LECTURE-05

Programming in C

Covered Topics

Characters Set ✓

Tokens & Identifiers ✓

Literals / Constants ✓

Datatypes in C ✓

✓ ☆ Input / output statements → Today

Output Statement :

↳ printf ( )

Output : CPU → Output Device

↳ Monitors, Printer, Speakers etc

* Printf is a library function that allows us to display the output.

→ This function is defined in the <stdio.h> header file.

Syntax :

printf (Statement 1, Statement 2 ....)

Statement 1 ⇓ formatted

Statement 2 ⇓ formatted

* printf ( ) ← Consider as String.

Ex printf ("Hello"); → Print कर दे Screen पे 'Hello'

"Hello" ← String

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

← Stdio = Standard Input / output

```
int main ( ) {
    printf ( "Hello World") ;
    return 0;
}
```

← Semi colomn is Compulsory

stdio.h

* filename . fileformat

filename ↓ Title of file

fileformat ↓ Type of file

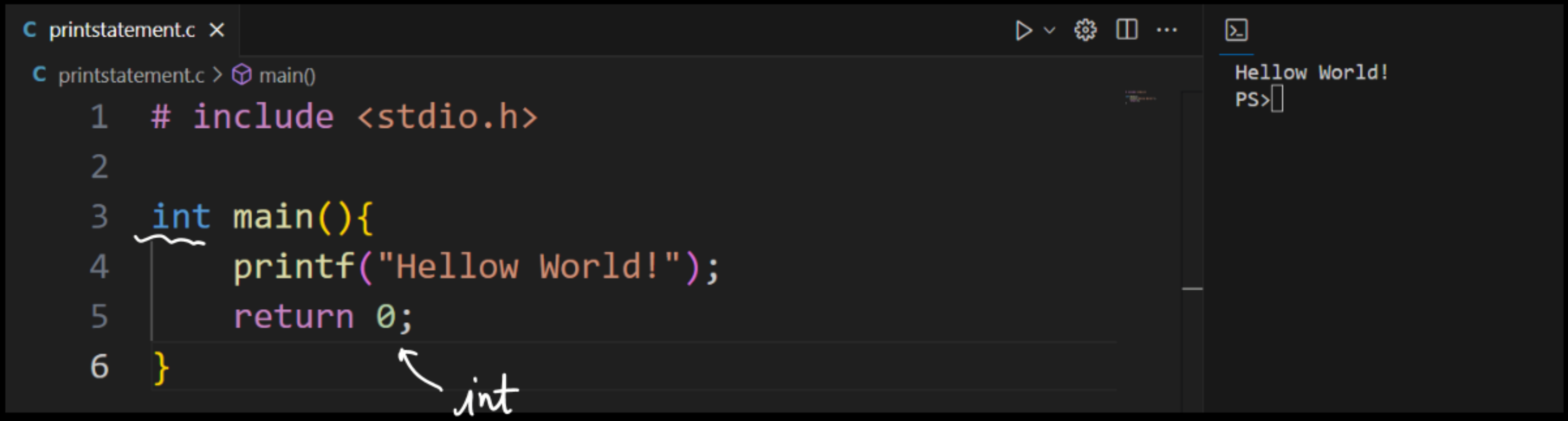
image.jpg

video.mp4

Program . c

Program: No spaces allowed

c ← C - program



```
# include <stdio.h>

int main(){
    printf("Hellow World!");
    return 0;
}
```

int ← 0

```
Hellow World!
PS>
```

```
Print ("Hello");
```

function Name

Argument

* Any function in C, can be identified by ( ) Parantheses (round bracket)

