

Mr C has a long floating name

ESC101: Foundations of Computing

Purushottam Kar

Announcements

- Special session on using computers
 - August 11, 2018 (coming Saturday), 5PM, NCL CC-02
 - Not a revision class – only for students who are new to computers
 - Thanks to Parth Sharma (tutor) for the suggestion and offer



Adding two unknown numbers

3



Adding two unknown numbers

3

HOW WE USUALLY SPEAK TO A HUMAN



Adding two unknown numbers

3

HOW WE USUALLY SPEAK TO A HUMAN



Adding two unknown numbers

3

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?



Adding two unknown numbers

3

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello



Adding two unknown numbers

3

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.



Adding two unknown numbers

3

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.



Adding two unknown numbers

3

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.



Adding two unknown numbers

3

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.



Adding two unknown numbers

3

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.

Goodbye



Adding two unknown numbers

3

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.

Goodbye



Adding two unknown numbers

3

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.

Goodbye



Adding two unknown numbers

3

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.

Goodbye



Adding two unknown numbers

3

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

3

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

3

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
    int a, b;  
    scanf("%d", &a);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

3

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
    int a, b;  
    scanf("%d", &a);  
    scanf("%d", &b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
    int a, b;  
    scanf("%d", &a);  
    scanf("%d", &b);  
    printf("%d", a + b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
    int a, b;  
    scanf("%d", &a);  
    scanf("%d", &b);  
    printf("%d", a + b);  
    return 0;
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
    int a, b;  
    scanf("%d", &a);  
    scanf("%d", &b);  
    printf("%d", a + b);  
    return 0;  
}
```

HOW WE USUALLY SPEAK TO A HUMAN

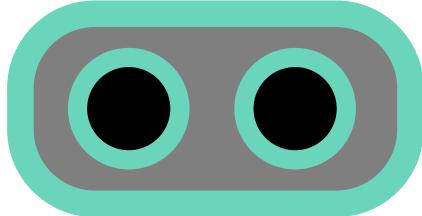
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
    int a, b;  
    scanf("%d", &a);  
    scanf("%d", &b);  
    printf("%d", a + b);  
    return 0;  
}
```



HOW WE USUALLY SPEAK TO A HUMAN

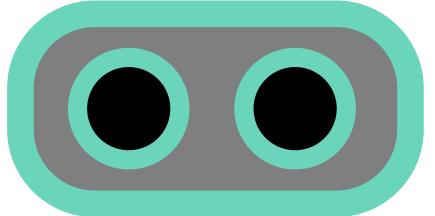
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;  
    a  
    scanf("%d", &a);  
    scanf("%d", &b);  
    printf("%d", a + b);  
  
    return 0;  
}
```



HOW WE USUALLY SPEAK TO A HUMAN

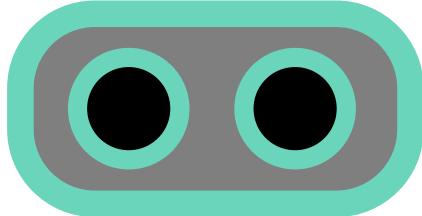
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    printf("%d", a + b);
    return 0;
}
```



a

b

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.

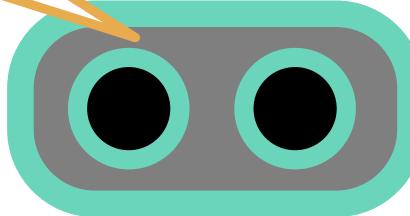
Goodbye



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    Please give
    me input + b);
    return 0;
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye



Adding two unknown numbers

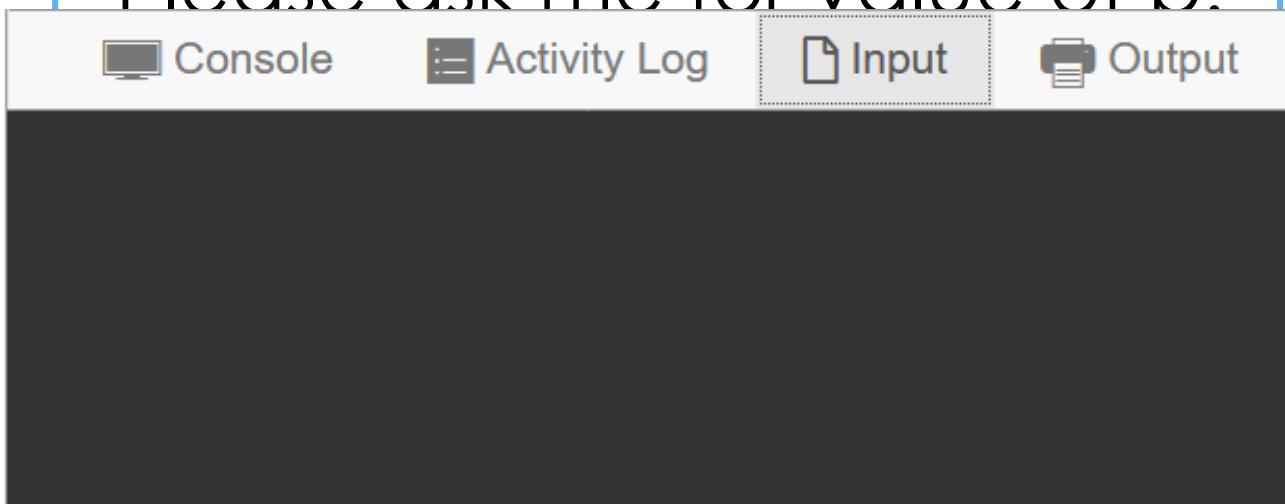
HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>

int main(){
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    Please give
    me input
    return a + b;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

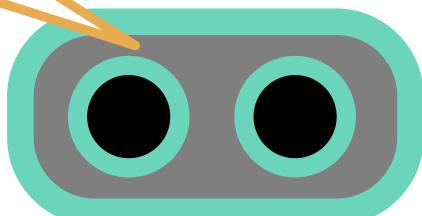
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.



Adding two unknown numbers

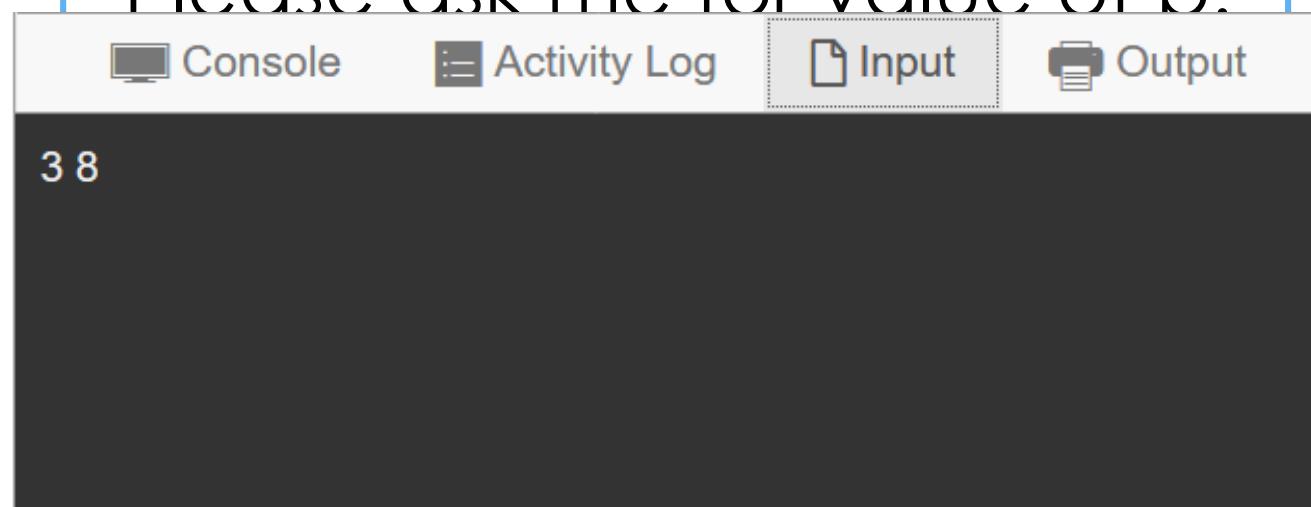
HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    Please give
    me input + b);
    return 0;
}
```



HOW WE USUALLY SPEAK TO A HUMAN

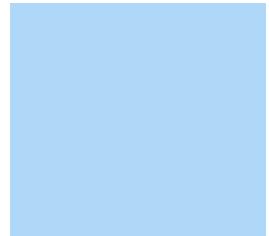
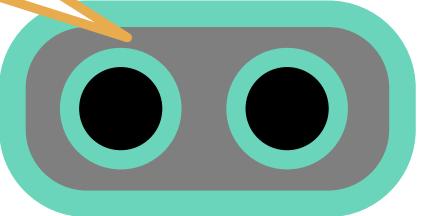
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

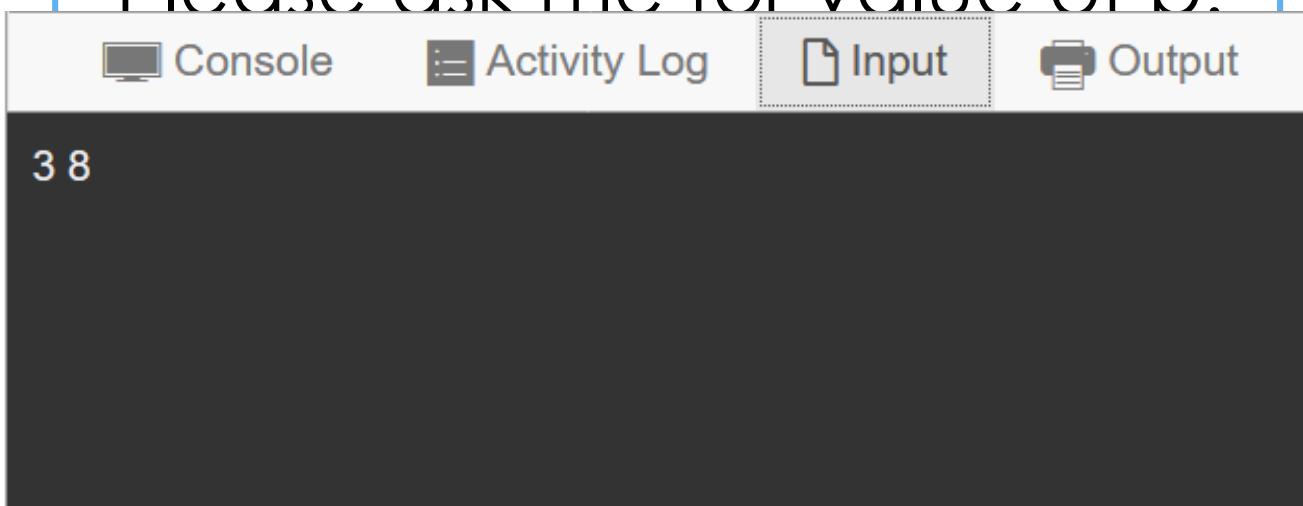
```
#include <stdio.h>
int main(){
    ...
    Thanks. Let me
    get back to work
    ...
    Please give
    me input
    return 0;
}
```



b

HOW WE USUALLY SPEAK TO A HUMAN

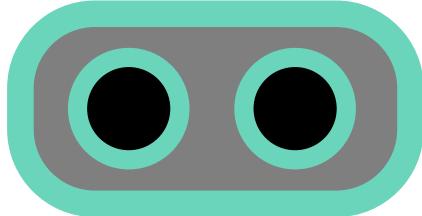
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.



Adding two unknown numbers

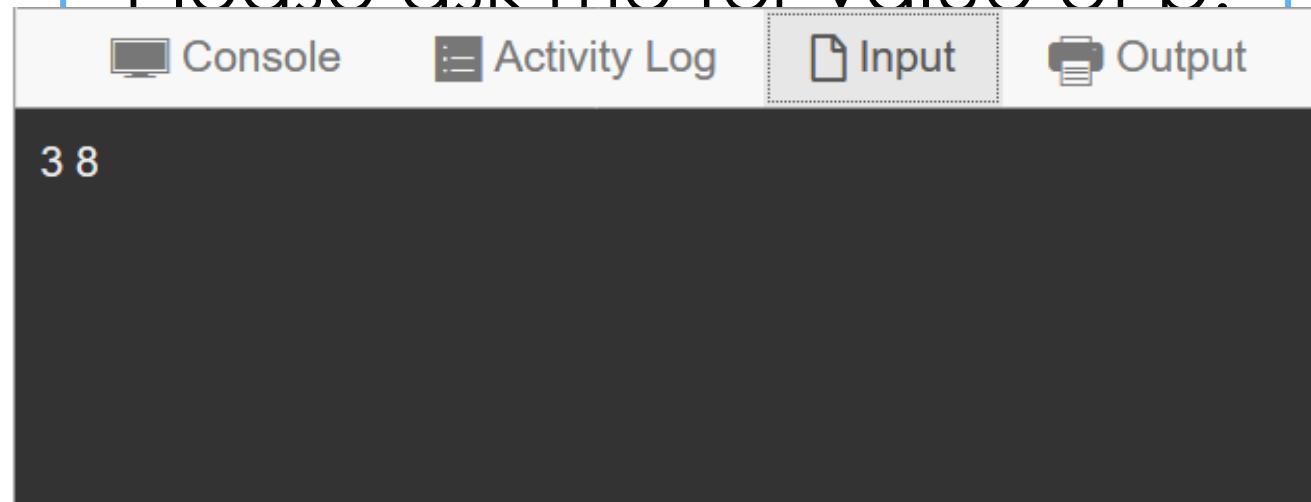
HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    printf("%d", a + b);
    return 0;
}
```



