Programming in C

Lecture - 06 :

Format Specifiers & Escape Sequence Characters

- %c
- %d

Comments in C

↳ Comments are human readable text that are ignored/skipped during program execution.

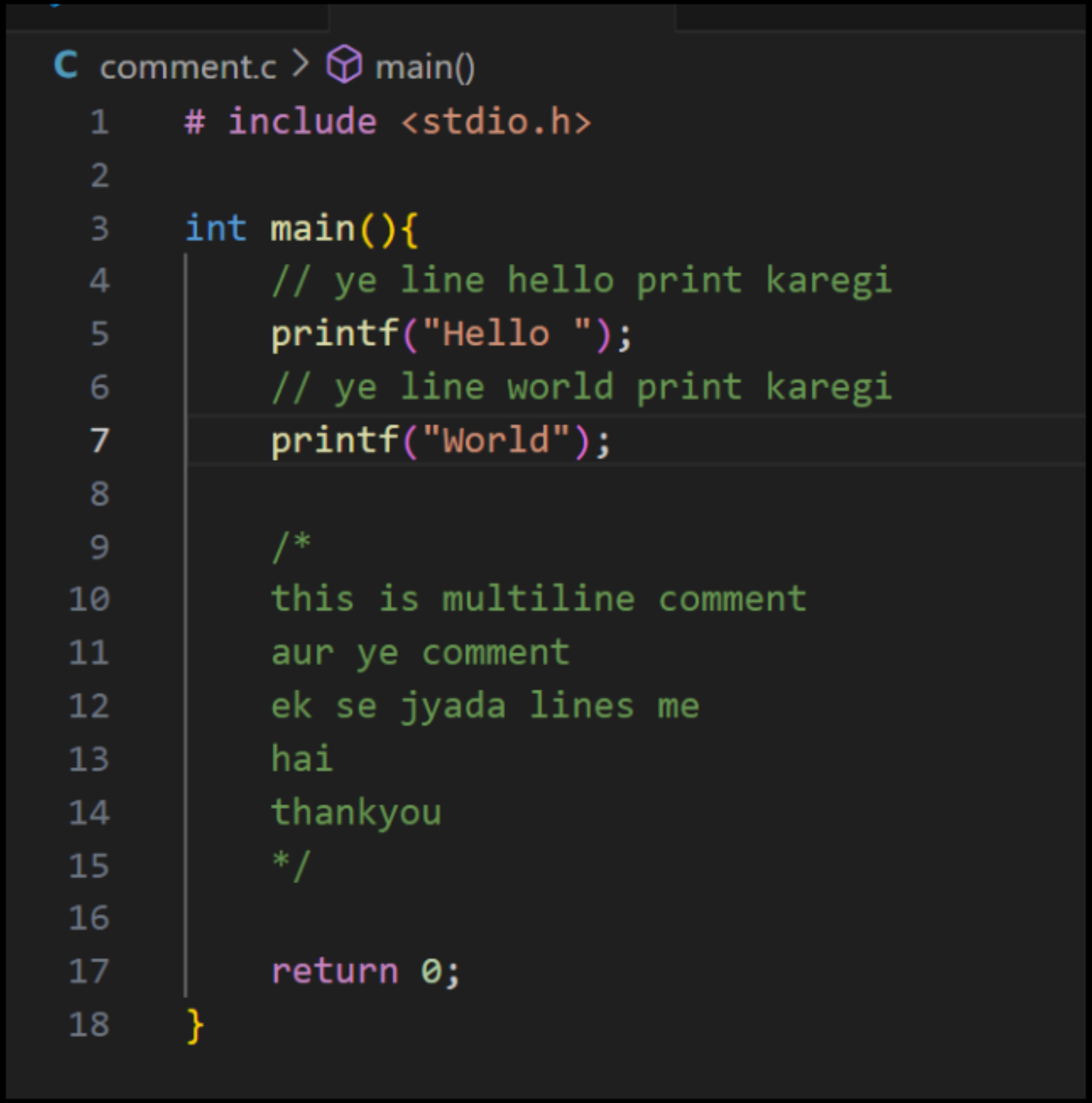
* Comments are used to explain the code, to make it more readable.

