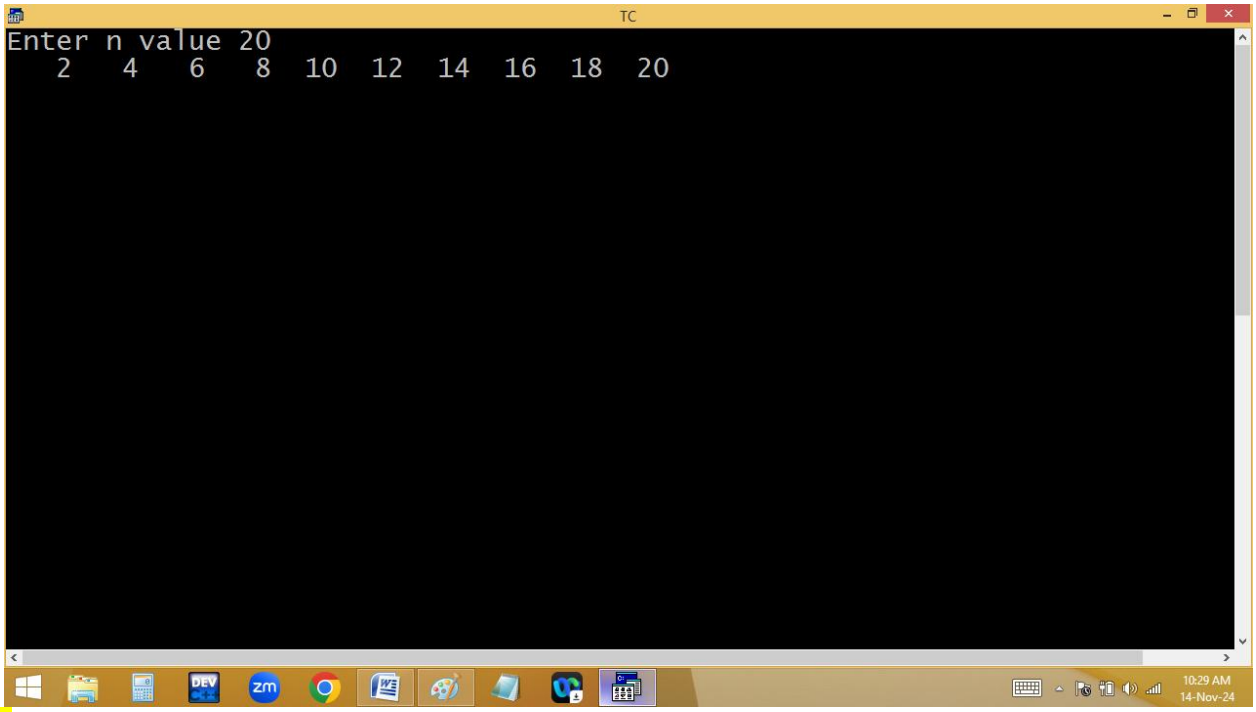
N=5 ➔ 5 4 3 2 1

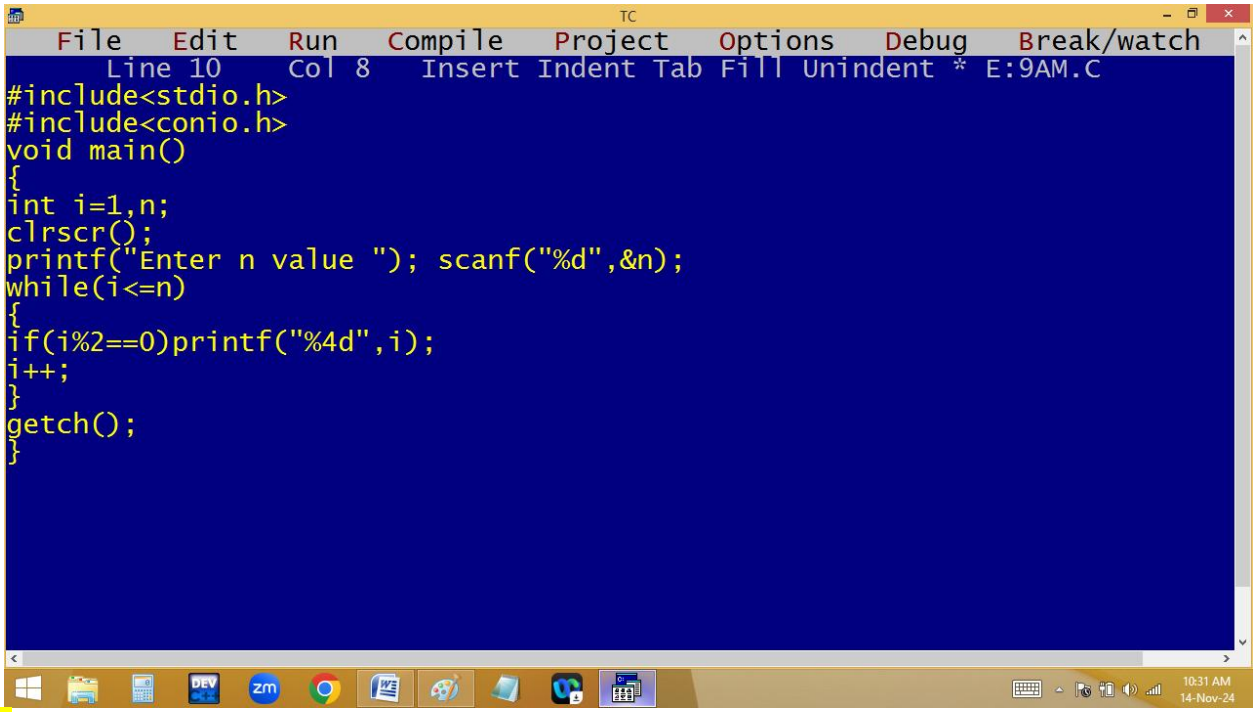

TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 26 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n;
clrscr();