HOW WE USUALLY SPEAK TO A HUMAN

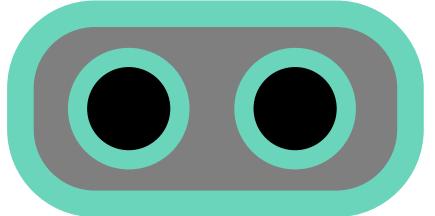
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    printf("%d", a + b);
    return 0;
}
```



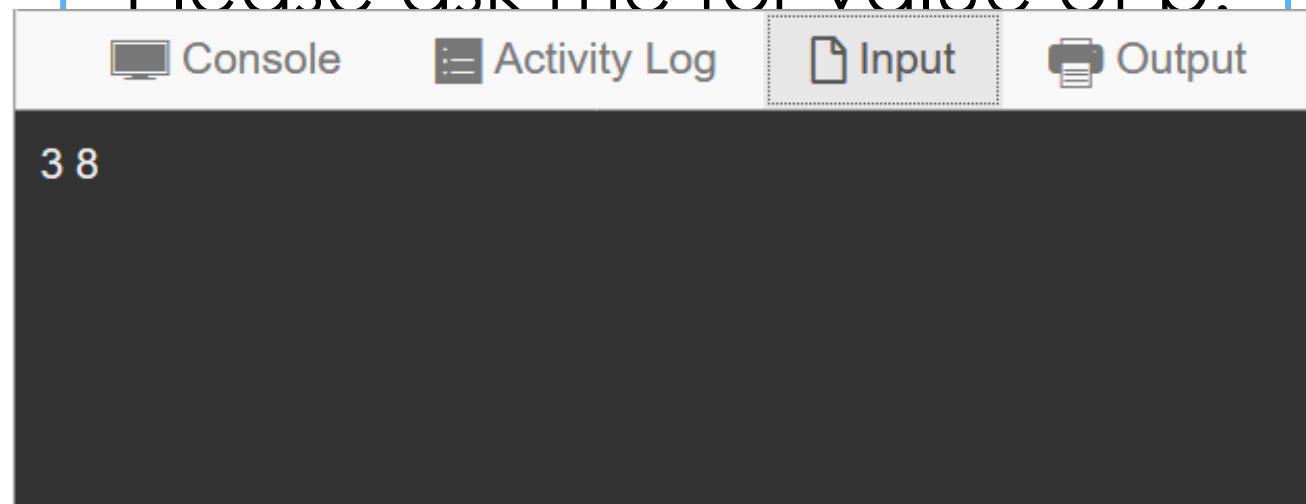
3

a

b

HOW WE USUALLY SPEAK TO A HUMAN

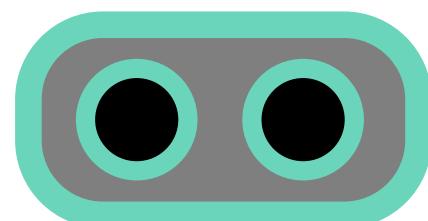
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

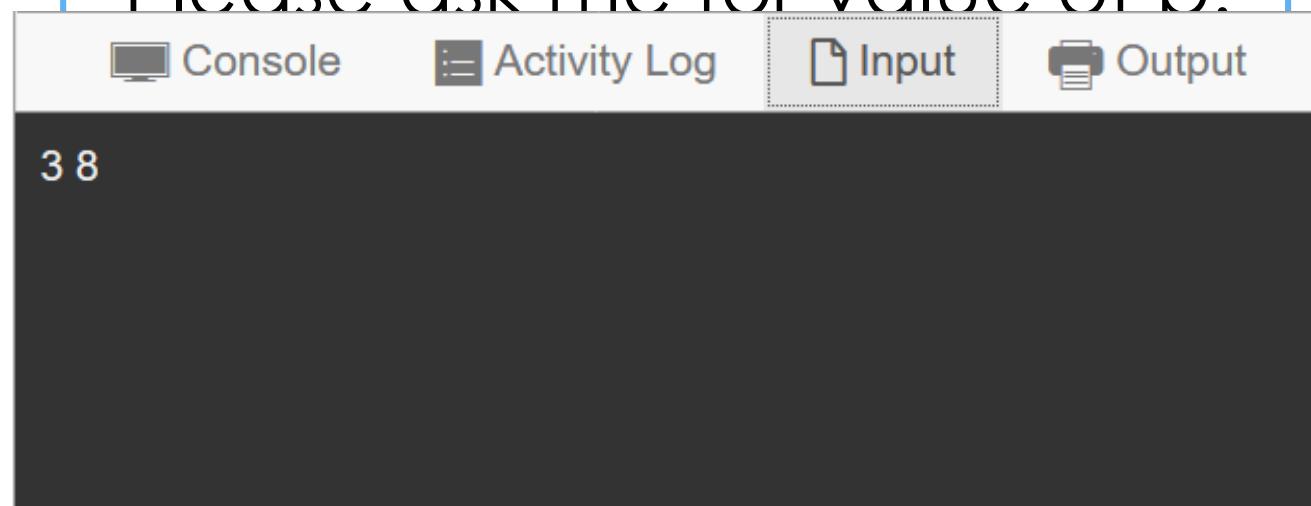
```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    printf("%d", a + b);
    return 0;
}
```



3 8
a b

HOW WE USUALLY SPEAK TO A HUMAN

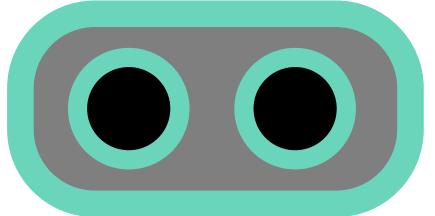
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    printf("%d", a + b);
    return 0;
}
```



3

8

a

b

11

HOW WE USUALLY SPEAK TO A HUMAN

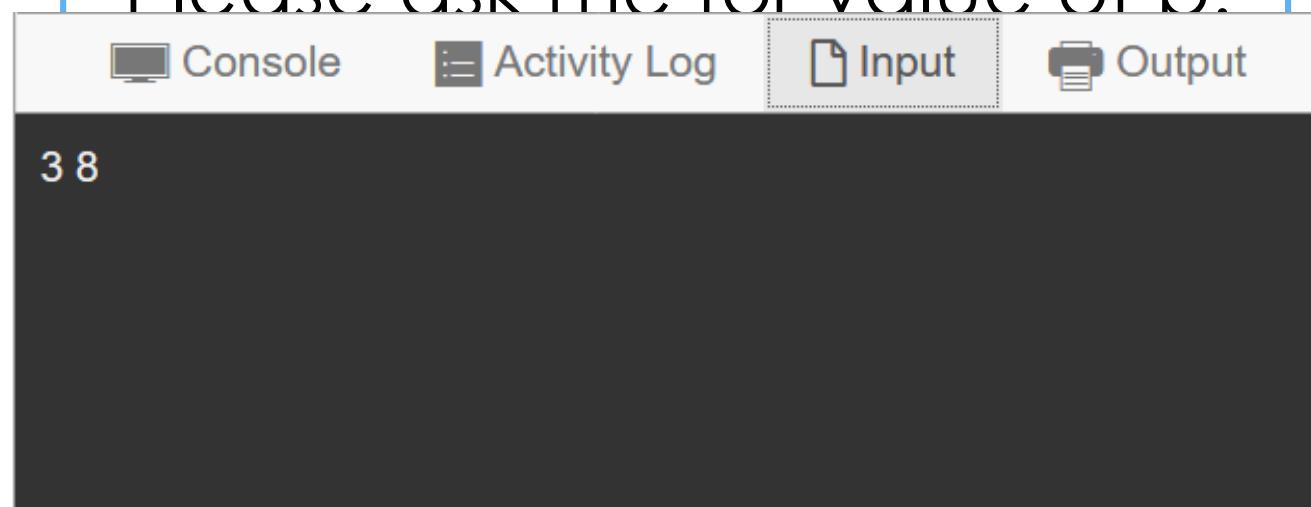
Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

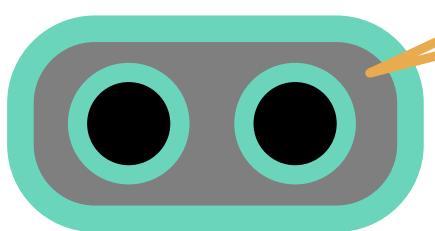
Please ask me for value of b.



Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d", &a);
    scanf("%d", &b);
    printf("%d", a + b);
    return 0;
}
```

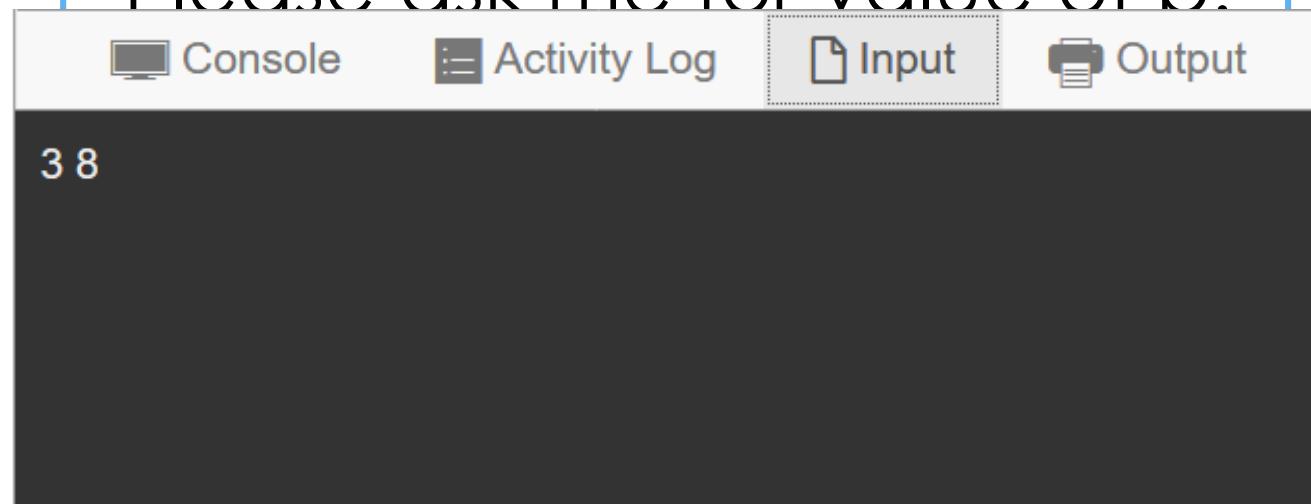


The code defines two integer variables, a and b, and reads their values from the user via scanf(). It then prints the sum of a and b using printf(). The final line returns 0.

3 8
a b
11

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.



Words of Caution

35



Words of Caution

35

In Prutor, input has to be specified before “Execute”



Words of Caution

35

In Prutor, input has to be specified before “Execute”
Please be very careful about this common mistake



Words of Caution

35

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