a) Single line Comment :

↳ // is used

b) Multiline Comment :

↳ /* Multi
line
Comment */



```c
# include <stdio.h>

int main(){
    // ye line hello print karegi
    printf("Hello ");
    // ye line world print karegi
    printf("World");

    /*
    this is multiline comment
    aur ye comment
    ek se jyada lines me
    hai
    thankyou
    */

    return 0;
}
```

Format Specifers:

%d → int

%c → Char

%i → int

%u → unsigned int

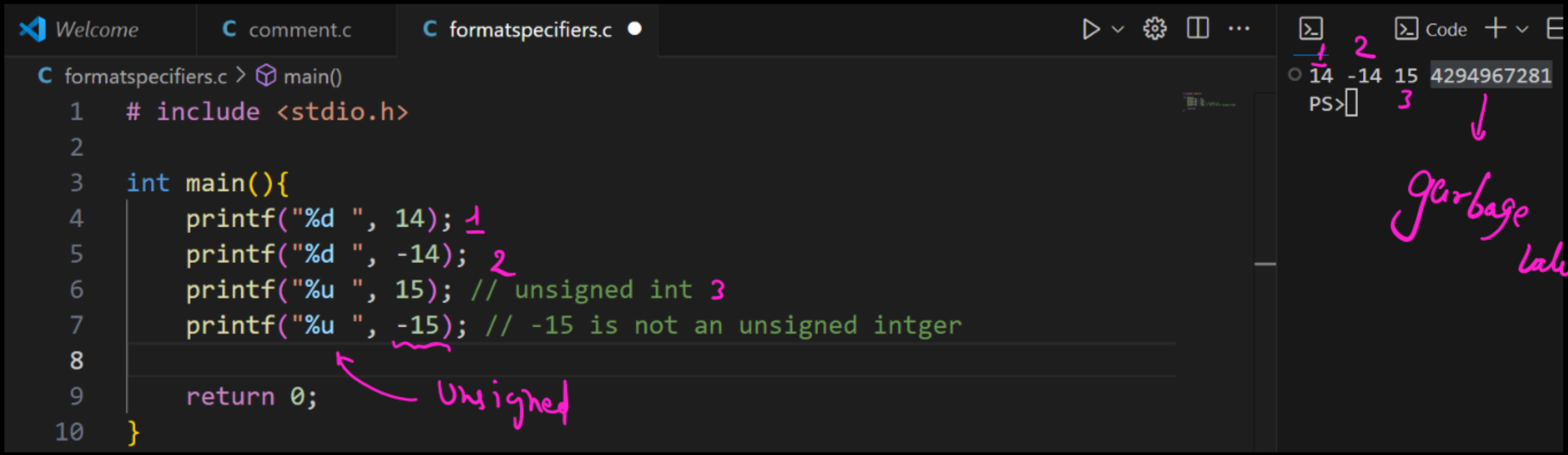
%f → float

⇓

%0.nf → float

Values after decimal point

%s → String

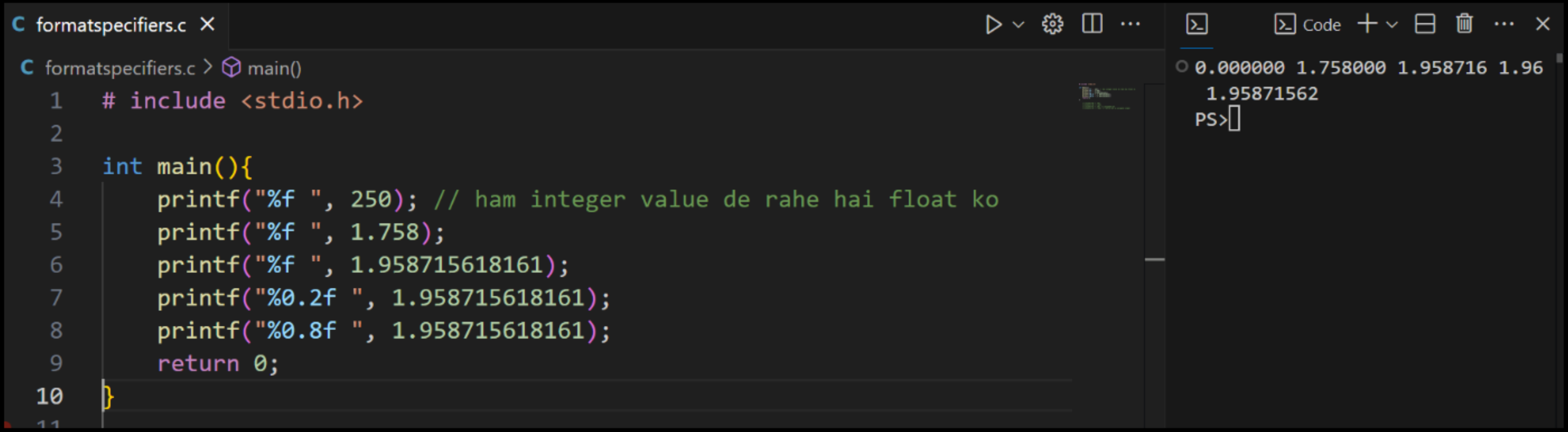


```c
