

## C-24 $\Rightarrow$ Formatted Output Function in C

printf()

\* stdio.h

$\rightarrow$  Standard input/output.

\* Standard input means, when we can take input from user only through keyboard but not through printer or any other

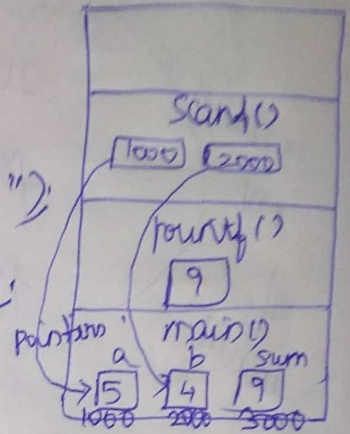
\* standard output means, we can see output in only monitor.

\* printf ("Control string", arg1, arg2 .... argn);

\* printf ("only Message");

printf ("Hello world");

```
int a, b, sum;
printf ("Enter 2 numbers:");
scanf ("%d %d", &a & b);
sum = a + b;
printf ("%d", sum);
```



// ~~scanf~~ printf ("sum = %d", sum);

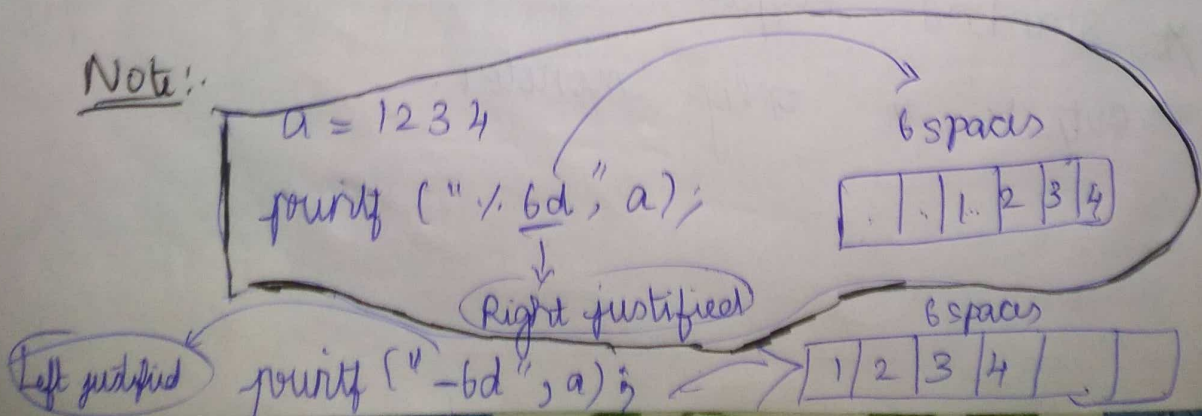
// printf ("a = %d, b = %d, sum = %d", a, b, sum);

(a = 5    b = 4    sum = 9)

// printf ("a = %d\n b = %d\n sum = %d\n", a, b, sum);

a = 5  
b = 4  
sum = 9

Note:





`printf("%3d", a);`  $\Rightarrow$ 

1	2	3	4
---	---	---	---

  
`printf("%04d", a);`  $\Rightarrow$ 

0	0	1	2	3	4
---	---	---	---	---	---

  
`printf("%-06d", a);`  $\Rightarrow$ 

1	2	3	4	Blank	
---	---	---	---	-------	--

  
 $a = 1234.3456$

`printf("%10.2f", a);`  $\Rightarrow$  Right justified  
 10 spaces.  

				1	2	3	4	.	3	4
--	--	--	--	---	---	---	---	---	---	---

  
 2 places after decimal point.

`printf("%-10.2f", a);`  $\Rightarrow$  Left justified  
 10 spaces.  

1	2	3	4	.	3	4			
---	---	---	---	---	---	---	--	--	--

`printf("%.2f", a);`  $\Rightarrow$  1234.345600  
 after decimal point 2 digits precision.

`printf("%.2f", a);`  $a = 1234.3456$   
 5 spaces.  


3	4	.	3	4
---	---	---	---	---

 $\Rightarrow$  Wrong  
 2 spaces after decimal.  
 1234.34

`printf("%.e", a);`  
 $a = 1234.3456$   
 $e^{-1} \Rightarrow \frac{1}{10}$

## CODE 1:


```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 1 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      int a=1234;
7      printf("%6d",a);
8      /* 6 spaces of memory are assigned & printed right justified by
9       leaving first two spaces empty since we give only 4 digits */
10 }
11
```

 "D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5\_Jennys Lecture

```
1234
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

## CODE 2:


```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 2 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      int a=1234;
7      printf("%-6d",a);
8      /* 6 spaces of memory are assigned & printed left justified by
9       leaving last two spaces empty since we give only 4 digits */
10 }
11
```

 "D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5\_Jennys Lectures\IN & OU"

```
1234
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

## CODE 3:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 3 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      int a=1234;
7      printf("%3d",a);
8  /* Even though we give 4 digits and 3 spaces of memory are
9     assigned there is no limitations for formatted output
10    functions .
11    It will print the output as it...
12    It will only align or justify the output */
13 }
14
```

 "D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5\_Jennys Lectures\IN

1234

Process returned 0 (0x0) execution time : 0.016 s

Press any key to continue.

■

## CODE 4:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 4 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      int a=1234;
7      printf("%06d",a);
8      /* 6 spaces of memory are assigned & printed right justified
9      by leaving first two spaces filled with 0 since we give
10     only 4 digits */
11     //printf("%-06d",a);
12     /*6 spaces of memory are assigned & printed left justified
13     by leaving last two spaces empty but not with 0....|
14     }
15
```

```
"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\IN & OUT
001234
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
E
3
E
3
E
```