```
scanf("%d",a);
```



Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

```
scanf("%d",a);
```



Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

```
scanf("%d",a);
```



```
scanf("%d",&a);
```



Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

```
scanf("%d",a);
```



```
scanf("%d",&a);
```



Words of Caution

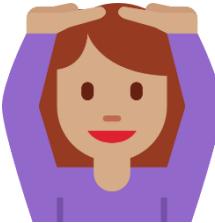
In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

`scanf("%d",a);`



`scanf("%d",&a);`



Will explain what this & means, in a few weeks



Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

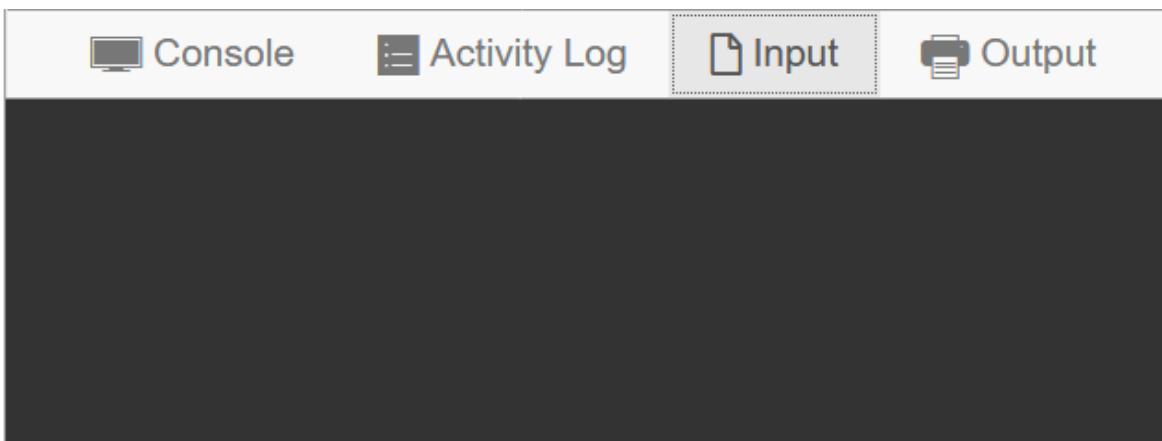
`scanf("%d",a);`



`scanf("%d",&a);`



Will explain what this & means, in a few weeks



Words of Caution

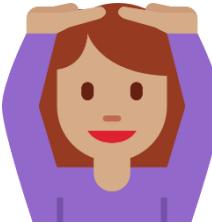
In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

`scanf("%d",a);`



`scanf("%d",&a);`



Will explain what this & means, in a few weeks

A screenshot of the Prutor application interface. At the top, there are four tabs: "Console", "Activity Log", "Input" (which is highlighted with a dotted border), and "Output". Below the tabs, the number "38" is displayed. The main area is a dark gray text input field where code can be entered.



Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

```
scanf("%d",a);
```



```
scanf("%d",&a);
```



Will explain what this & means, in a few weeks

A screenshot of the Prutor application interface. At the top, there is a navigation bar with four items: "Console", "Activity Log", "Input" (which is highlighted with a dotted border), and "Output". Below the navigation bar is a large dark gray area where the number "38" is displayed in the top-left corner. The rest of the area is blank and dark.

A screenshot of the Prutor application interface, similar to the one above but with a key difference. The "Input" tab at the top is no longer highlighted with a dotted border. The rest of the interface, including the dark gray main area and the number "38" in the top-left corner, remains the same.

Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

```
scanf("%d",a);
```



```
scanf("%d",&a);
```



Will explain what this & means, in a few weeks

A screenshot of the Prutor interface. At the top, there are four tabs: "Console", "Activity Log", "Input" (which is highlighted with a dotted border), and "Output". In the "Input" field, the code "scanf("%d",a);" is entered. Below the tabs, the "Console" area shows the output "3 8", which is incorrect because it shows two separate inputs instead of one combined value.

```
scanf("%d",a);
```

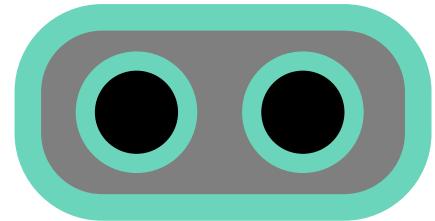
```
3 8
```

A screenshot of the Prutor interface. The tabs are the same: "Console", "Activity Log", "Input" (highlighted with a dotted border), and "Output". In the "Input" field, the code "scanf("%d",&a);" is entered. Below the tabs, the "Console" area shows the output "3\n8", which is correct because it shows a single input value being assigned to the variable "a".

```
scanf("%d",&a);
```

```
3  
8
```

Words of Caution



35

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

`scanf("%d",a);`



`scanf("%d",&a);`



Will explain what this & means, in a few weeks

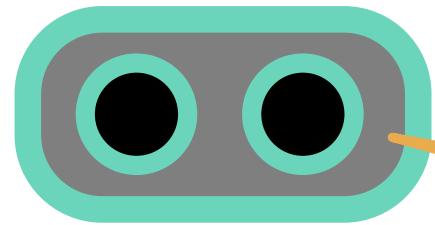
Console Activity Log Input Output

3
8

Console Activity Log Input Output

3
8

Words of Caution



35

In Prutor, input has to be specified before “E Both work!
Please be very careful about this common Experiment!

`scanf("%d",a);`



`scanf("%d",&a);`



Will explain what this & means, in a few weeks

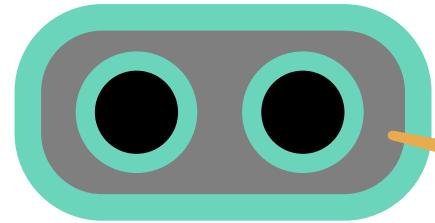
Console Activity Log Input Output

3
8

Console Activity Log Input Output

3
8

Words of Caution



35

In Prutor, input has to be specified before “E Both work!
Please be very careful about this common Experiment!

`scanf("%d",a);`



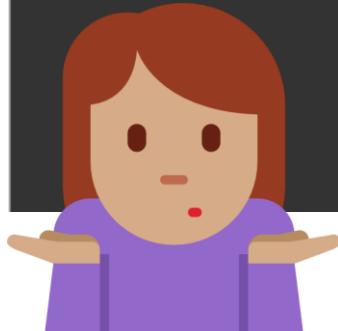
`scanf("%d",&a);`



Will explain what this & means, in a few weeks

Console Activity Log Input Output

3 8

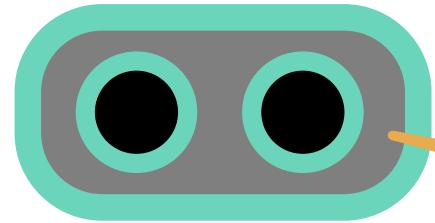


Console Activity Log Input Output

3
8



Words of Caution



35

In Prutor, input has to be specified before “E Both work!
Please be very careful about this common Experiment!

`scanf("%d",a);`



`scanf("%d",&a);`



Will explain what this & means, in a few weeks

Console

Activity Log

Input

Output

3 8

Why? Space is not same as newline 😐

Console

Activity Log

Input

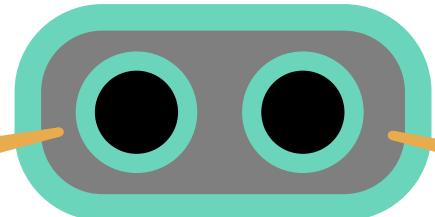
Output

3

8

Words of Caution

35



Yes, in printf they are different but in
scanf, both look like *whitespaces* to me

Please be very careful about this common mistake!

ecified before “E”

Both work!
Experiment!

scanf("%d",a);



scanf("%d",&a);



Will explain what this & means, in a few weeks

Console

Activity Log

Input

Output

3 8

Why? Space is not same as newline 😞

Console

Activity Log

Input

Output

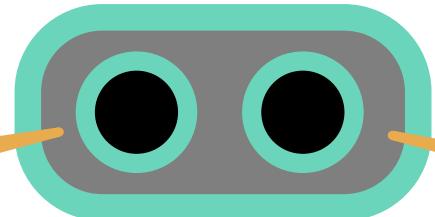
3

8



Words of Caution

35



Yes, in printf they are different but in
scanf, both look like *whitespaces* to me

Please be very careful about this common mistake!

ecified before “E”

Both work!
Experiment!

scanf("%d",a);



scanf("%d",&a);



Will explain what this & means, in a few weeks

Why? Space is not
same as newline 😞

Huh! What is a
whitespace?

Console

Activity Log

Input

Output

3 8

Console

Activity Log

Input

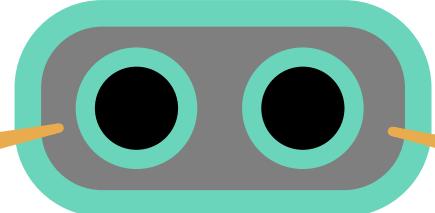
Output

3

8

Words of Caution

35



Yes, in printf they are different but in
scanf, both look like *whitespaces* to me

Please be very careful

ecified before “E”
ut this common mistake

Both work!
Experiment!

Space, Tab, Newline are called whitespace
characters since they are invisible ☺

scanf(“%d”,&a);



Will explain what this & means, in a few weeks

Console Activity Log Input Output

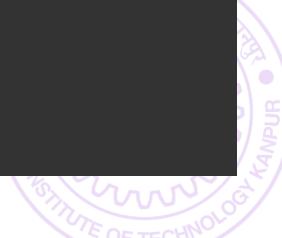
3 8

Why? Space is not
same as newline ☺

Huh! What is a
whitespace?

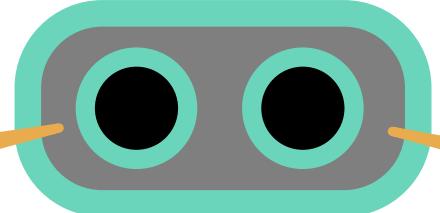
Console Activity Log Input Output

3
8



Words of Caution

35



Yes, in printf they are different but in
scanf, both look like *whitespaces* to me

Please be very careful

ecified before “E”

ut this common mistake

Both work!

Experiment!

Space, Tab, Newline are called whitespace
characters since they are invisible ☺

```
3 | ... int a, b, c, d;
```

```
scanf("%d",&a);
```



Will explain what this & means, in a few weeks

Console Activity Log Input Output

3 8

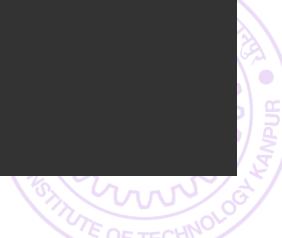
Why? Space is not same as newline ☺

Huh! What is a whitespace?

Console Activity Log Input Output

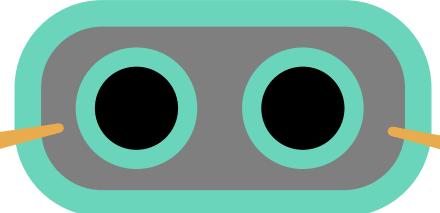
3

8



Words of Caution

35



Yes, in printf they are different but in
scanf, both look like *whitespaces* to me

Please be very careful

ecified before “E”
ut this common mistake

Both work!
Experiment!

Space, Tab, Newline are called whitespace
characters since they are invisible ☺

```
3 int a, b, c, d;
```

Explains what this & means, in a few weeks

scanf("%d",&a);

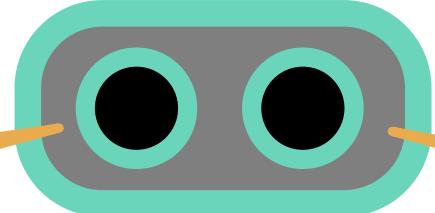


A screenshot of a programming interface. At the top, there are tabs for "Console", "Activity Log", "Input", and "Output". The "Console" tab is active, showing the numbers "3" and "8" on separate lines. A speech bubble from a character on the left says: "Why? Space is not same as newline ☺". Another speech bubble from the same character below says: "Huh! What is a whitespace?".

A screenshot of a programming interface. At the top, there are tabs for "Console", "Activity Log", "Input", and "Output". The "Console" tab is active, showing the numbers "3" and "8" on separate lines. This part of the slide also includes the circular logo of IIT Kanpur in the bottom right corner.

Words of Caution

35



Yes, in printf they are different but in
scanf, both look like *whitespaces* to me

Please be very careful

ecified before “E”
ut this common mistake

Both work!
Experiment!

Space, Tab, Newline are called whitespace
characters since they are invisible ☺