# include <stdio.h>

int main(){
    printf("%d ", 14); 1
    printf("%d ", -14); 2
    printf("%u ", 15); // unsigned int 3
    printf("%u ", -15); // -15 is not an unsigned intger

    return 0;
}
```

Unsigned

```
1  2
14 -14 15 4294967281
PS>   3
```

garbage value



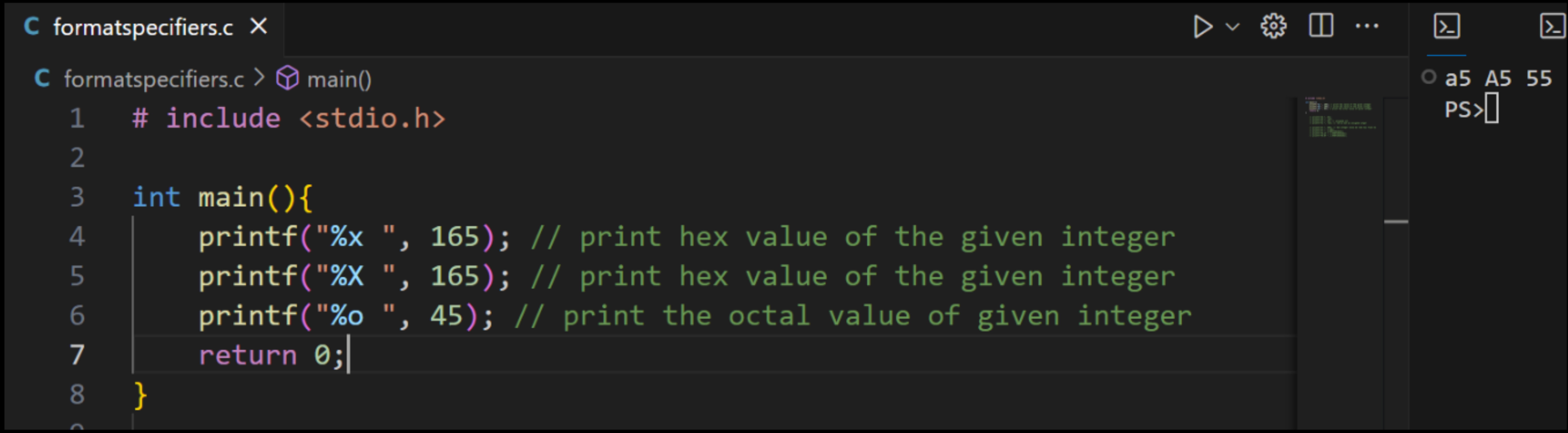
```c
# include <stdio.h>

int main(){
    printf("%f ", 250); // ham integer value de rahe hai float ko
    printf("%f ", 1.758);
    printf("%f ", 1.958715618161);
    printf("%0.2f ", 1.958715618161);
    printf("%0.8f ", 1.958715618161);
    return 0;
}
```

```
0.000000 1.758000 1.958716 1.96
 1.95871562
PS>
```