written for a specific task

Eg

```
printf( 12 );
```

integer literal

format Specifier

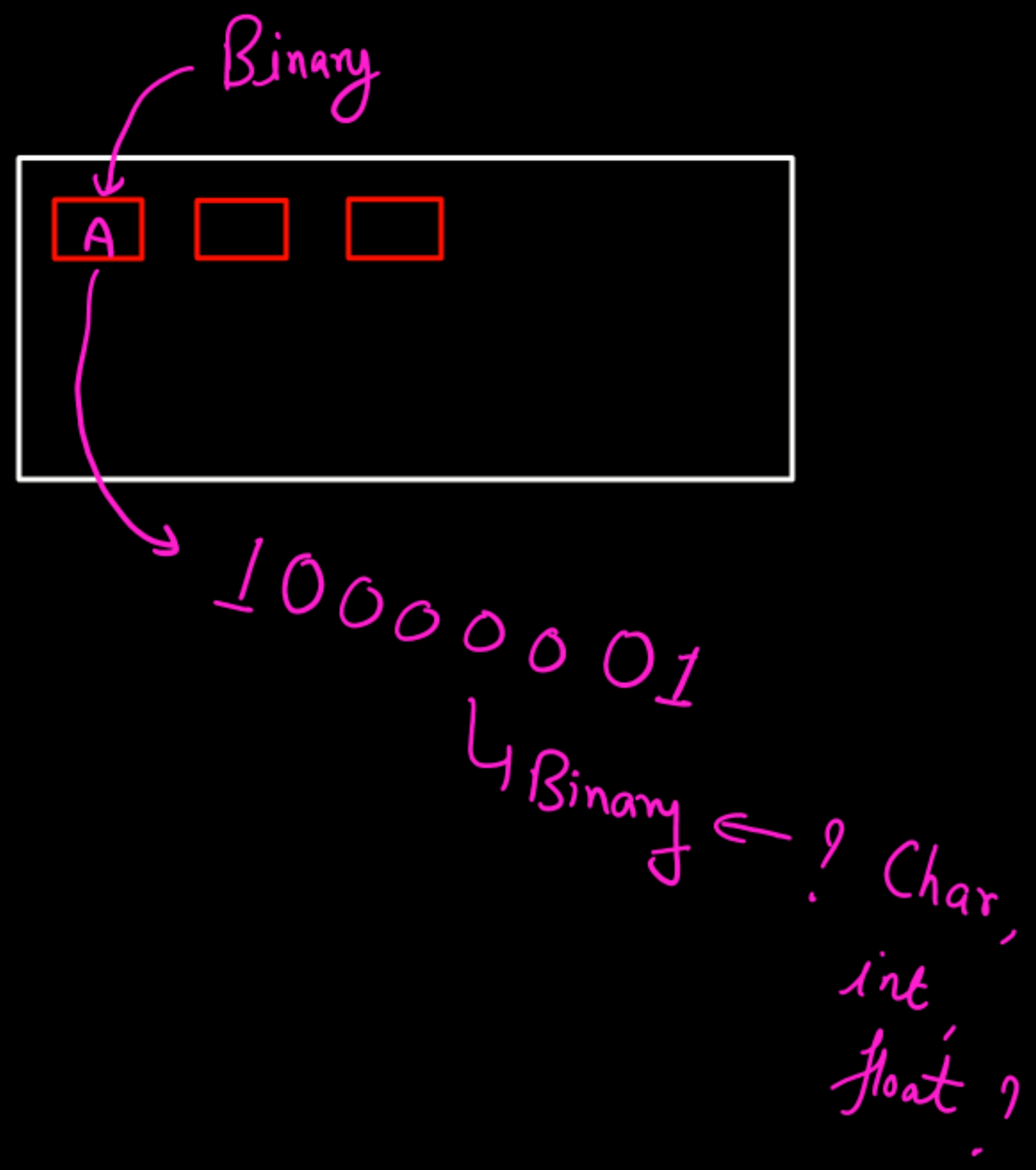
* Except String, we have to specify the type of data inside printf()

Format Specifiers:

↳ It specify the format of data.