printf("Enter n value "); scanf("%d",&n);
while(n>=1)printf("%4d",n--);
getch();
}
10:27 AM
14-Nov-24

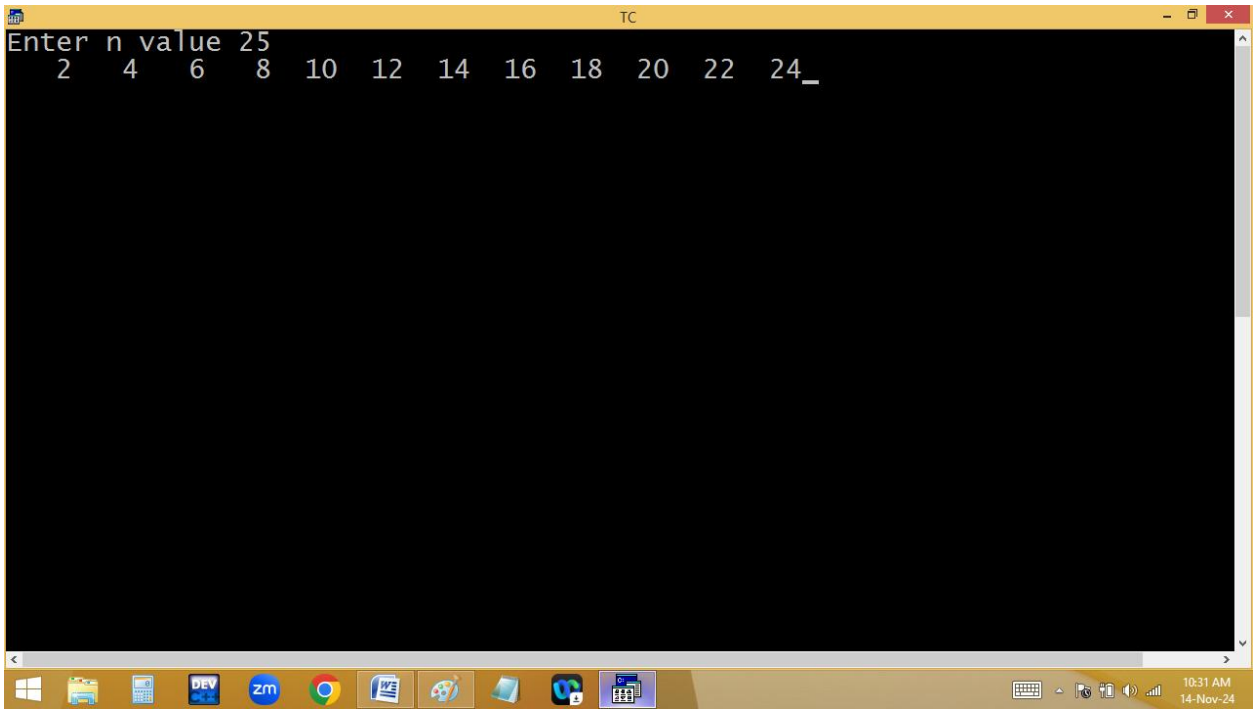

TC
Enter n value 7
7 6 5 4 3 2 1
10:27 AM
14-Nov-24

Print 1 .. n even no's:

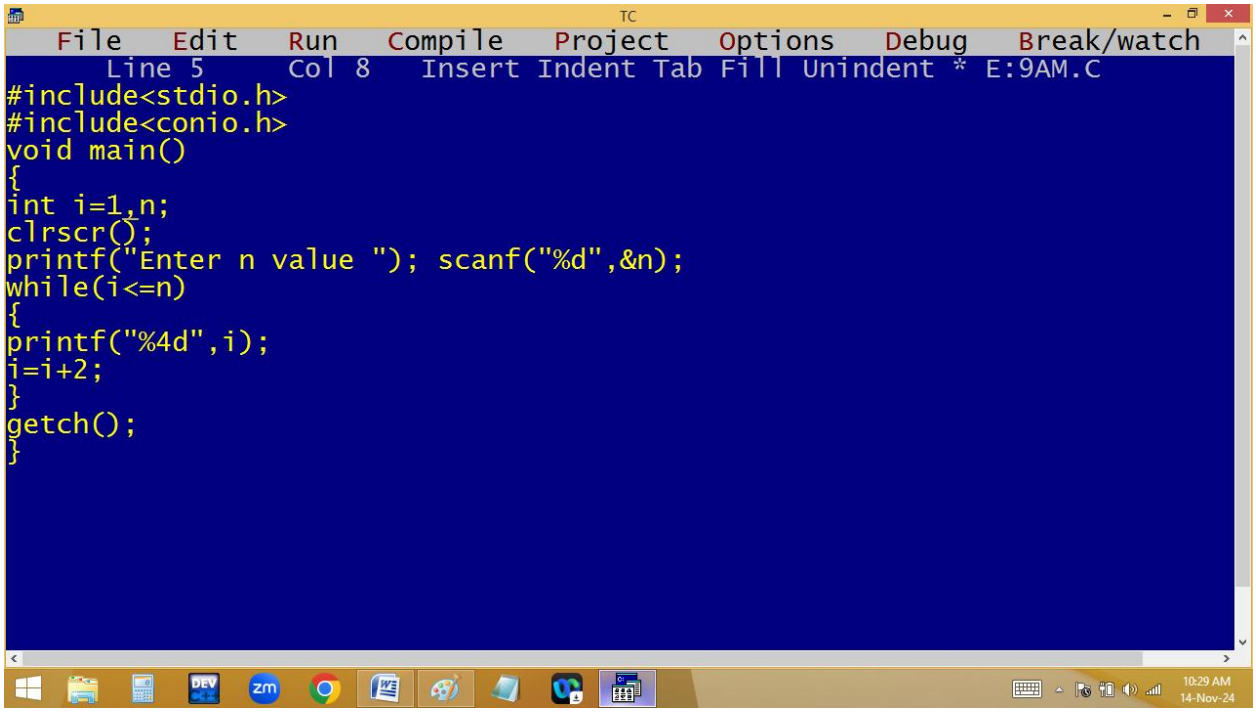


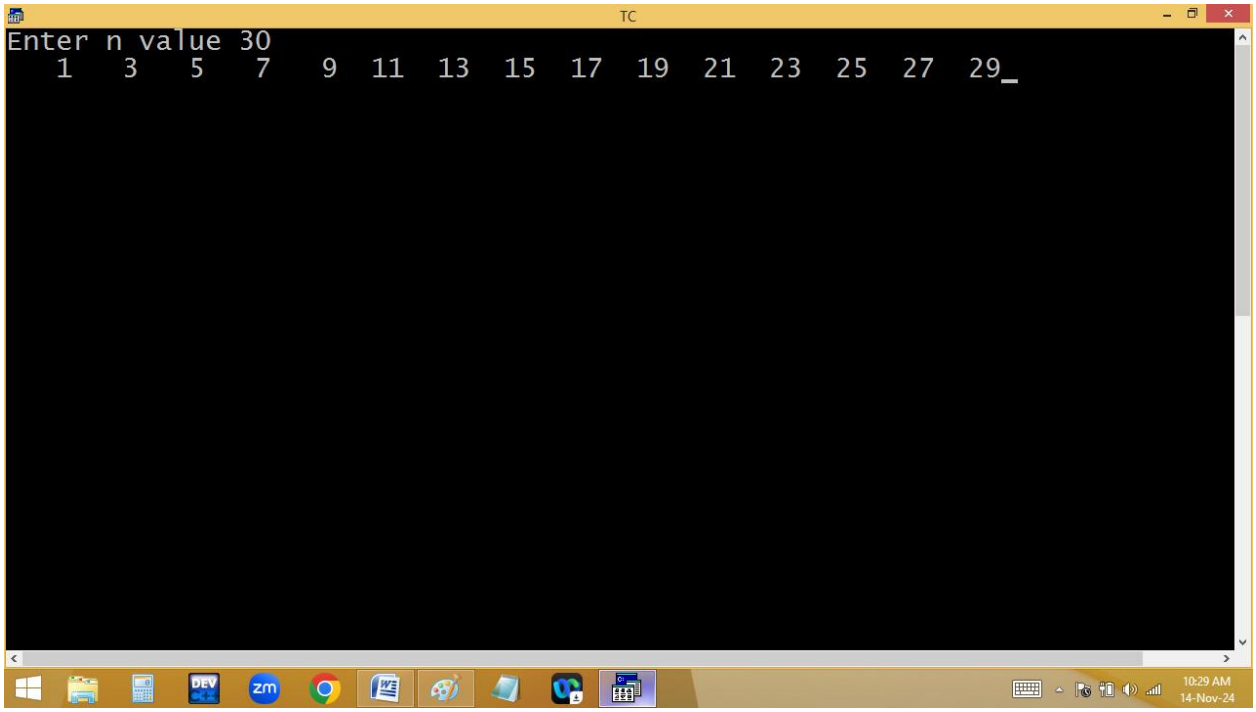


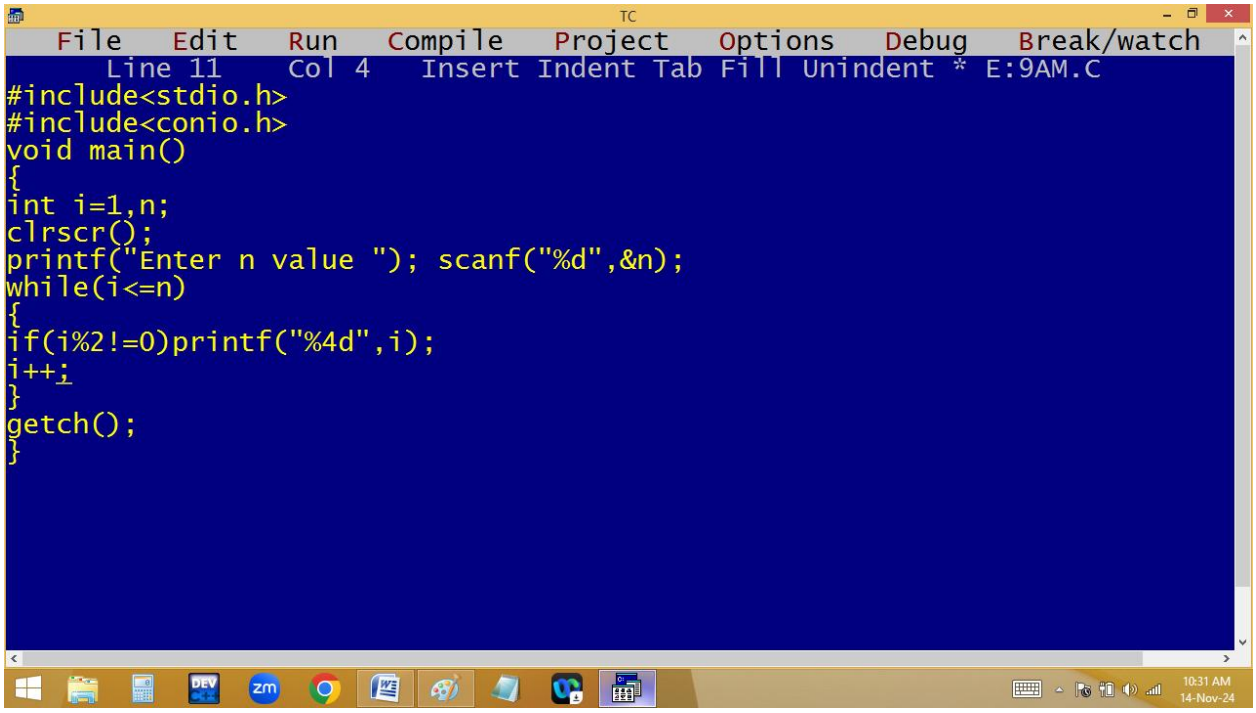