```
3     ...int0a,0b,0c,0d;
```

TAB SPACE

scanf("%d",&a);



means, in a few weeks

Console Activity Log Input Output

3
8

Why? Space is not same as newline ☺

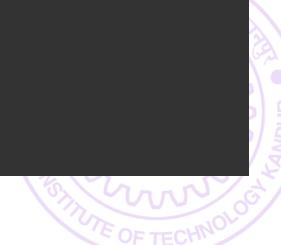
Huh! What is a whitespace?

A cartoon girl with brown hair and a purple shirt is shown at the bottom left, gesturing with her hands. A speech bubble from her contains the text "Why? Space is not same as newline ☺". Another speech bubble below her contains the text "Huh! What is a whitespace?". The terminal window above shows the numbers 3 and 8 on separate lines.

Console Activity Log Input Output

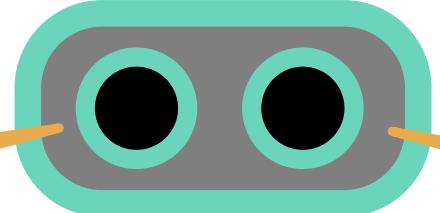
3 8

The terminal window shows the numbers 3 and 8 on the same line, separated by a single space character.



Words of Caution

35



Yes, in printf they are different but in
scanf, both look like *whitespaces* to me

Please be very careful

ecified before “E”
ut this common mistake

Both work!
Experiment!

Space, Tab, Newline are called whitespace
characters since they are invisible ☺

3 ...int0a,0b,0c,0d;0
TAB SPACE NEWLINE

scanf("%d",&a);



... means, in a few weeks

Console Activity Log Input Output

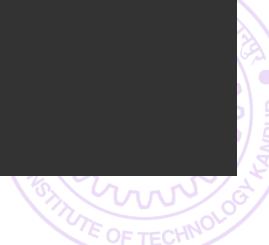
3 8

Why? Space is not
same as newline ☺

Huh! What is a
whitespace?

Console Activity Log Input Output

3
8



Shorthand for multiple inputs

58



ESC101: Fundamentals
of Computing

Shorthand for multiple inputs

58

HOW WE USUALLY SPEAK TO A HUMAN



Shorthand for multiple inputs

58

HOW WE USUALLY SPEAK TO A HUMAN



Shorthand for multiple inputs

58

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?



Shorthand for multiple inputs

58

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello



Shorthand for multiple inputs

58

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.



Shorthand for multiple inputs

58

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



Shorthand for multiple inputs

58

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.



Shorthand for multiple inputs

58

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I
will give both values separated
by a whitespace
Please print their sum.
Goodbye



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I
will give both values separated
by a whitespace
Please print their sum.
Goodbye



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;  
  
    scanf("%d%d", &a, &b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace
Please print their sum.
Goodbye



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;  
  
    scanf("%d%d", &a, &b);  
  
    printf("%d", a + b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace
Please print their sum.
Goodbye



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;  
  
    scanf("%d%d", &a, &b);  
  
    printf("%d", a + b);  
  
    return 0;
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace
Please print their sum.
Goodbye



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;  
  
    scanf("%d%d", &a, &b);  
  
    printf("%d", a + b);  
  
    return 0;  
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace
Please print their sum.
Goodbye

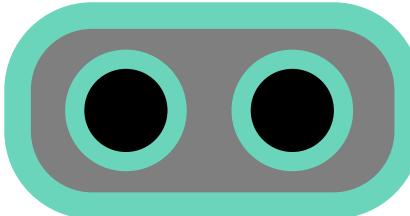


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;  
  
    scanf("%d%d", &a, &b);  
  
    printf("%d", a + b);  
  
    return 0;  
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace
Please print their sum.
Goodbye

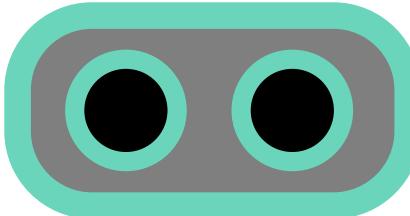


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;  
  
    scanf("%d%d", &a, &b);  
  
    printf("%d", a + b);  
  
    return 0;  
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace
Please print their sum.
Goodbye

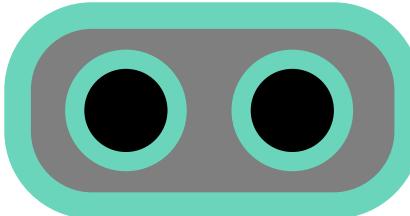


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
    int a, b;  
    scanf("%d%d", &a, &b);  
    printf("%d", a + b);  
    return 0;  
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace
Please print their sum.
Goodbye

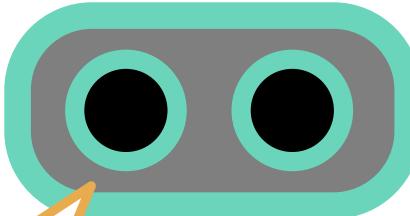


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
    int a, b;  
    scanf("%d%d", &a, &b);  
    printf("%d", a + b);  
    return 0;  
}
```



Input please

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace
Please print their sum.
Goodbye

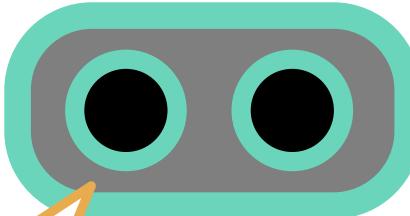


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```



Input please

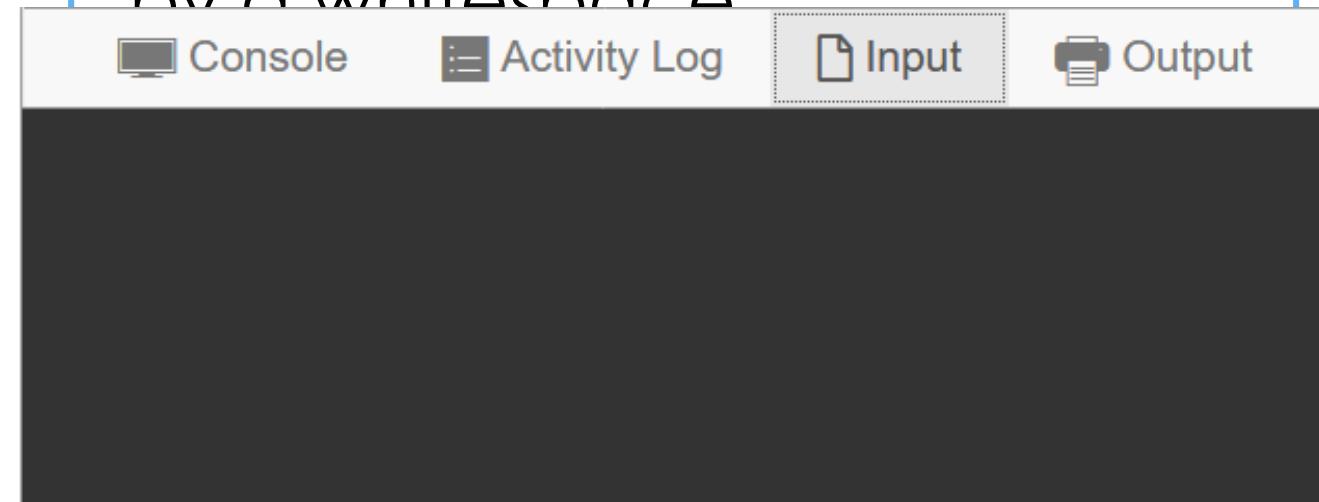
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

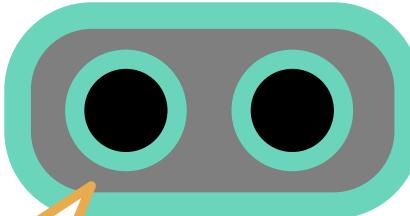


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```



Input please

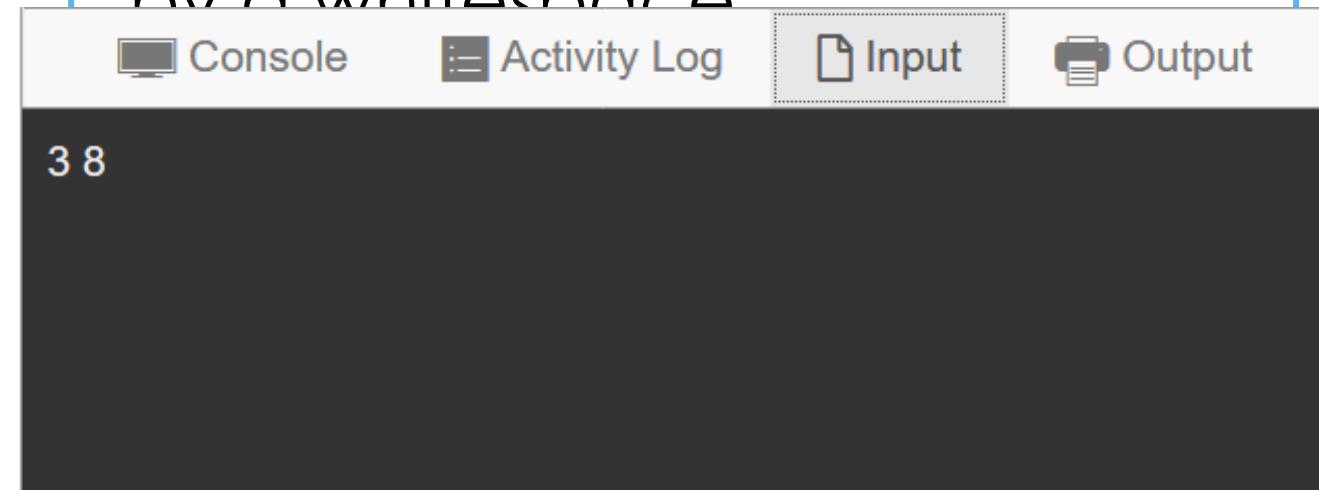
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

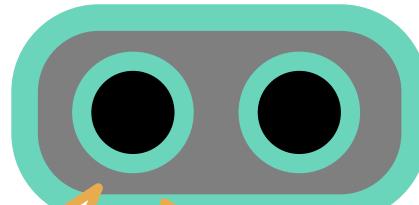


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```



Input please

Thanks

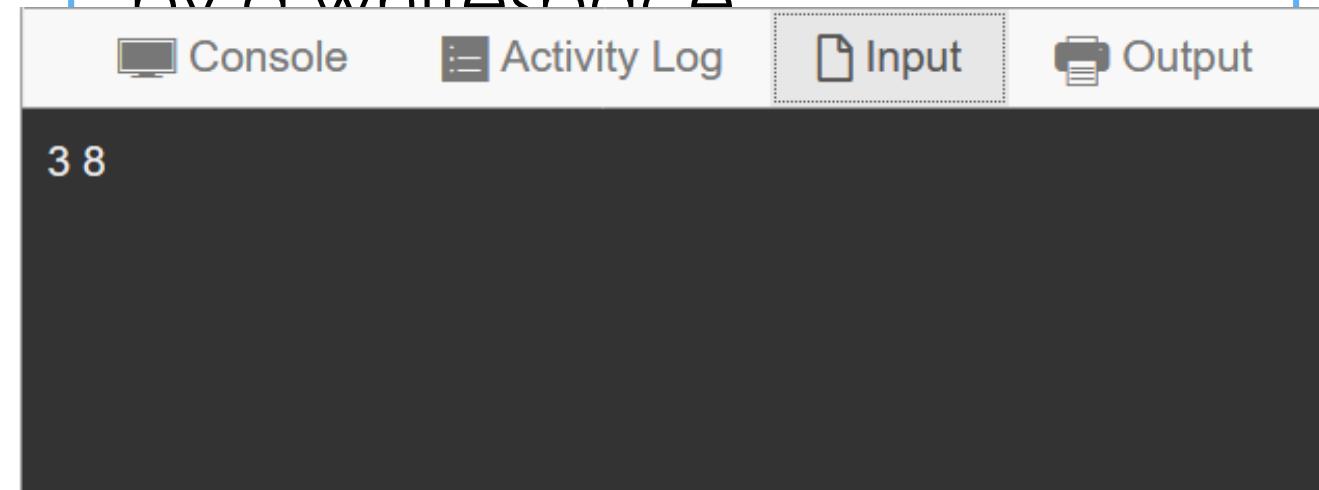
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