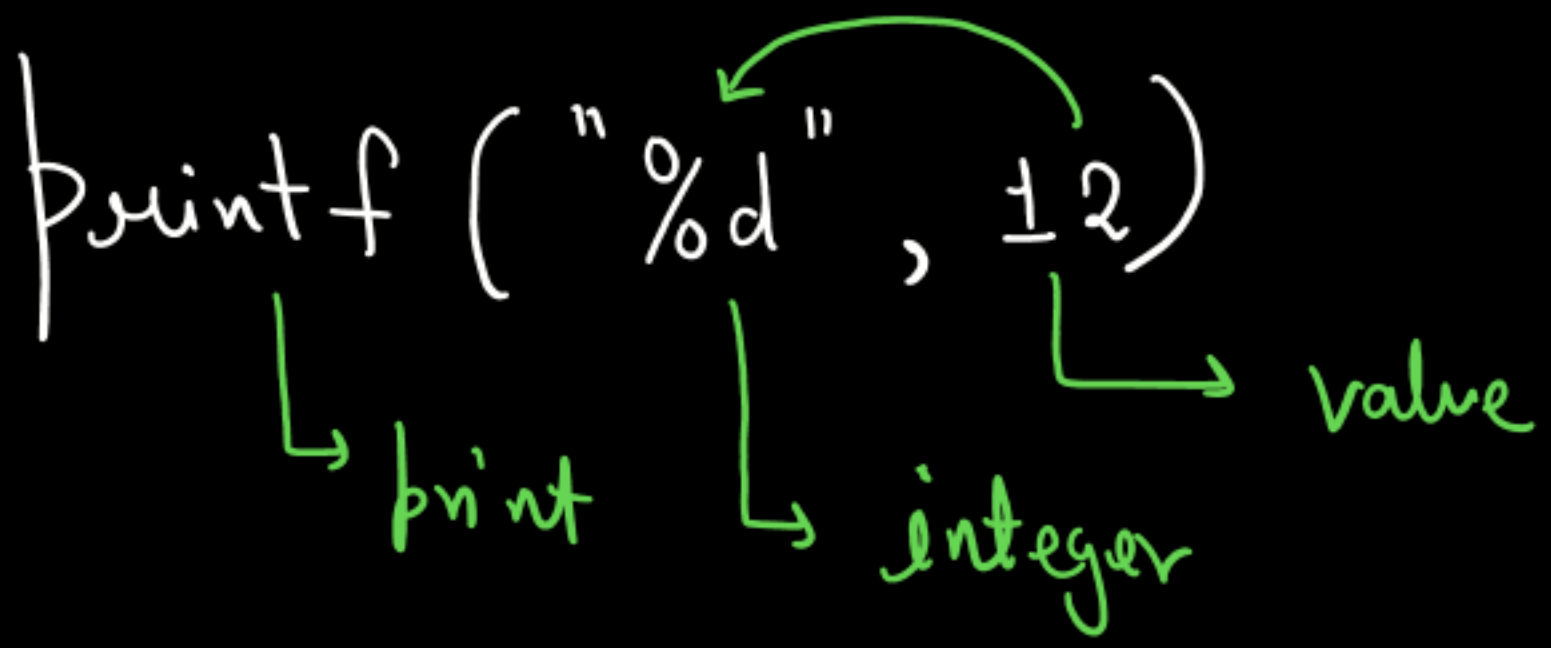
* It represents in the form of string.
↳ " "