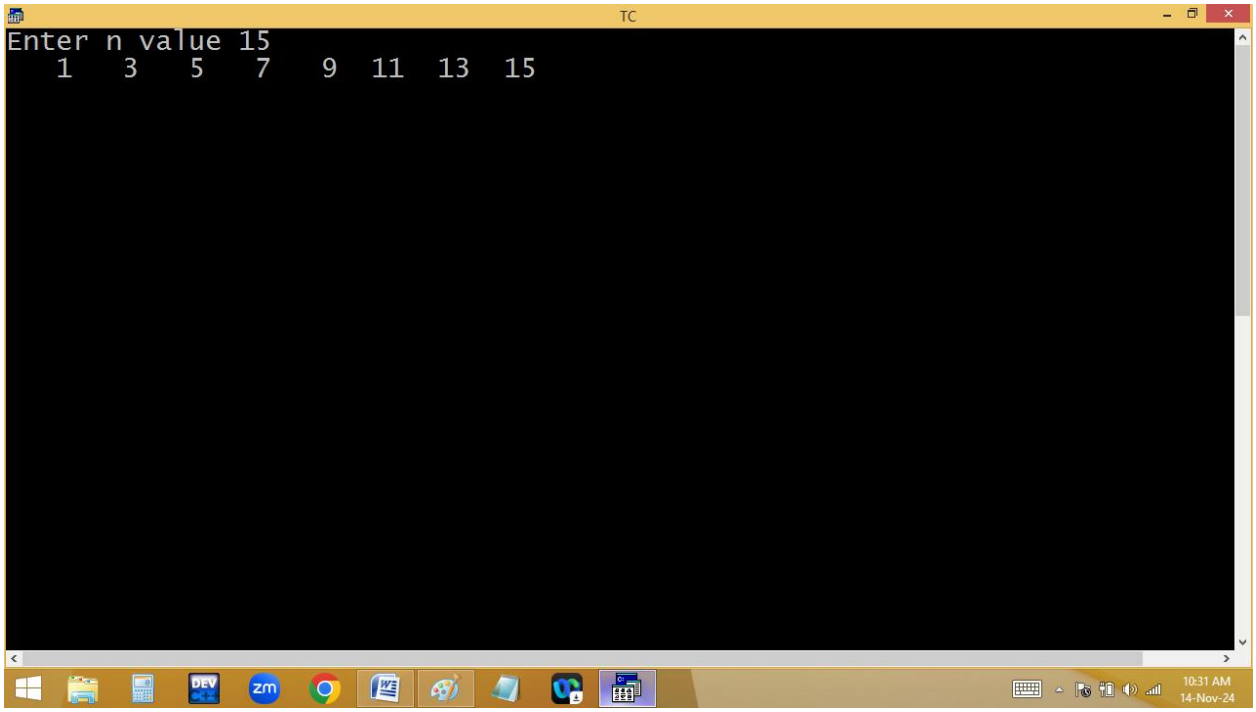


Printing 1..n odd numbers:


TC
File Edit Run Compile Project Options Debug Break/watch
Line 5 Col 8 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int i=1,n;
clrscr();
printf("Enter n value "); scanf("%d",&n);
while(i<=n)
{
printf("%4d",i);
i=i+2;
}
getch();
}