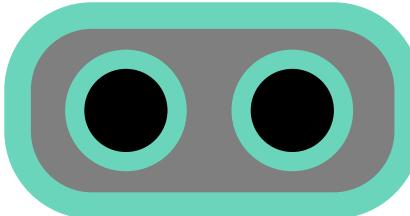


Shorthand for multiple inputs

58

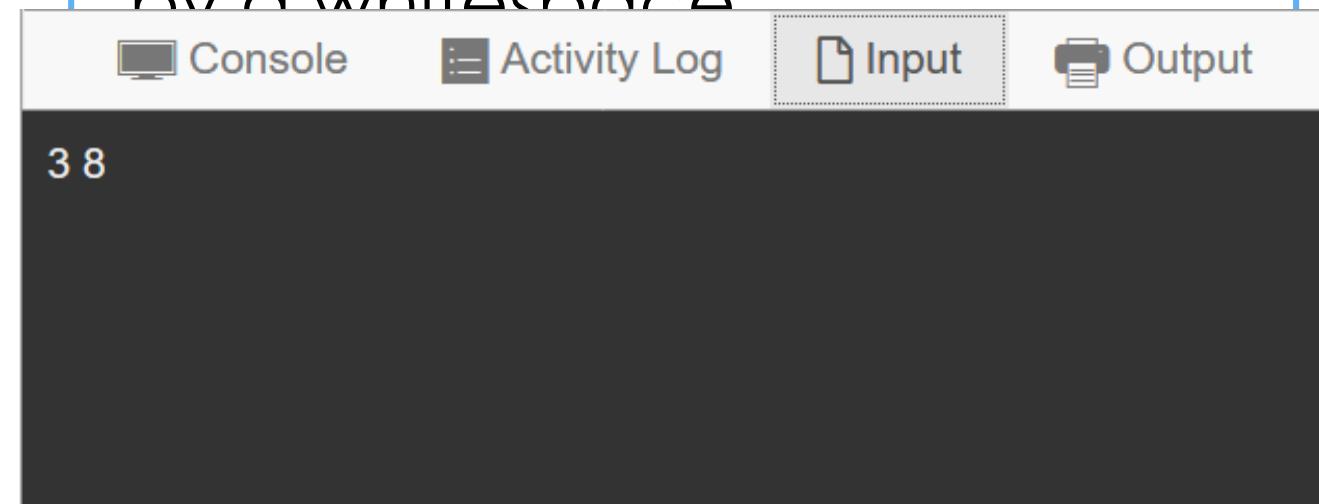
HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>  
  
int main(){  
  
    int a, b;  
  
    scanf("%d%d", &a, &b);  
  
    printf("%d", a + b);  
  
    return 0;  
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I
will give both values separated
by a whitespace.

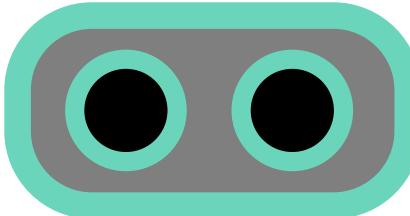


Shorthand for multiple inputs

58

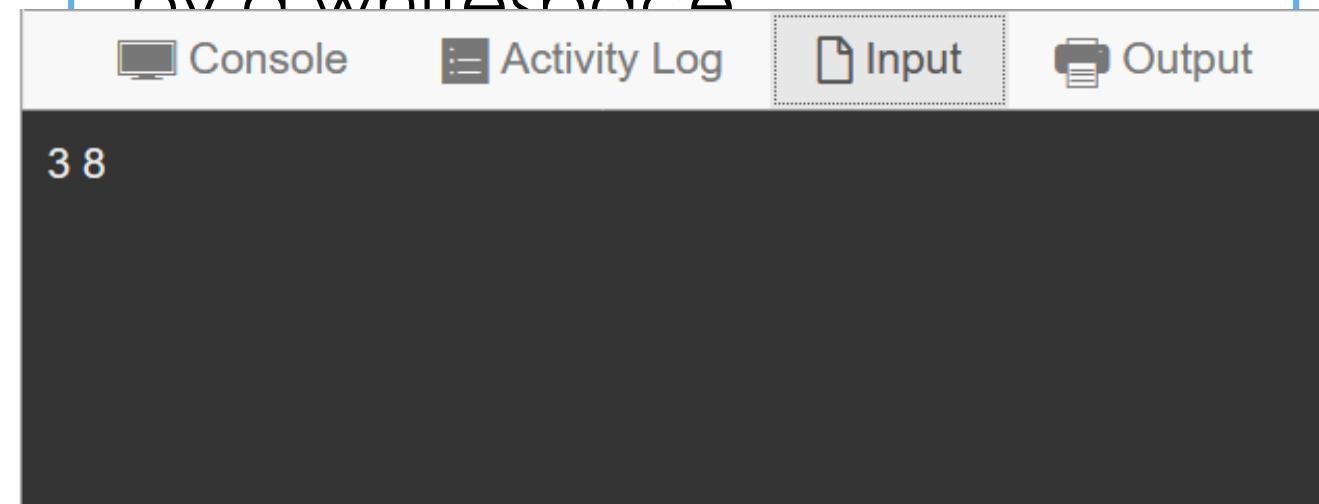
HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace.

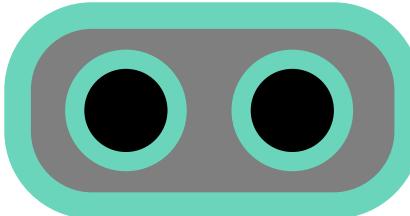


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```



3

8

a

b

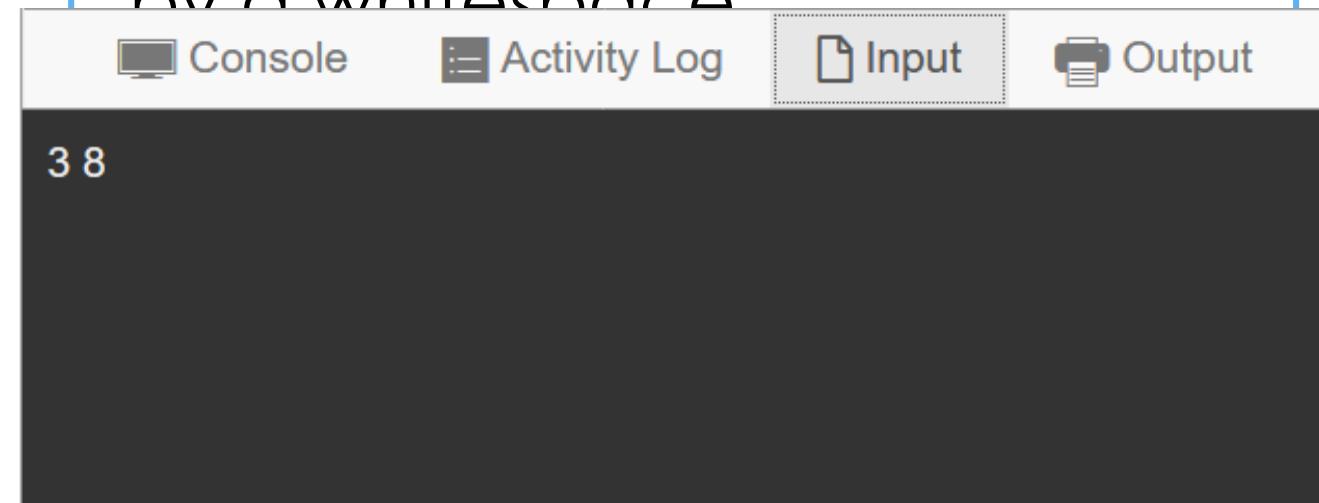
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

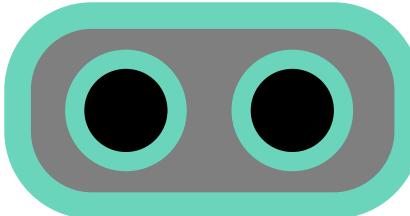


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```



3

8

a

b

11

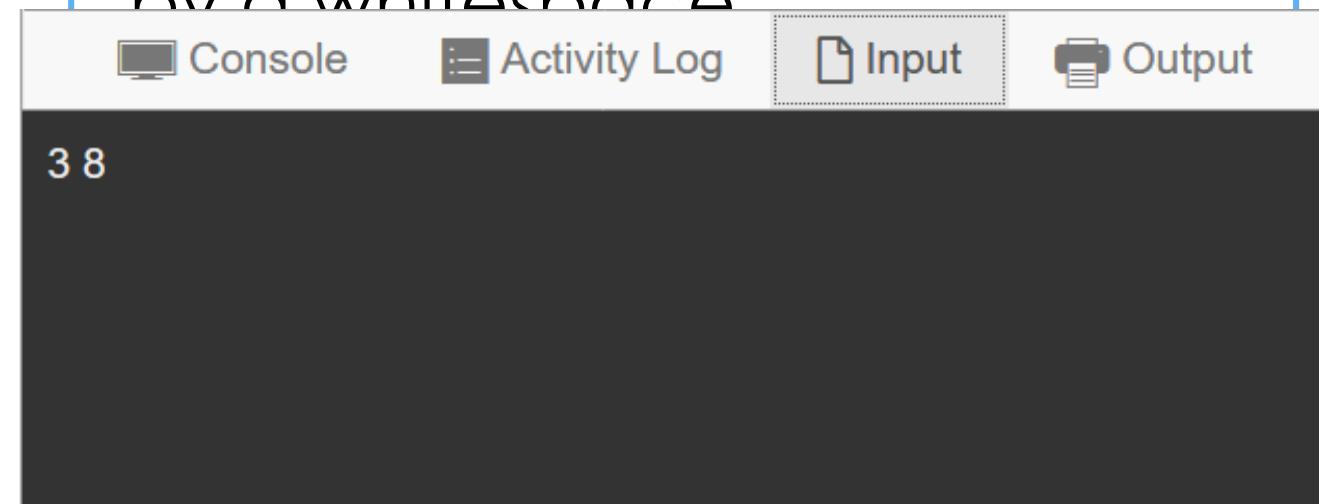
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