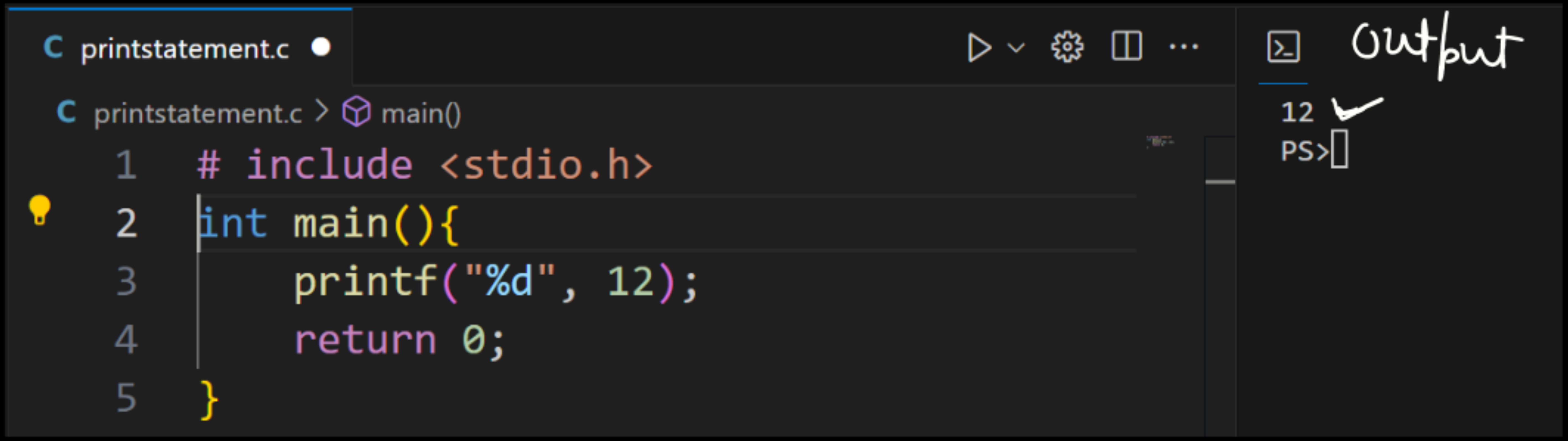
Specifier	Description
"%d"	integer (signed)
"%i"	decimal integer (signed)
"%u"	Unsigned decimal integer
"%f"	floating point Number
"%lf"	double (long float)
"%c"	Single Character



Specifier	Description	
'%s'	String (default)	
"%x"	Unsigned Hexadecimal integer	same
"%X"	— same —	
"%o"	Unsigned Octal Number (integer)	
"%p"	pointer address	
"%e", "%E"	Scientific Notation (3.1E25) →	3.1×10^{25}
"%%"	'%' print % literal	
"%.2f"	Print float upto 2 decimal Places	
	Ex ("%.2f", 3.14159)	
	3.14	

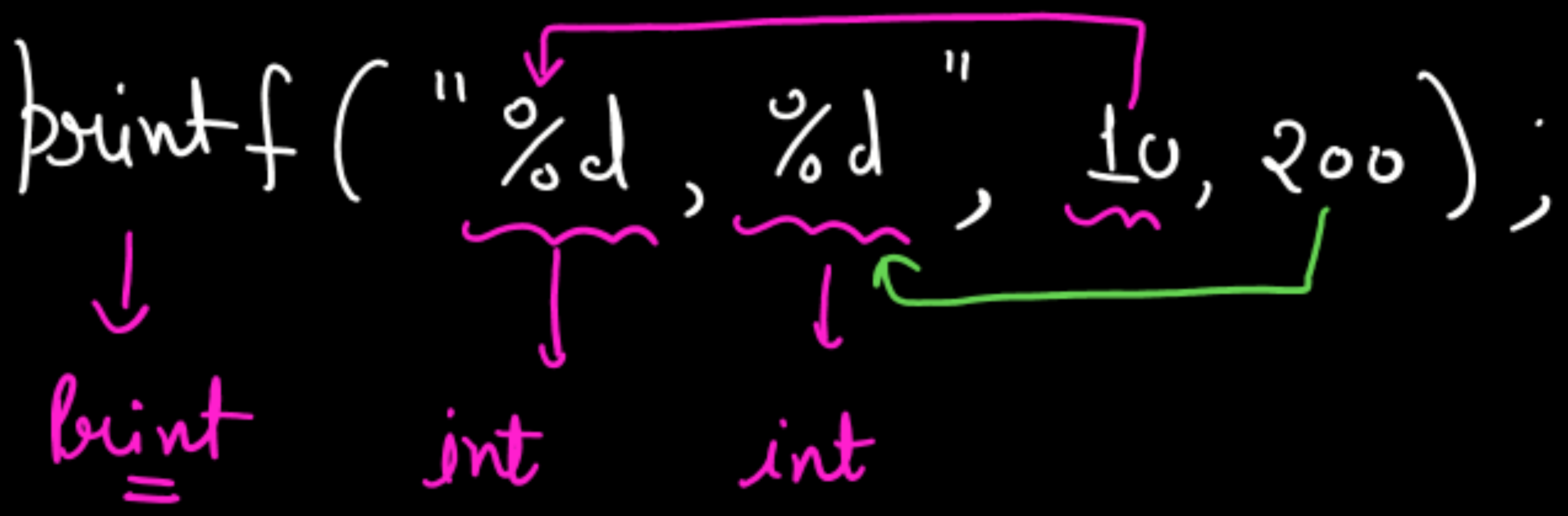
Use of format Specifiers in Print function