%x , %X → Hexadecimal int

%o → Octal value

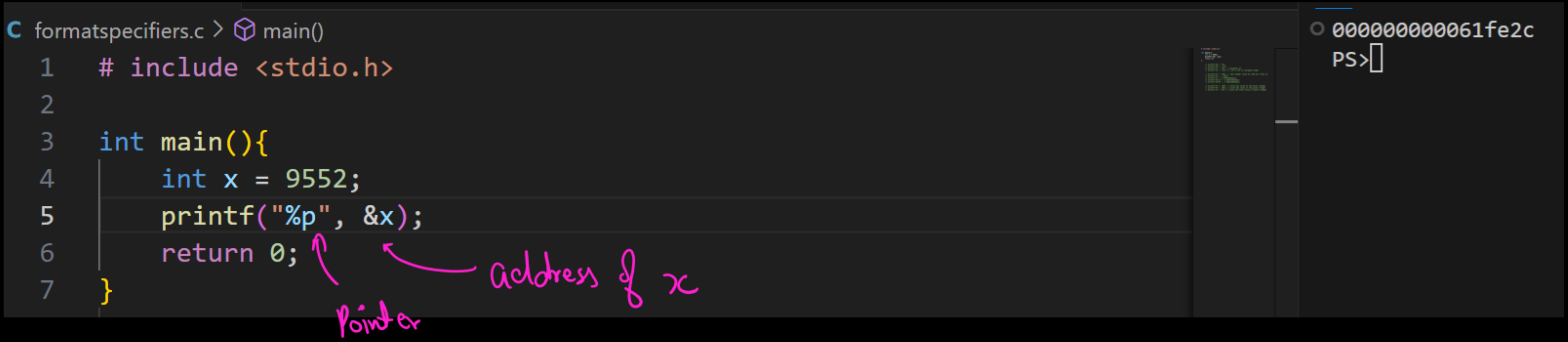
↳ both are unsigned



```c
# include <stdio.h>

int main(){
    printf("%x ", 165); // print hex value of the given integer
    printf("%X ", 165); // print hex value of the given integer
    printf("%o ", 45); // print the octal value of given integer
    return 0;
}
```

```
a5 A5 55
PS>
```

Pointer Value



```c
# include <stdio.h>

int main(){
    int x = 9552;
    printf("%p", &x);
    return 0;
}
```

```
000000000061fe2c
PS>
```

Escape Sequence characters :

printf (" $\underbrace{\%d}_{}$ "); → We Can not print the reserved characters of a String directly

↳ integer

printf (" $\underbrace{\%f}_{}$ ") ⟹ $\underbrace{\%}$ sign is reserved for specific purpose in a string

↳ float

⟹ To print these kind of special reserved character we use Escape Sequence Characters.