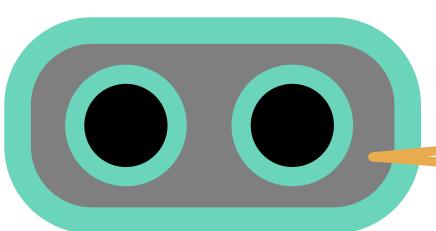


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

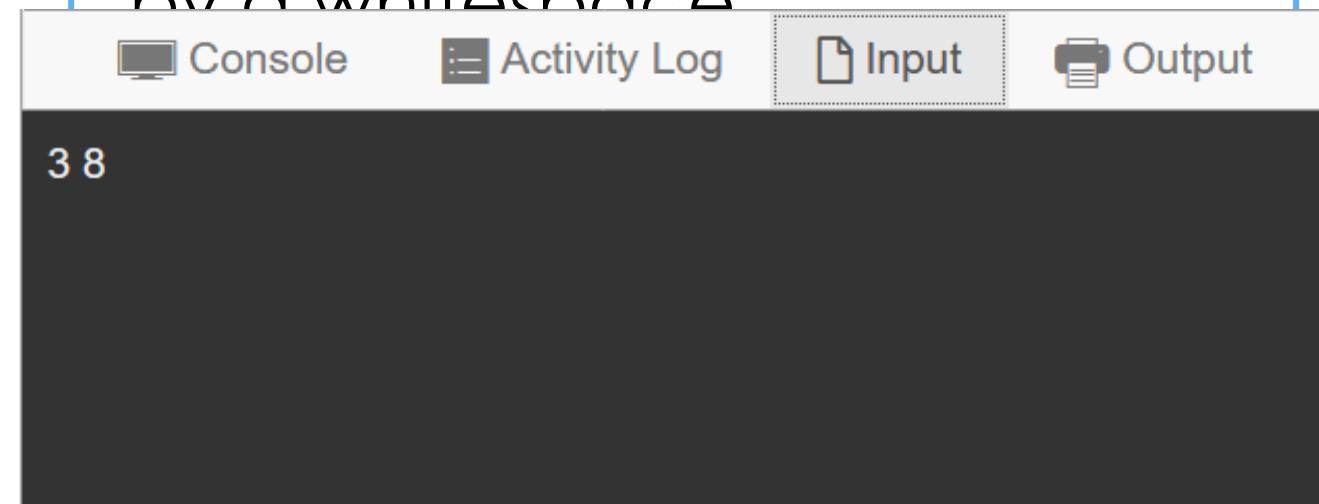
```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```



The diagram shows a teal-colored face with two black circular eyes. An orange arrow points from the number 11 in the code above to the right eye of the face.

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace.



Shorthand for multiple inputs

58

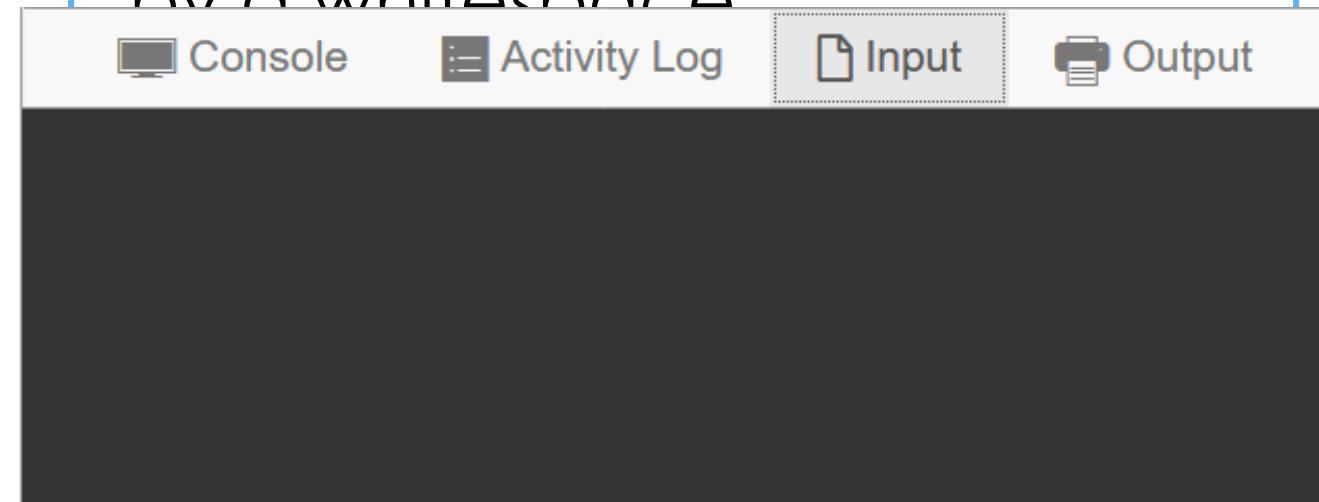
HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```

The diagram illustrates the flow of data from input to output. On the left, two blue boxes contain the numbers 3 and 8. Arrows point from these boxes to the variables 'a' and 'b' in the code. On the right, an orange box contains the number 11, which is the result of the expression 'a + b' in the printf statement.

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace.



Shorthand for multiple inputs

58

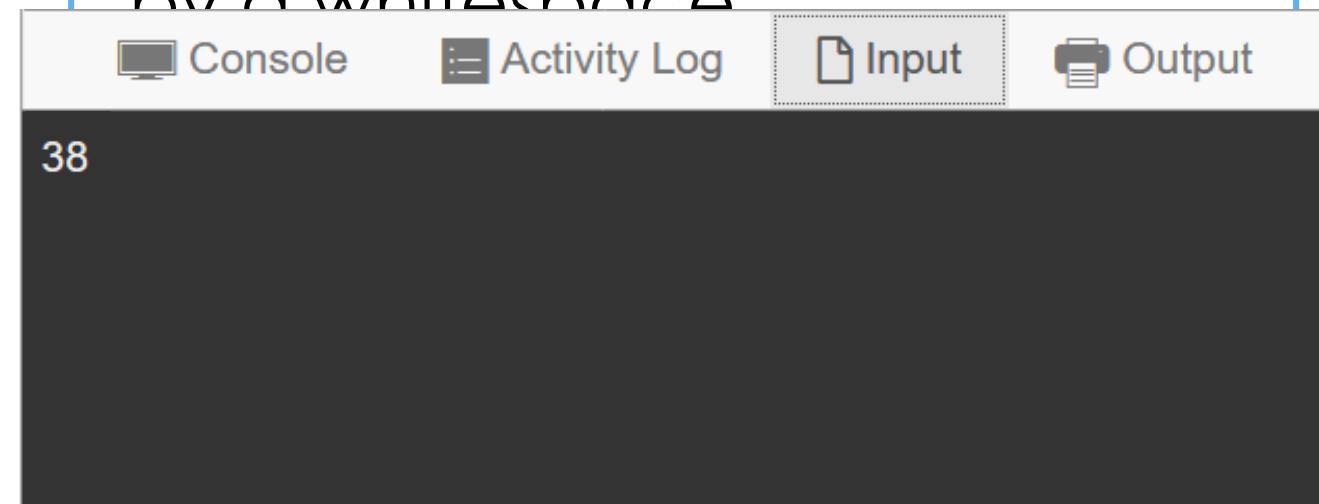
HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```

The diagram illustrates the flow of data from input to output. On the left, two blue boxes contain the numbers 3 and 8. Arrows point from these boxes to the variables 'a' and 'b' in the code. On the right, an orange box contains the number 11, which is the result of the expression 'a + b' in the printf statement.

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace.



Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    You entered
    only one integer
    printf("a = %d, b = %d\n", &a, &b);
    return 0;
}
```

3 8
a b
11

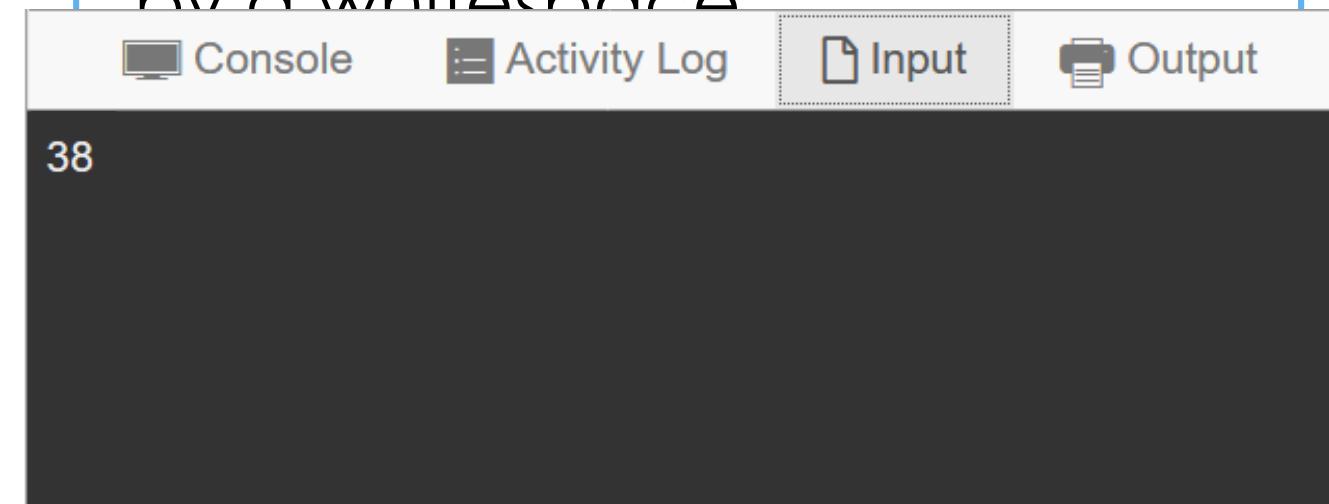
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace.