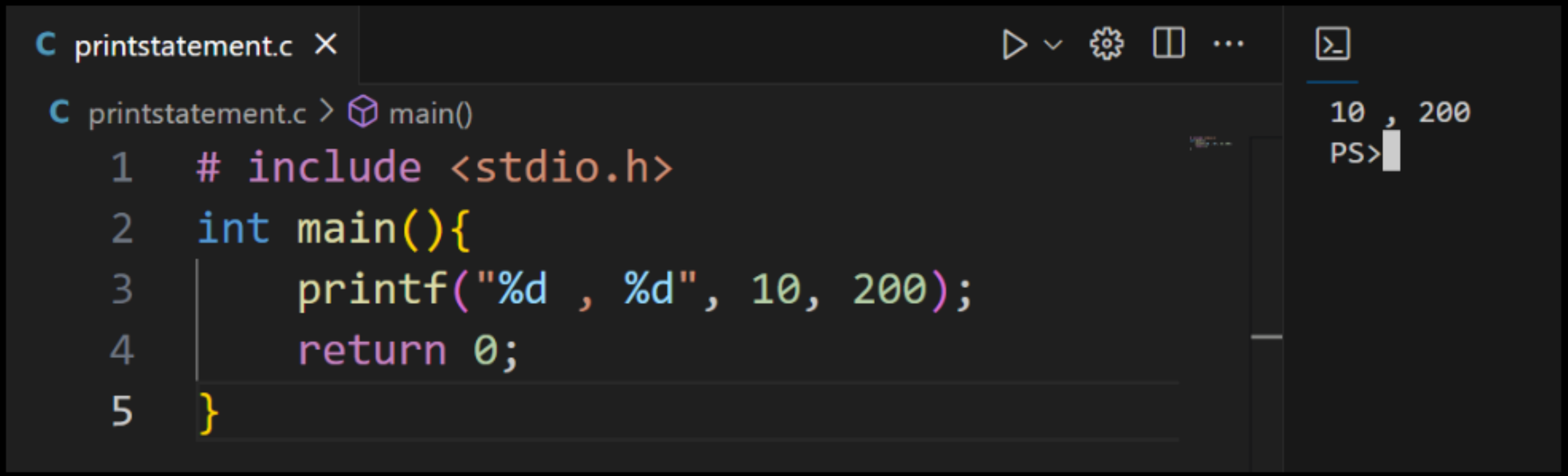




```c
# include <stdio.h>
int main(){
    printf("%d", 12);
    return 0;
}
```

```
12 ✓
PS>
```





```c
# include <stdio.h>
int main(){
    printf("%d , %d", 10, 200);
    return 0;
}
```