a) \\ ⟹ Insert a backslash

\ ← reserved

b) \' ⟹ Insert a single quotation mark

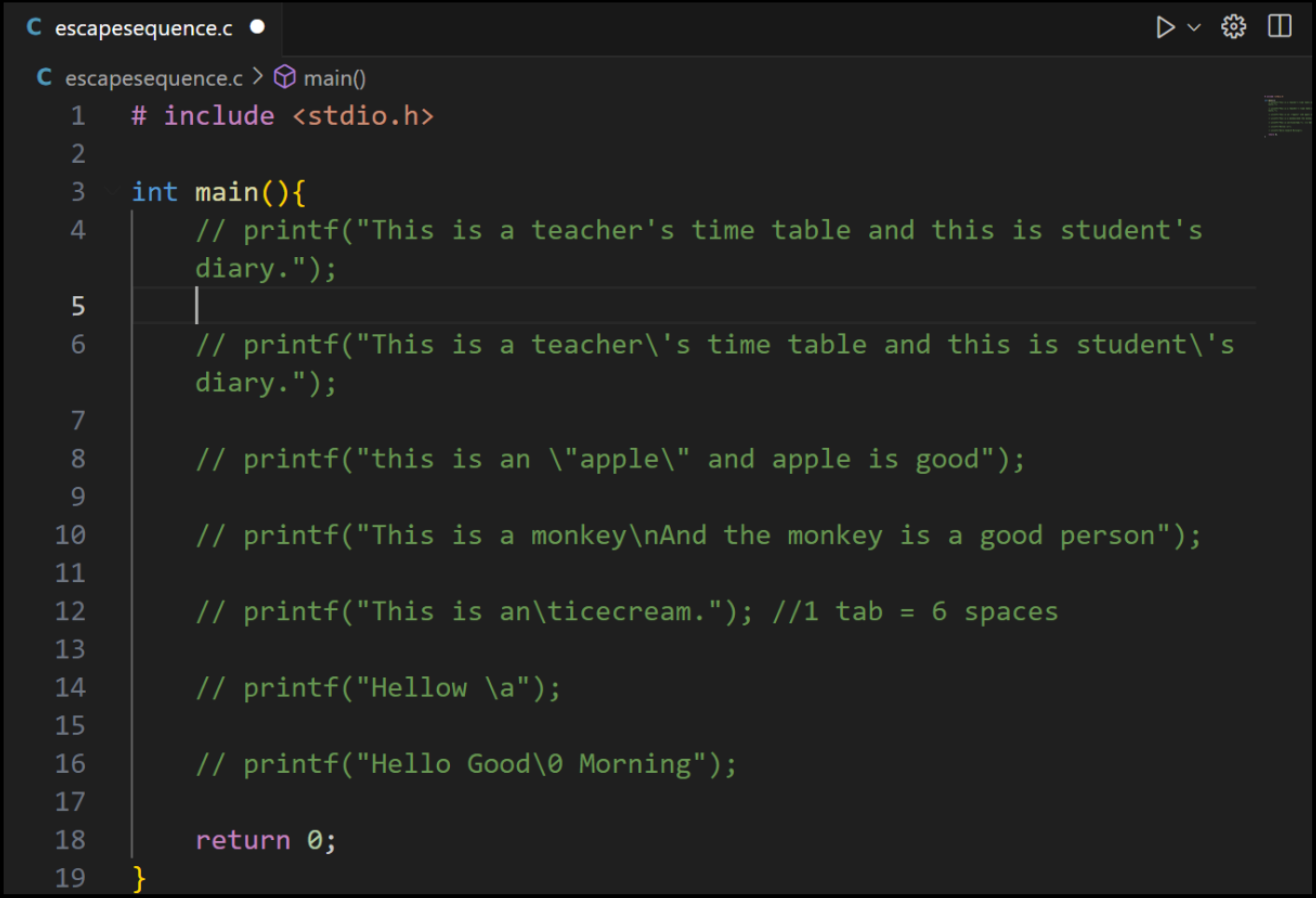
c) \" ⟹ Insert a double quotation mark

d) \n → Newline character

e) \t → give a tab space
  ↳ group of 4 spaces

f) \a → alarm bell (Alert Bell)

g) \0 → Null Character


escapesequence.c
escapesequence.c > main()

```c
# include <stdio.h>

int main(){
    // printf("This is a teacher's time table and this is student's diary.");
    
    // printf("This is a teacher\'s time table and this is student\'s diary.");

    // printf("this is an \"apple\" and apple is good");

    // printf("This is a monkey\nAnd the monkey is a good person");

    // printf("This is an\ticecream."); //1 tab = 6 spaces

    // printf("Hellow \a");

    // printf("Hello Good\0 Morning");

    return 0;
}
```

