Computer Programming & Problem Solving

CS100

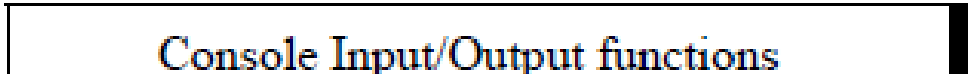
Mrs Sanga G. Chaki
Department of Computer Science and Engineering
National Institute of Technology, Goa
January, 2023

Console Input/Output

1. C language inherently has no provision for receiving/sending data from/to any of the input/output devices.
2. But we use printf() and scanf() functions – right?
3. Well, these are just that - standard I/O functions – that the C compiler developers wrote and stored in libraries.
4. I/O facilities might be OS dependent.
5. Standard I/O functions are of following types
    a) Console – Input from keyboard, output to monitor
    b) File – I/O from/to disks

Console Input/Output

1. Console = Screen + Keyboard
2. Formatted and unformatted I/O



Formatted functions

Type	Input	Output
char	scanf( )	printf( )
int	scanf( )	printf( )
float	scanf( )	printf( )
string	scanf( )	printf( )

Unformatted functions

Type	Input	Output
char	getch( ) getche( ) getchar( )	putch( ) putchar( )
int	-	-
float	-	-
string	gets( )	puts( )

printf()

1. General format:
    a) printf ("format string", list of variables) ;
2. The format string can contain:
    a) Characters that are simply printed as they are
    b) Conversion specifications that begin with a % sign
    c) Escape sequences that begin with a \ sign
3. How printf() works – we all know.

printf()

```
main( )
{
      int  avg = 346 ;
      float  per = 69.2 ;
      printf ( "Average = %d\nPercentage = %f", avg, per ) ;
}
```

1. **It examines the format string from left to right.**
2. **As long as no % or \ is encountered, it dumps the characters**
3. **When it comes across a conversion specification (%) in the format string, it picks the first variable the variables list and prints its value in the specified format**
4. **Similarly, when an escape sequence is met it takes the appropriate action**

Format Specifiers

Data type		Format specifier
Integer	short signed	%d or %I
	short unsigned	%u
	long singed	%ld
	long unsigned	%lu
	unsigned hexadecimal	%x
	unsigned octal	%o
Real	float	%f
	double	%lf
Character	signed character	%c
	unsigned character	%c
String		%s

Specifiers

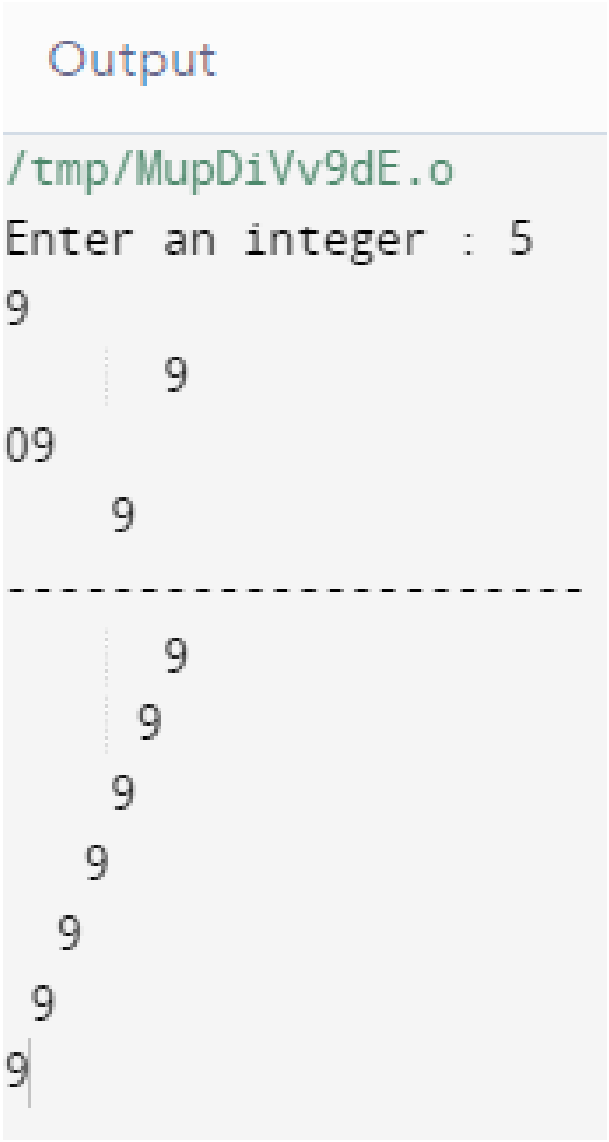
1. We can also specify
    a) Field width
    b) Precision

```
printf ( "\n%f %f %f", 305.0, 1200.9, 3005.3 ) ;
printf ( "\n%10.1f %10.1f %10.1f", 305.0, 1200.9, 3005.3 );
printf ( "\n%20s%20s", firstname1, surname1 ) ;
```

Specifiers - Example

```c
#include<stdio.h>
int main(){
int a=9; // Number to print
int b; // Variable spaces
printf("\n Enter an integer : "); // Scan 'b'
scanf("%d",&b);
printf("\n%d",a); // Normal print
printf("\n%7d",a); // Print in 7 spaces
printf("\n%02d",a); // Print in 2 spaces - 0's for blank spaces
printf("\n%*d",b,a); // Print in 'b' spaces
printf("\n----------------------");
printf("\n%7d",a);
printf("\n%6d",a);
printf("\n%5d",a);
printf("\n%4d",a);
printf("\n%3d",a);
printf("\n%2d",a);
printf("\n%d",a);
}
```

Specifiers - Example



Escape Sequences

1. The backslash symbol (\) is considered as an ‘escape’ character
2. It causes an escape from the normal interpretation of a string,
3. So that the next character is recognized as one having a special meaning.

Esc. Seq.	Purpose	Esc. Seq.	Purpose
\n	New line	\t	Tab
\b	Backspace	\r	Carriage return
\f	Form feed	\a	Alert
\’	Single quote	\”	Double quote
\\	Backslash		

scanf()

1. General format:
    a) scanf ("format string", list of addresses of variables) ;
2. The values that are supplied through the keyboard must be separated by either of these escape sequences
    a) blank(s),
    b) tab(s), or
    c) newline(s).
3. Do not include these in the format string for scanf().
4. All the format specifications that we learnt in printf( ) function are applicable to scanf( ) function as well.