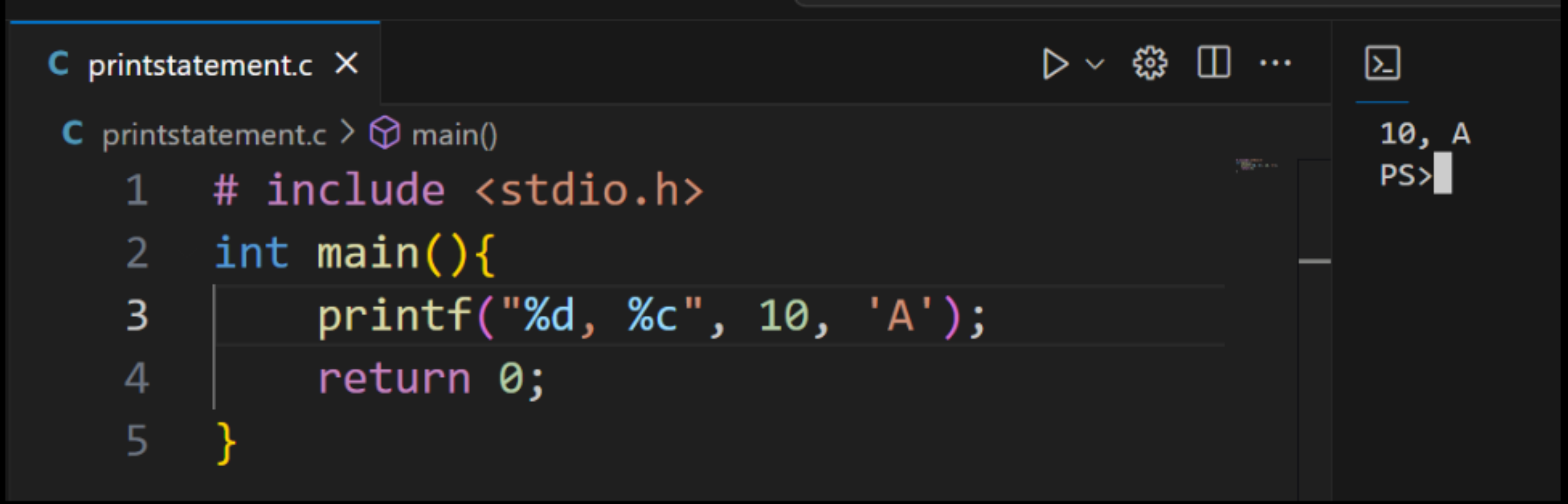
```
10 , 200
PS>
```

Printf("%d, %c", 100, 'A');

print

int

Character



```c
# include <stdio.h>
int main(){
    printf("%d, %c", 10, 'A');
    return 0;
}
```