Cyclic Property of an integer

Range → Unsigned int → 0 to $2^n - 1$

↳ signed int → $[-2^{n-1}$ to $2^{n-1} - 1]$

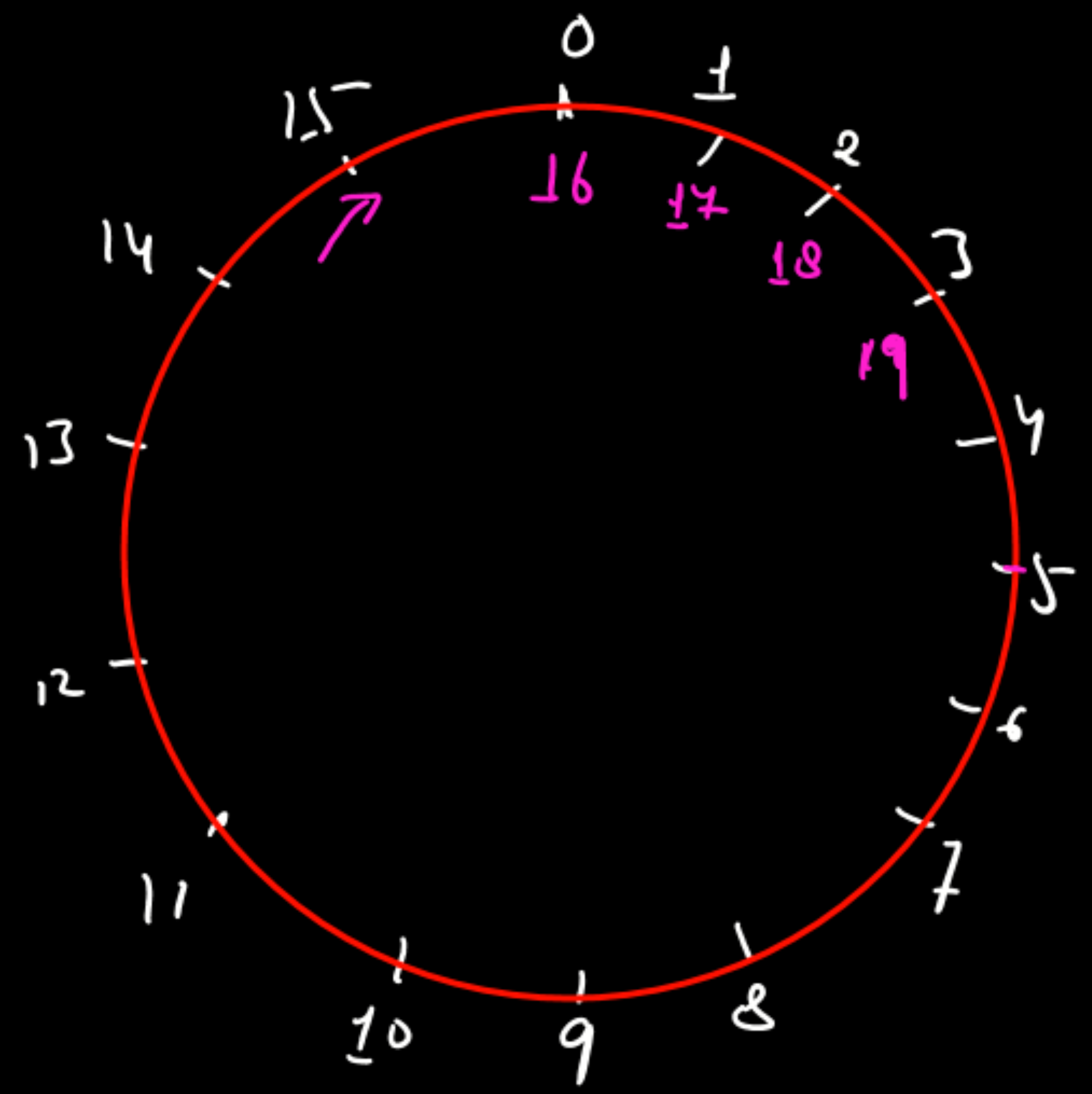
↳ 2's Complement form

Ex n = 4 bits (Unsigned)

Range = 0 to $2^4 - 1$ ⇒ 0 to 15

Suppose, we want to enter a number greater than 15.

out of range → Start from lower value.



n = 16 ⇒ 0

n = 17 ⇒ 1

n = 18 ⇒ 2

n = 19 ⇒ 3

⋮

$\textcircled{n}$ times

n = 32 → 0

n = 33 → 1

⋮

n = 4 bits (Signed)