## CODE 5:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 5 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      float a=1234.3456;
7      printf("%10.2f",a);
8  /* 10 spaces of memory are allotted.
9     In floating, point is also included in memory.
10    with With point 9 values are there but
11    floating precession is given as .2f
12    So after decimal point 2 digits are precessed
13    So now 1234.34, with decimal point 7 values are there
14    So 7 values are printed within 10 memory spaces
15    Justified from right, 3 spaces are blank*/
16  }
```


"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5\_Jennys Lectures\IN

```
1234.35
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```



## CODE 6:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 6 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      float a=1234.3456;
7      printf("%-10.2f",a);
8  /* 10 spaces of memory are allotted.
9     In floating, point is also included in memory.
10    with With point 9 values are there but
11    floating precession is given as .2f
12    So after decimal point 2 digits are precessed
13    So now 1234.34, with decimal point 7 values are there
14    So 7 values are printed within 10 memory spaces
15    %-10.2f Justified from left, last 3 spaces are blank*/
16  }
17
```

 "D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5\_Jenny

1234.35

Process returned 0 (0x0) execution time : 0.047 s

Press any key to continue.

## CODE 7:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 7 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      float a=1234.3456;
7      printf("%010.2f",a);
8      /* 10 spaces of memory are allotted.
9       In floating, point is also included in memory.
10      with With point 9 values are there but
11      floating precession is given as .2f
12      So after decimal point 2 digits are precessed
13      So now 1234.34, with decimal point 7 values are there
14      So 7 values are printed within 10 memory spaces
15      %10.2f Justified from right, first 3 spaces are Zeros*/
16  }
```

"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5\_Jennys


0001234.35

Process returned 0 (0x0) execution time : 0.047 s

Press any key to continue.

## CODE 8:

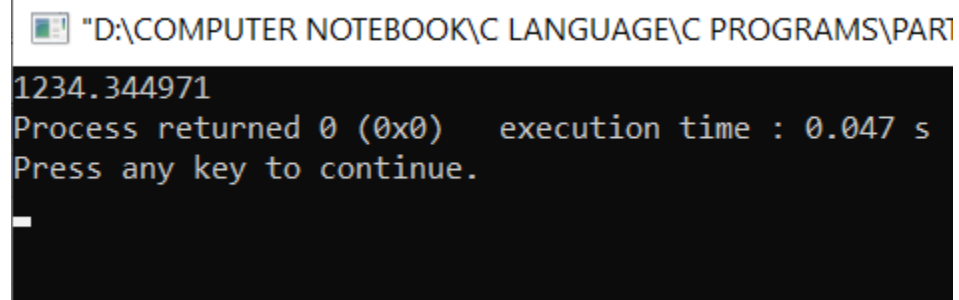
```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 8 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      float a=1234.3456;
7      printf("%-010.2f",a);
8      /* 10 spaces of memory are allotted.
9       In floating, point is also included in memory.
10      with With point 9 values are there but
11      floating precession is given as .2f
12      So after decimal point 2 digits are precessed
13      So now 1234.34, with decimal point 7 values are there
14      So 7 values are printed within 10 memory spaces
15      %-10.2f Justified from LEFT,
16      LAST 3 spaces are NOT Zeros because .2f is precessed */
17 }
```

 "D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5\_Jennys Le

```
1234.35
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

## CODE 9:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 9 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      float a=1234.345;
7      printf("%f",a);
8      /* Here float will precessed with 6 digits after decimal
9      point */
10 }
11
```

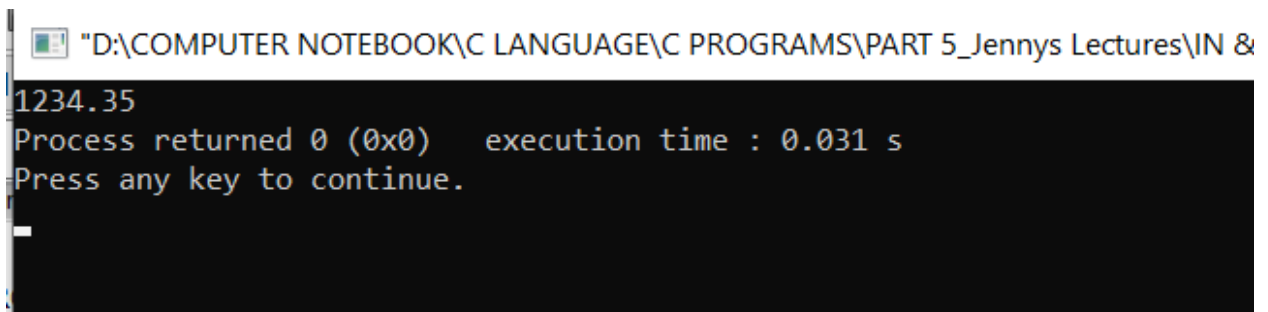


"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART..."

1234.344971  
Process returned 0 (0x0) execution time : 0.047 s  
Press any key to continue.

## CODE 10:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 10 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      float a=1234.3456;
7      printf("%5.2f", a);
8      /* Even though we 5 memory spaces are allotted, it will
9      print all the values but it will only justify since
10     it is formatted output not formatted input function */
11
12 }
13
```



"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5\_Jennys Lectures\IN &

1234.35


Process returned 0 (0x0) execution time : 0.031 s

Press any key to continue.



## CODE 11:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /* 11 FORMATTED OUTPUT FUNCTIONS */
4  int main()
5  {
6      float a=1234.3456;
7      printf("%e", a);
8  }
9
```

 "D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5\_Jennys Lectures\IN 8

1.234346e+003

Process returned 0 (0x0) execution time : 0.047 s

Press any key to continue.