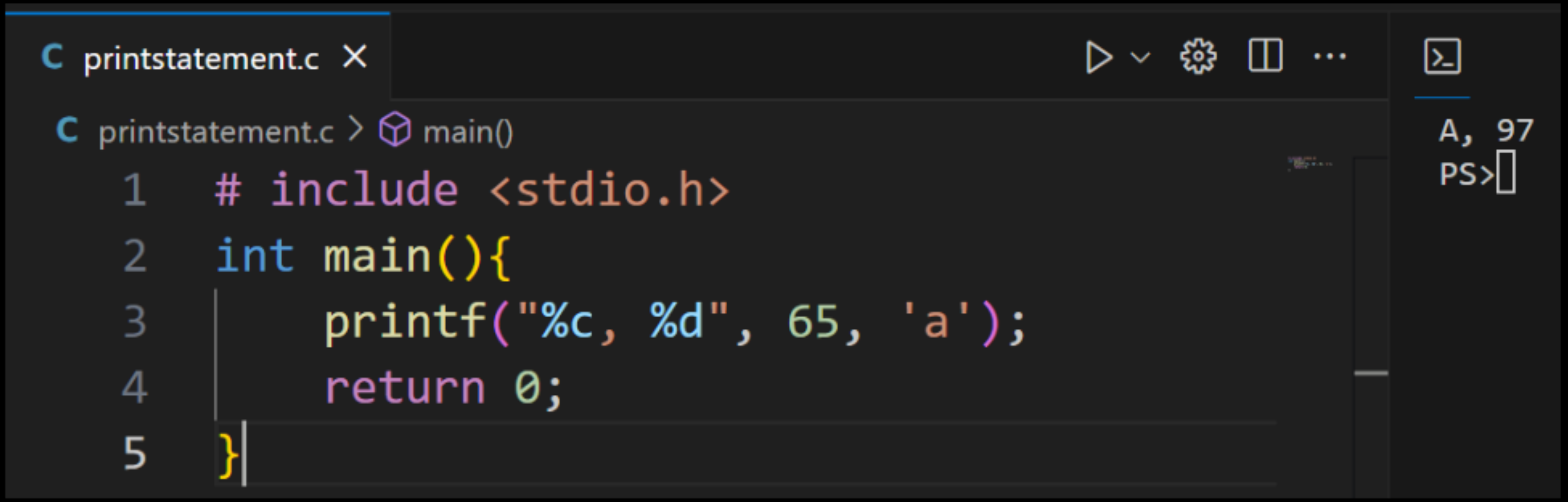
```
10, A
PS>
```

printf ("%c", "%d", 65, 'a');

print

Char

int



```c
# include <stdio.h>
int main(){
    printf("%c, %d", 65, 'a');
    return 0;
}
```

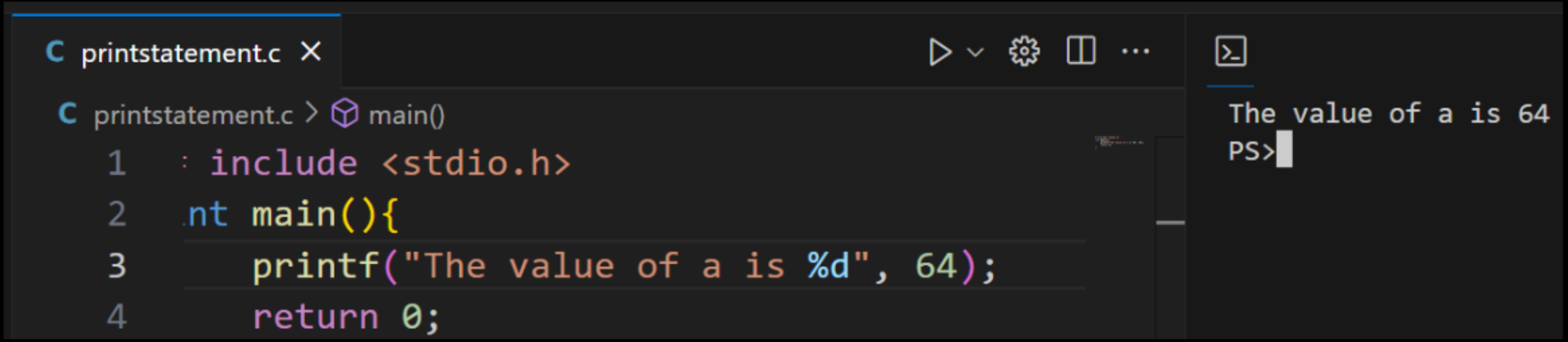
```
A, 97
PS>
```

printf ("The value of a is %d", 64);

print

int

The value of a is 64



```c
: include <stdio.h>
nt main(){
   printf("The value of a is %d", 64);
   return 0;
```

```
The value of a is 64
PS>
```

Taking user input :

& ← Address of

scanf ( ) ← used to take the input from user.