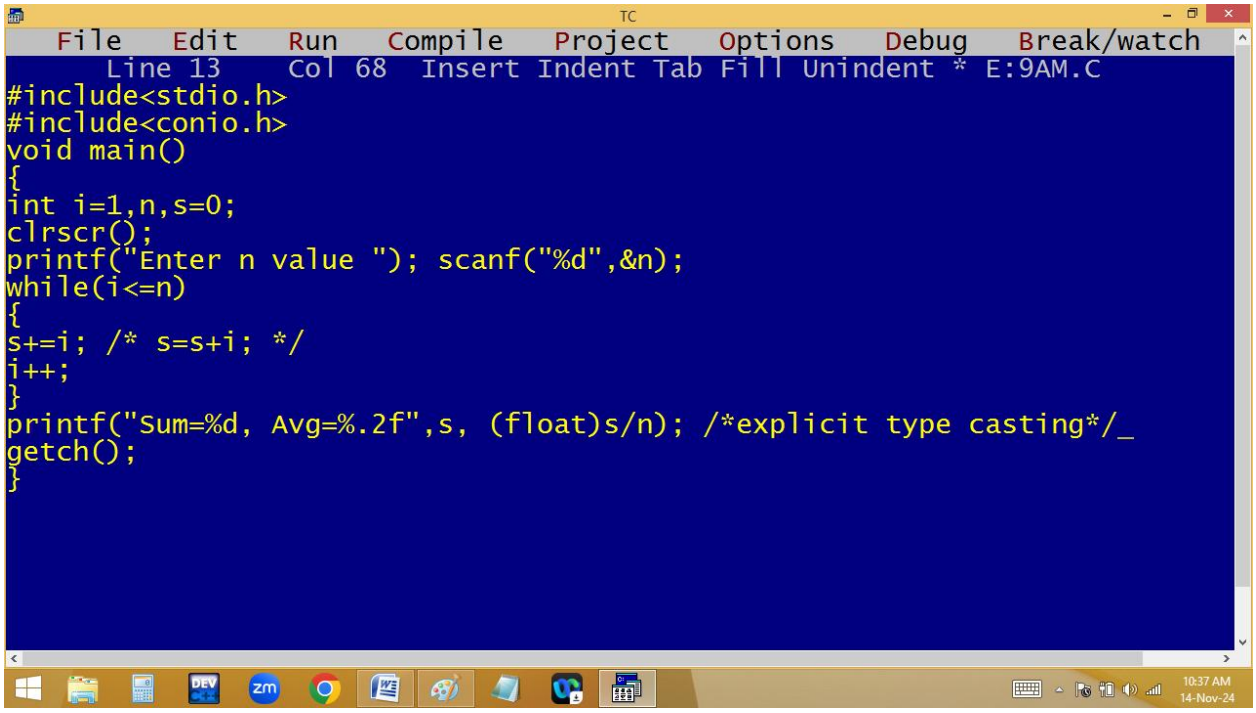

TC
Enter n value 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29

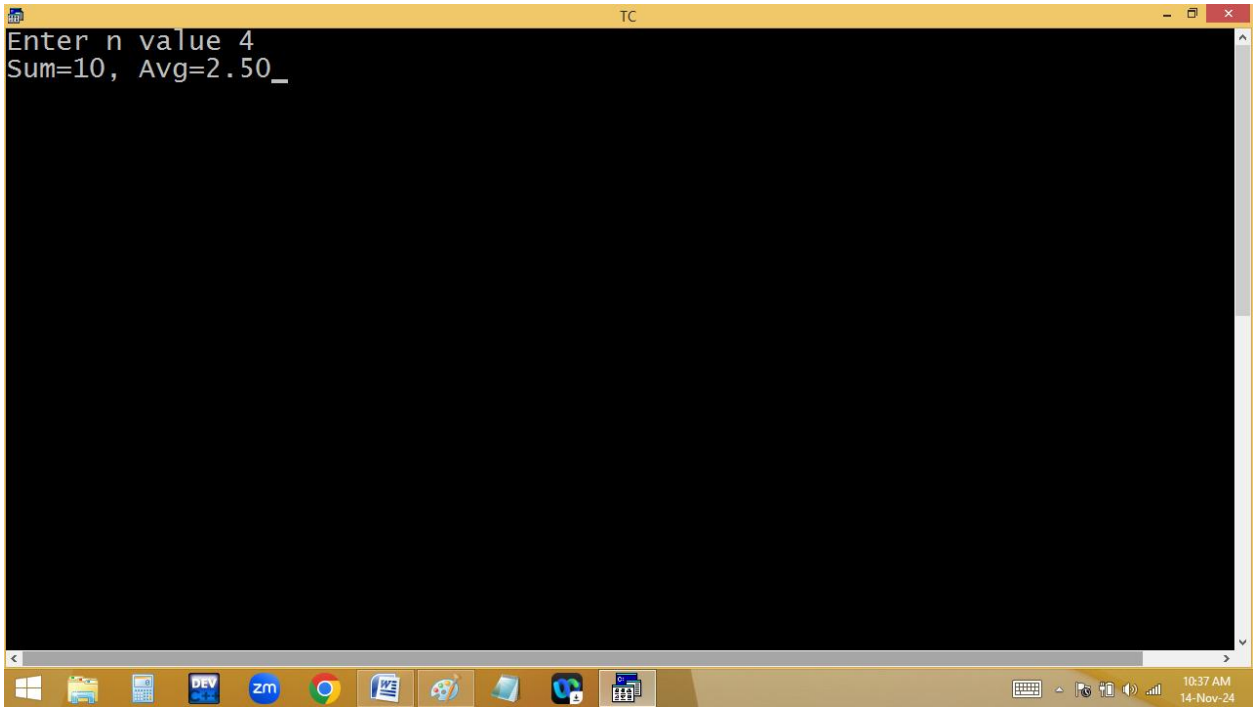


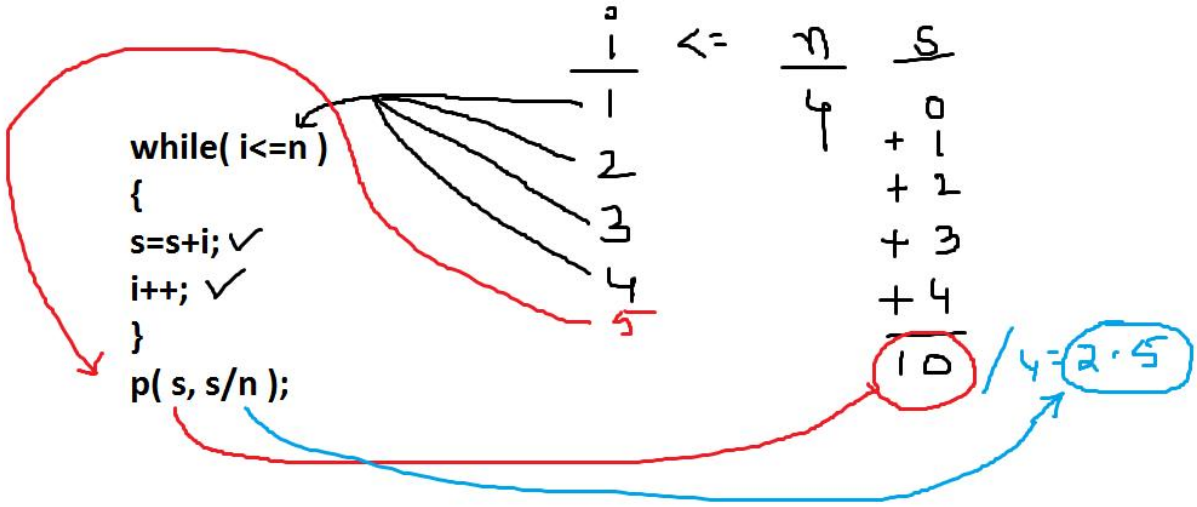


Finding 1..n numbers sum and avg [mean]:

n=4 → 1 + 2 + 3 + 4 = 10 ← sum → 10/4=2.5←avg


TC
File Edit Run Compile Project Options Debug Break/watch