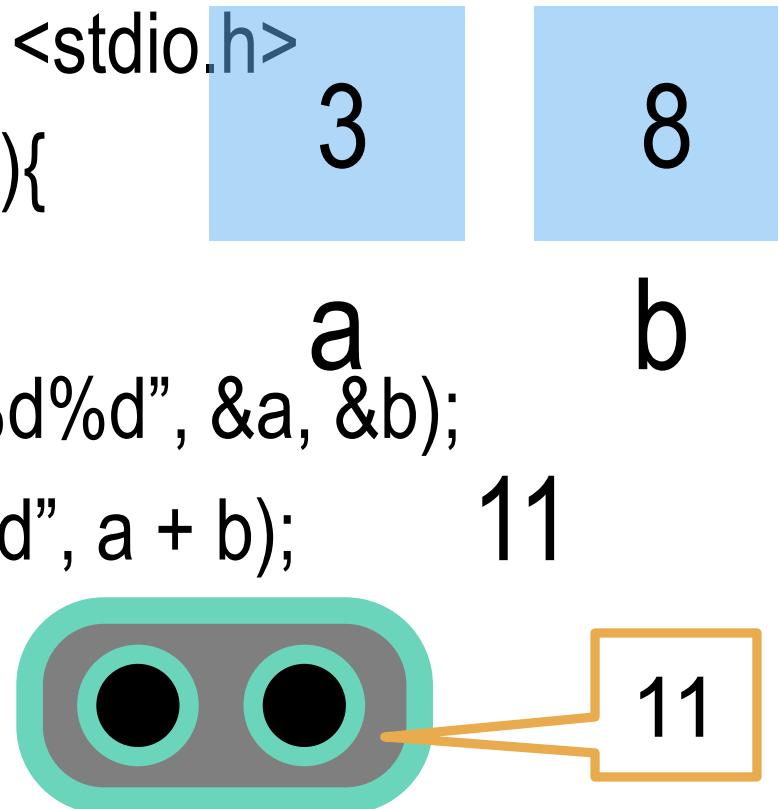


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

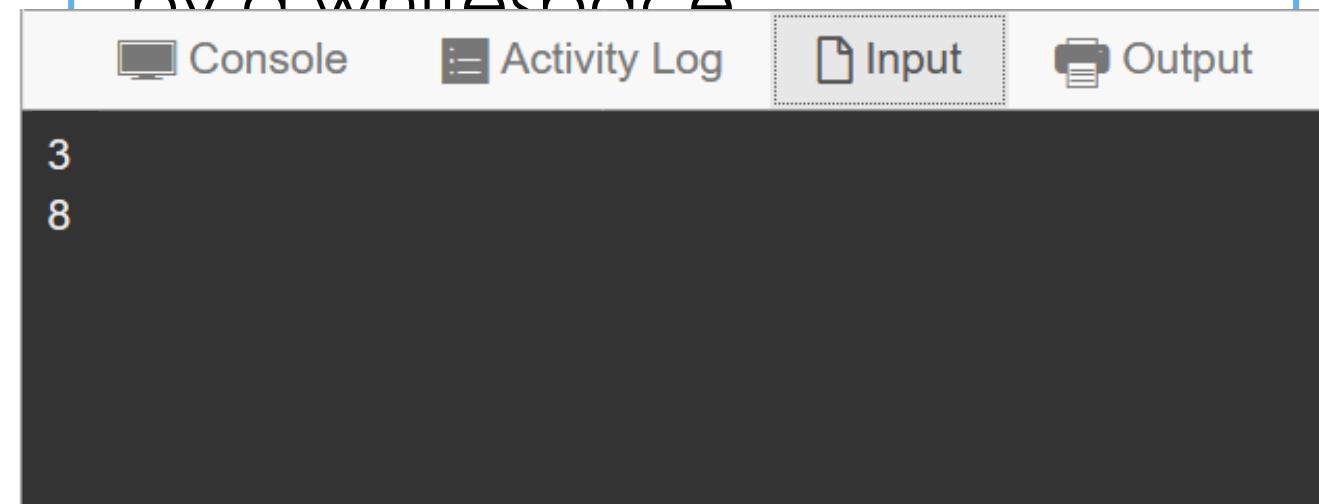
```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```



The diagram illustrates the input required for the program. Two blue boxes contain the integers 3 and 8. An orange arrow points from the sum 11, which is enclosed in a yellow box, to a teal-colored icon representing a double circular input port.

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace.

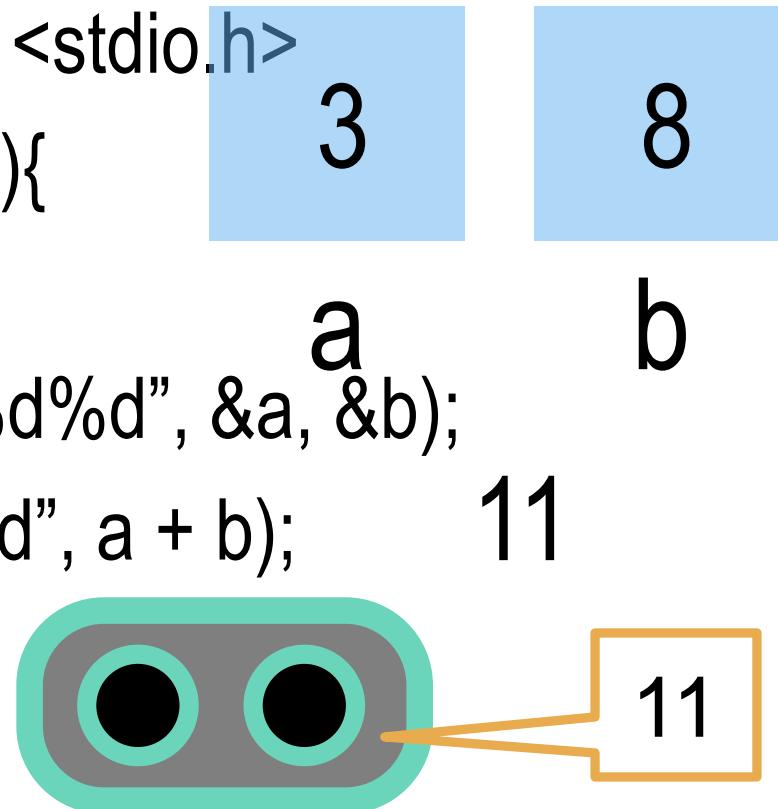


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

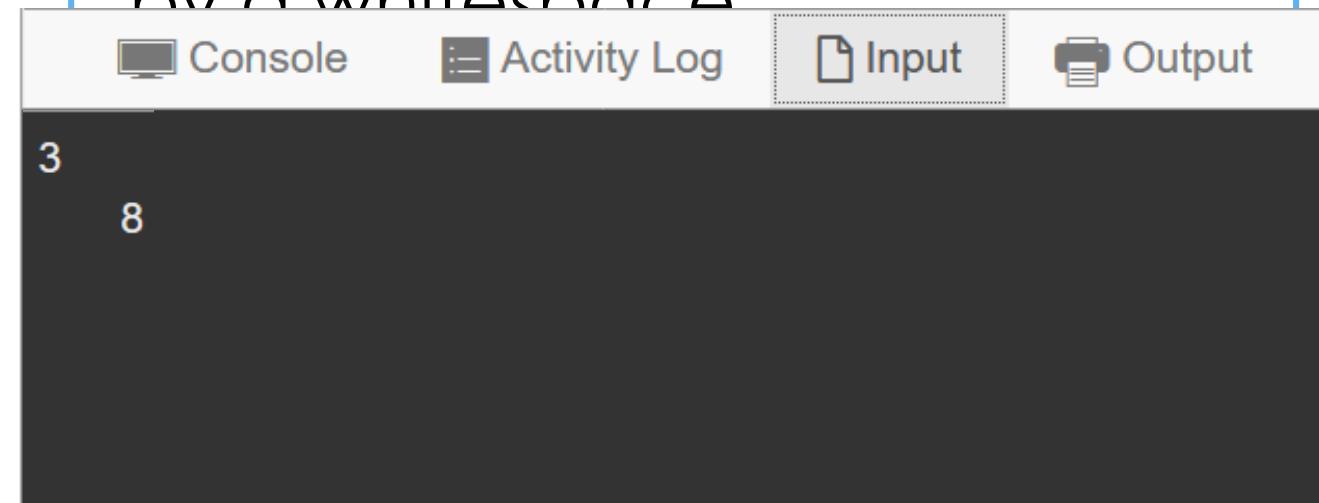
```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d", a + b);
    return 0;
}
```



The code demonstrates how to read two integers from standard input and print their sum. The input values 3 and 8 are shown in blue boxes, and the resulting output value 11 is shown in an orange box.

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace.

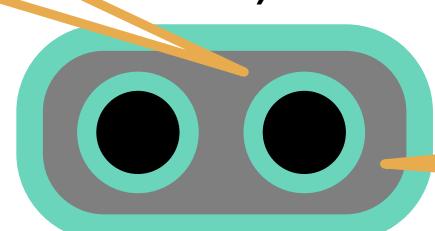


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

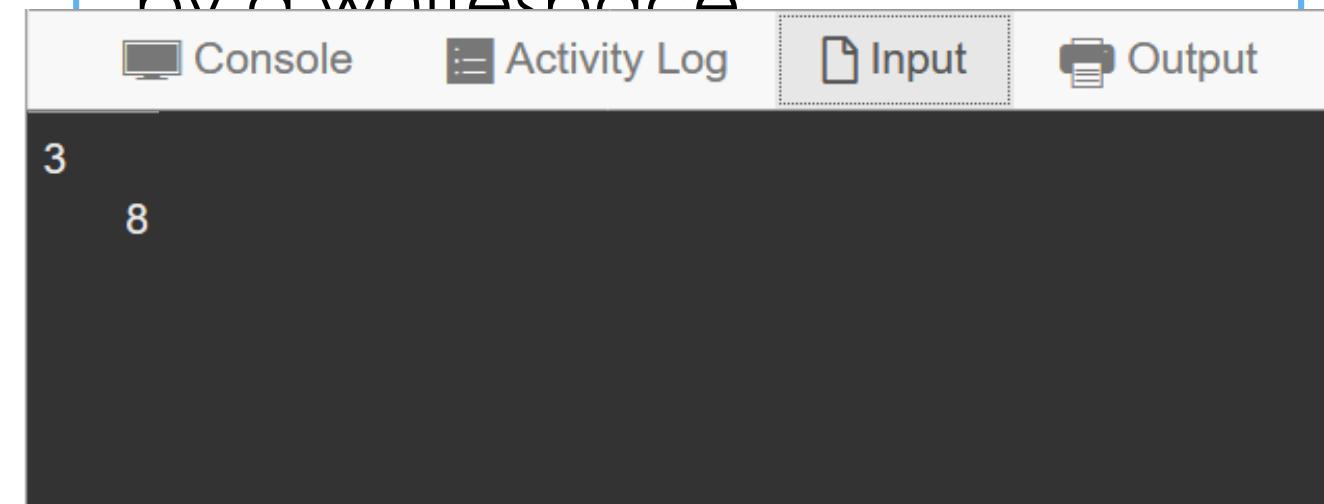
```
#include <stdio.h>
int main(){
    int a, b;
    All look the same to me
    printf("%d\n%d", &a, &b);
    a = 3;
    b = 8;
    printf("a + b = %d", a + b);
    return 0;
}
```



The code demonstrates how to read two integers from standard input. The printf statements at the bottom are highlighted with orange boxes. Arrows point from the numbers 3 and 8 in the code to a teal-colored eye icon, which has two black pupils. An arrow also points from the number 11 in the code to a yellow-outlined box containing the number 11, representing the output of the program.

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace.



Shorthand for multiple inputs



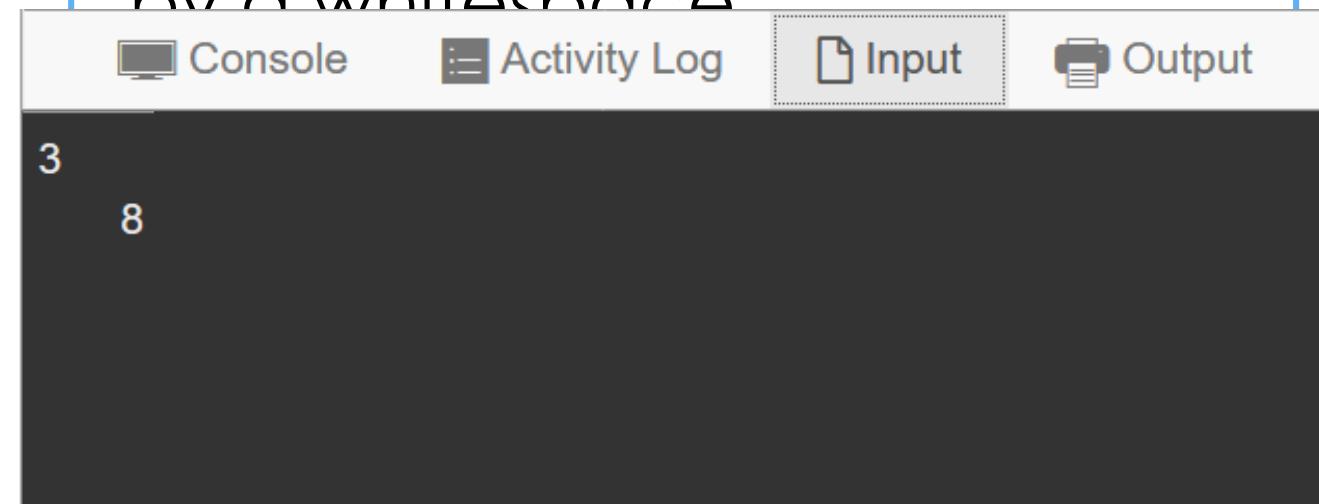
HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    All look the same to me
    printf("%d\n%d", &a, &b);
    a = 3;
    b = 8;
    printf("a + b = %d", a + b);
    return 0;
}
```

The code above is shown in a pink-bordered box. A diagram below it illustrates how the compiler sees the input. It features a teal-colored face with two black circular eyes. Four orange arrows point from the numbers 3, 8, a, and b to the respective eyes. The number 11 is also highlighted with an orange box and points to one of the eyes.

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Ask me for values of a and b. I will give both values separated by a whitespace.



Shorthand for multiple inputs



HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    All look the same to me
    printf("%d\n%d", &a, &b);
    a = 3;
    b = 8;
    printf("a + b = %d", a + b);
    return 0;
}
```

The code block shows a C program that reads two integers from standard input and prints their sum. The input values 3 and 8 are shown in blue boxes. Arrows point from these boxes to the variable declarations in the code. Another arrow points from the value 11 in a blue box to the printf statement, indicating it is the result of the addition.

HOW WE USUALLY TALK

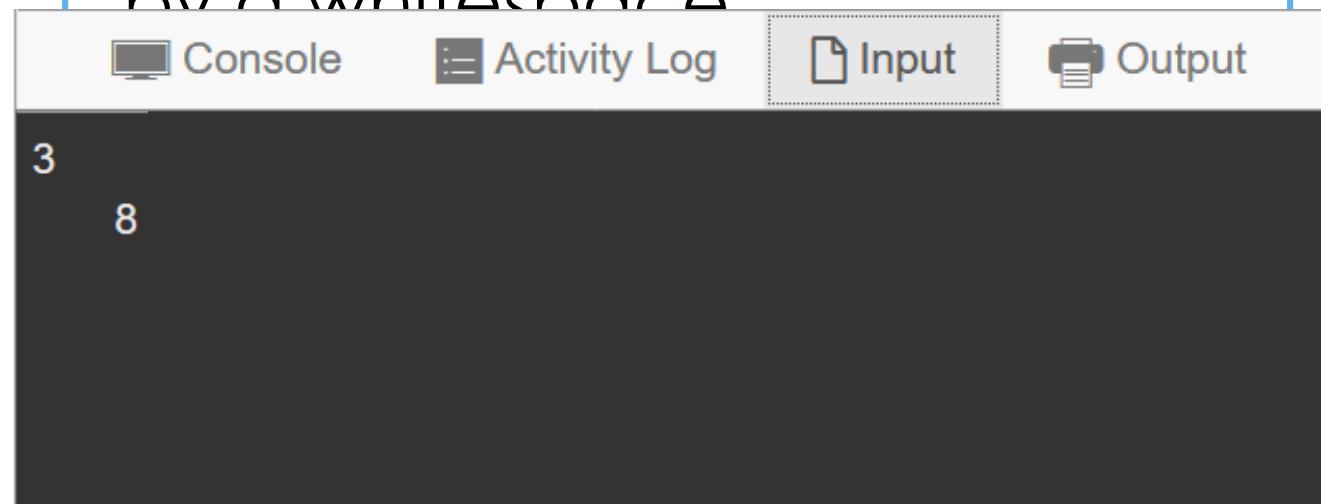
Help!!!

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

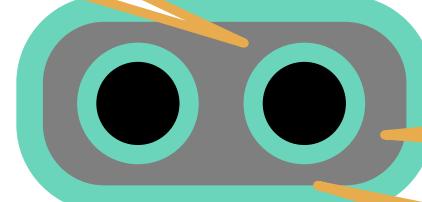


Shorthand for multiple inputs

58

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int a, b;
    All look the same to me
    printf("%d\n%d", &a, &b);
    a = 3;
    b = 8;
    printf("a + b = %d", a + b);
    return 0;
}
```



Okay okay

HOW WE USUALLY TALK

Help!!!