↳ scanf ("format specifier", & variable);

Ex

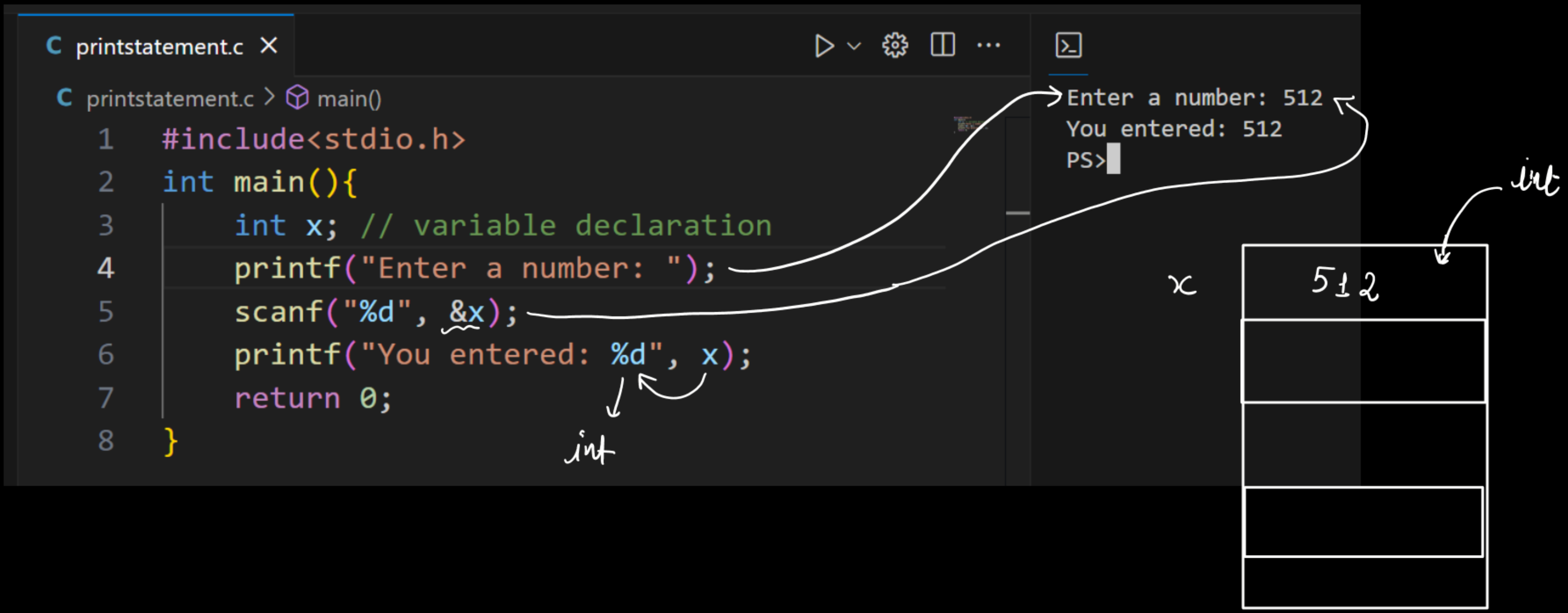
```
#include <stdio.h>
int main(){
    int x;
    prinf ("Enter the number");
    scanf ("%d", &x);
    printf("You Entered %d", x);
    return 0;
}
```

⇓ take integer input

Store at the address of x

↓ print You Entered x

Program to take an input from user & print the value :



printstatement.c

printstatement.c > main()

```c
#include<stdio.h>
int main(){
    int x; // variable declaration
    printf("Enter a number: ");
    scanf("%d", &x);
    printf("You entered: %d", x);
    return 0;
}
```

```
Enter a number: 512
You entered: 512
PS>
```

printstatement.c
C printstatement.c > ...

```c
// Write a program to add two numbers
// but take the numbers as input

#include <stdio.h>

int main(){
    int a, b;
    printf("Enter the value of a: ");
    scanf("%d", &a);
    printf("Enter the value of b: ");
    scanf("%d", &b);
    int c;
    c = a + b;
    printf("The sum of %d and %d is %d", a, b, c);
    return 0;
}
```

```
Enter the value of a: 250
Enter the value of b: 500
The sum of 250 and 500 is 750
PS>
```

C = 250 + 500
int
a
int
b
int
c
int
a
250
b
500
int
c
250 + 500
⇒ 750