Line 13 Col 68 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int i=1,n,s=0;
clrscr();
printf("Enter n value "); scanf("%d",&n);
while(i<=n)
{
s+=i; /* s=s+i; */
i++;
}
printf("Sum=%d, Avg=%.2f",s, (float)s/n); /*explicit type casting*/_
getch();
}
10:37 AM
14-Nov-24



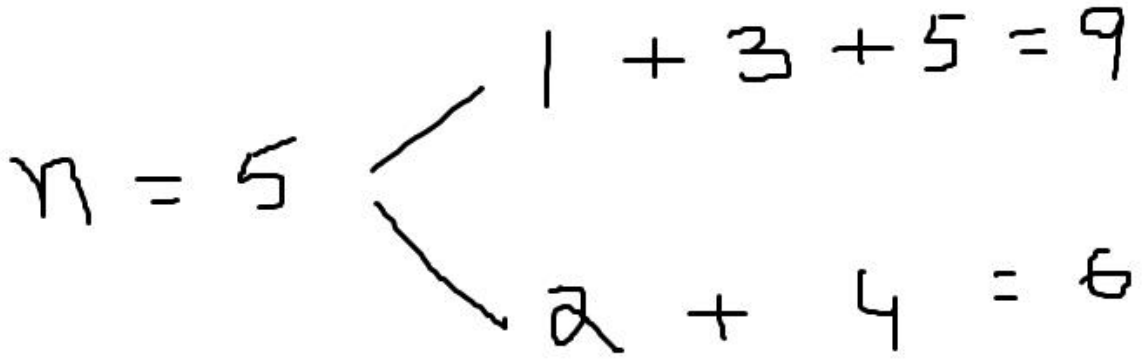


Without using loop:

$4 * 5 = 20/2 = 10$

```
s = n * ( n+1 ) /2 ;
```

Find 1..n even , odd numbers sum:

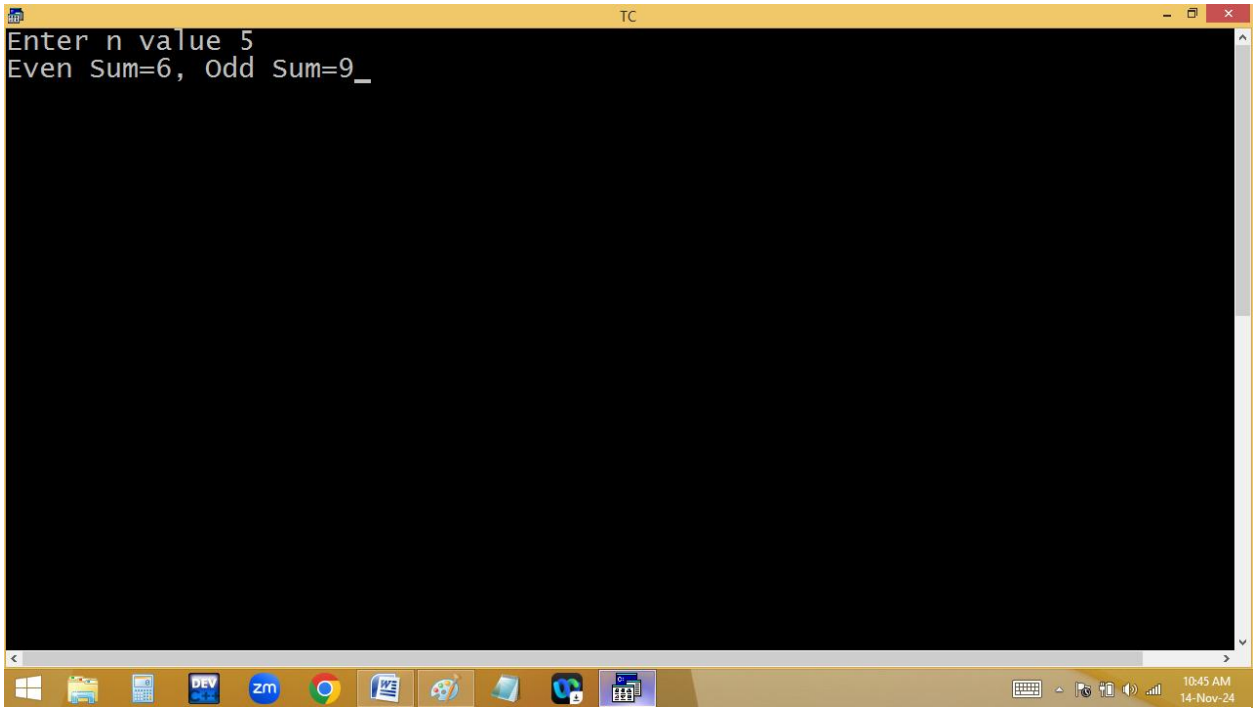


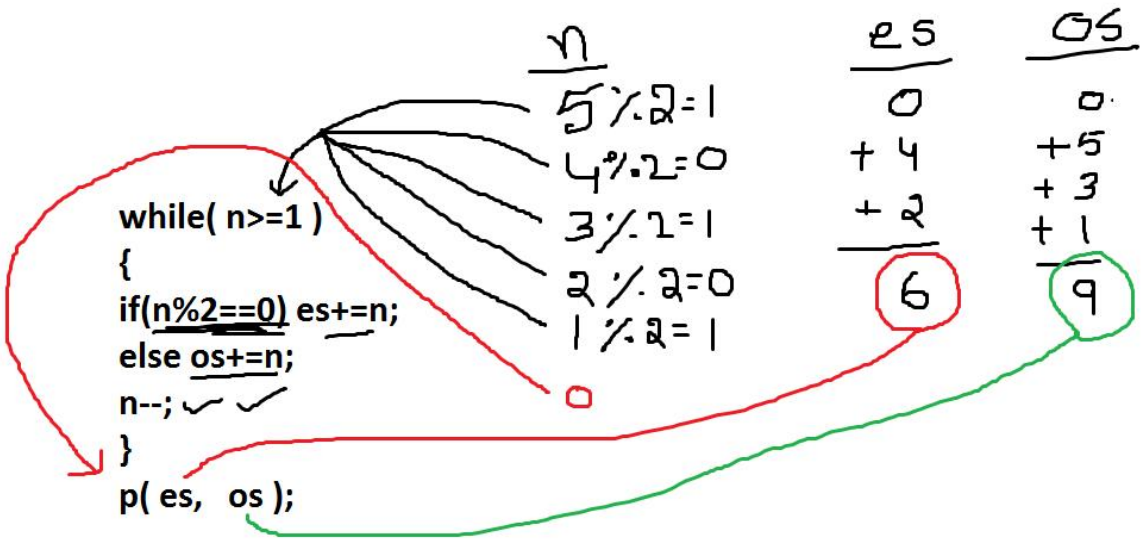
TC

File Edit Run Compile Project Options Debug Break/watch

Line 15 Col 1 Insert Indent Tab Fill Unindent * E:9AM.C

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,es=0,os=0;
clrscr();
printf("Enter n value "); scanf("%d",&n);
while( n>=1 )
{
if(n%2==0)es+=n;
else os+=n;
n--;
}
printf("Even Sum=%d, Odd Sum=%d",es,os);
getch();
}
```


TC
Enter n value 5
Even Sum=6, Odd Sum=9_


while(n>=1)
{
if(n%2==0) es+=n;
else os+=n;
n--;
}
p(es, os);
n
5 % 2 = 1
4 % 2 = 0
3 % 2 = 1
2 % 2 = 0
1 % 2 = 1
0
es
0
+ 4
+ 2
6
os
0
+5
+ 3
+ 1
9