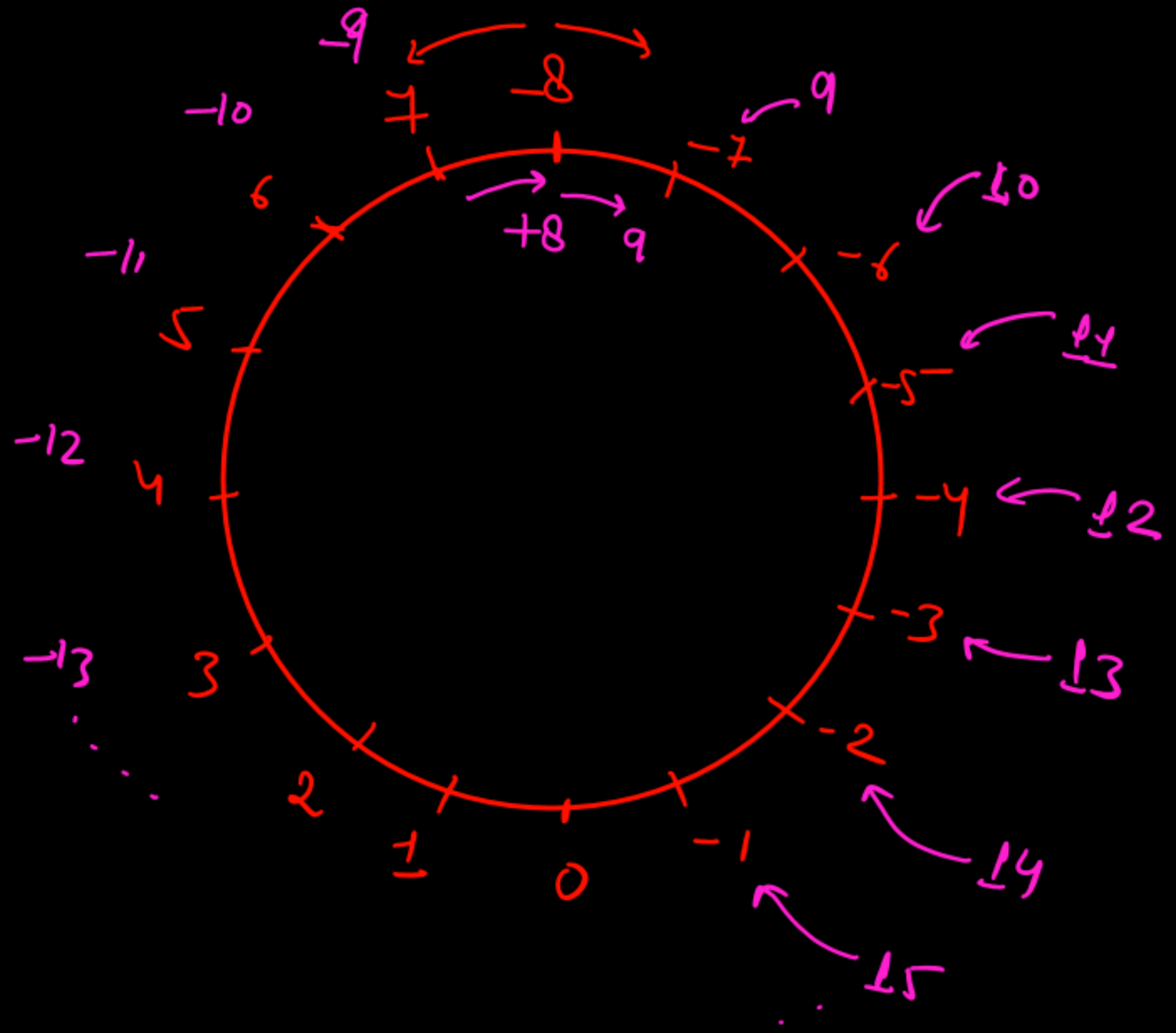
↓

2's Complement

Range = -2^{n-1} to $2^{n-1}-1$

R = -2^{4-1} to $2^{4-1}-1$

[-8 to +7]



i = 8 ⇒ -8

i = 9 ⇒ -7

i = 10 ⇒ -6

i = -9

↳ +7

* Important

Smallest to largest

→ Short = $\underline{\underline{2}}$ bytes ⇒ 16 bits

↳ unsigned ⇒ $R = 0$ to $2^{16} - 1$ ⇒ 0 to $\underset{\sim\sim}{65,535}$ ↙ max

↳ signed ⇒ $R = -2^{n-1}$ to $2^{n-1} - 1$

$R = -2^{15}$ to $2^{15} - 1$ ⇒ $\underset{\sim\sim}{-32768}$ to $\underset{\sim\sim}{32767}$

for unsigned short integer, the maximum value will be 65,535

$\underline{\underline{Eg}}$ $\left.\begin{array}{l} i = 65536 \rightarrow 0 \\ i = 65537 \Rightarrow 1 \end{array}\right\}$ warning

for signed integer, the maximum value will be 32767

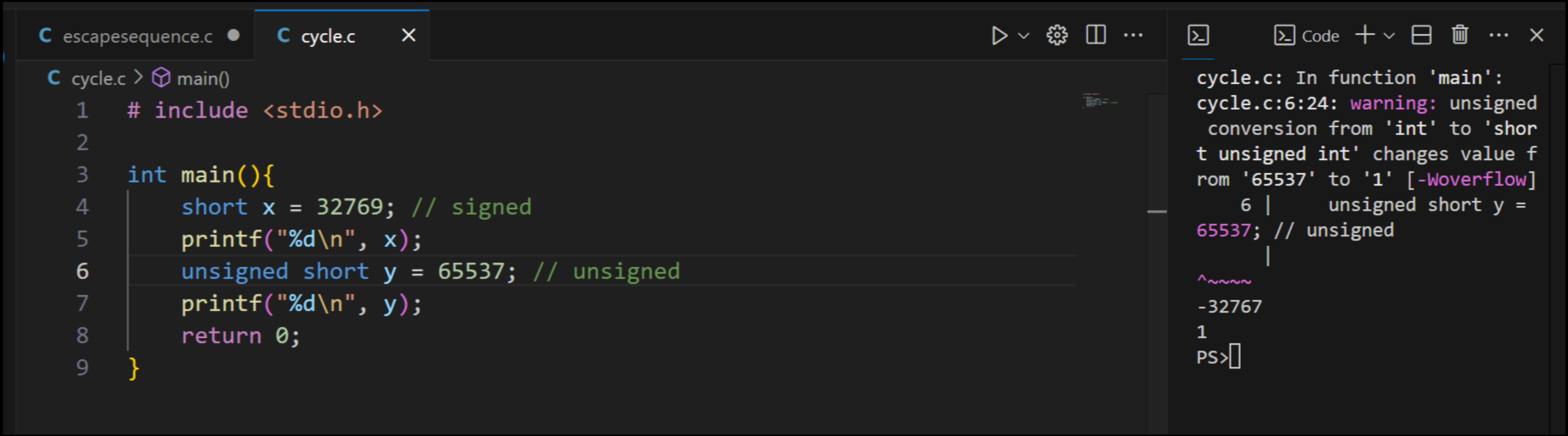
$i = 32768 \longrightarrow -32768$

$32769 \longrightarrow -32767$

⋮

[Repeat the number]

Practical Implementation:



```c
# include <stdio.h>

int main(){
    short x = 32769; // signed
    printf("%d\n", x);
    unsigned short y = 65537; // unsigned
    printf("%d\n", y);
    return 0;
}
```

```
cycle.c: In function 'main':
cycle.c:6:24: warning: unsigned
 conversion from 'int' to 'shor
t unsigned int' changes value f
rom '65537' to '1' [-Woverflow]
    6 |     unsigned short y =
65537; // unsigned
      |
^~~~~
-32767
1
PS>
```

tomorrow →
↳ Operators ← imp