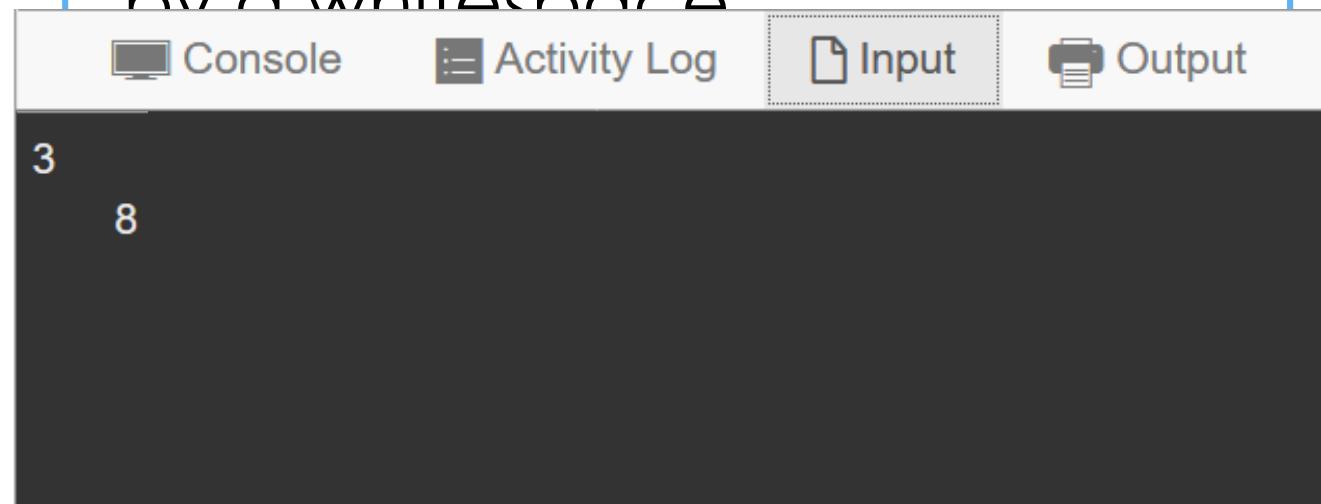


Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



What is going on with scanf?

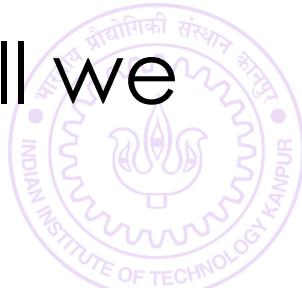
97



What is going on with scanf?

97

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

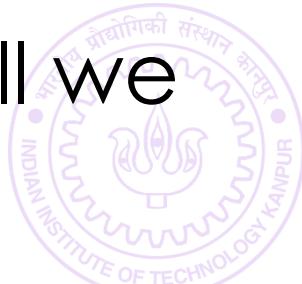


What is going on with scanf?

97

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



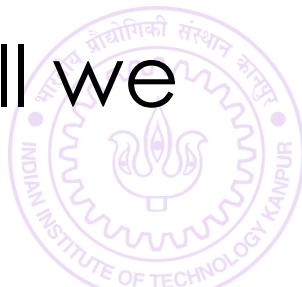
What is going on with scanf?

97

HOW WE USUALLY SPEAK TO A HUMAN

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

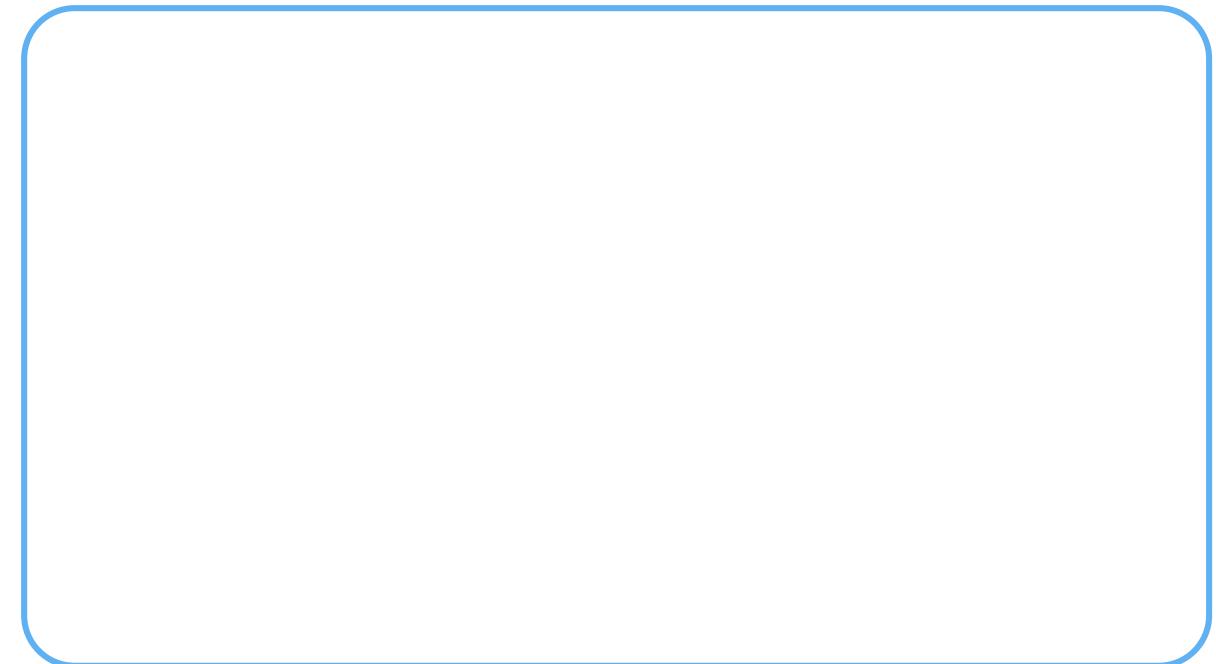
Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

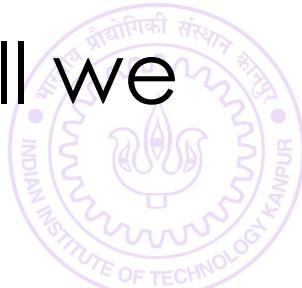
97

HOW WE USUALLY SPEAK TO A HUMAN



Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

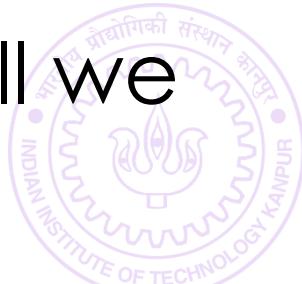
97

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

97

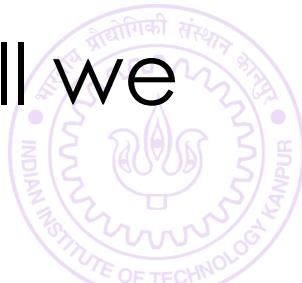
HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

97

HOW WE MUST SPEAK TO MR. COMPILER

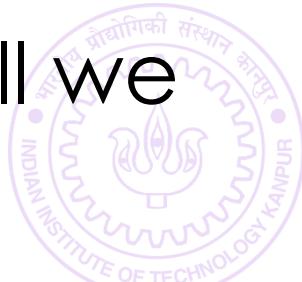
HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

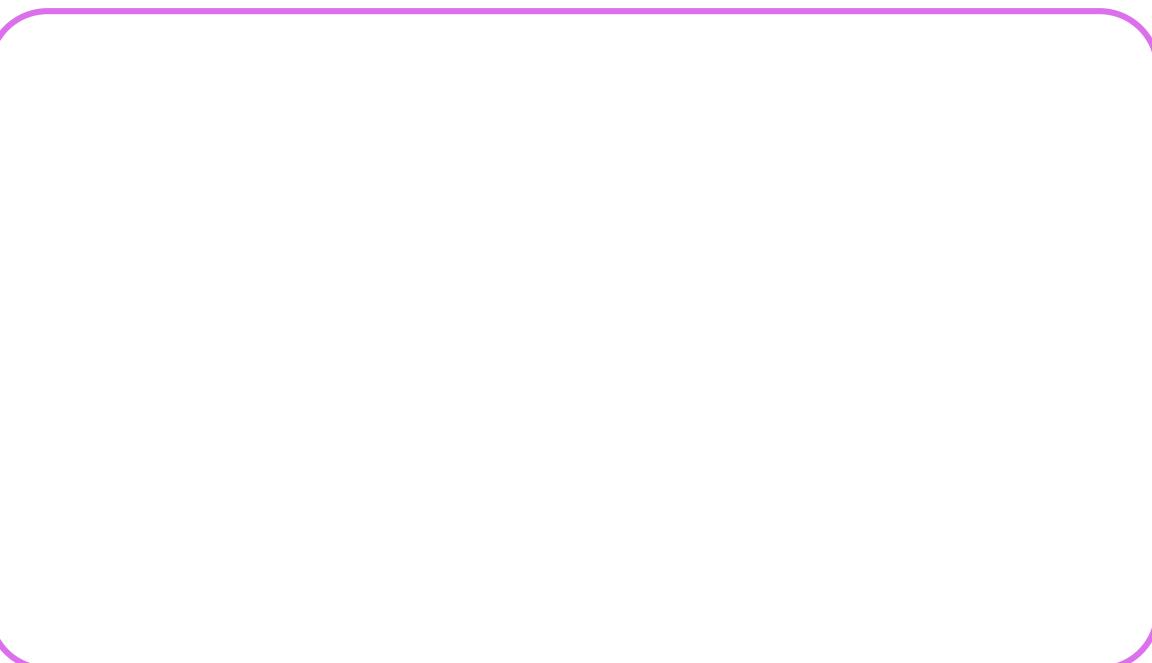
Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

97

HOW WE MUST SPEAK TO MR. COMPILER



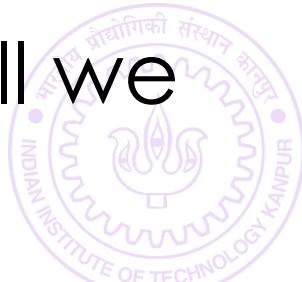
HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

97

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

97

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

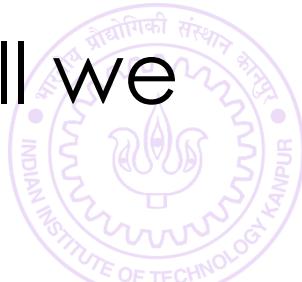
HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

97

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

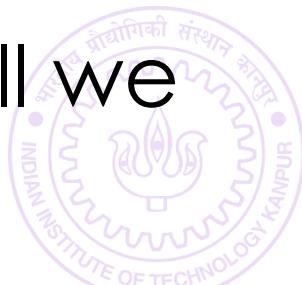
HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

97

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

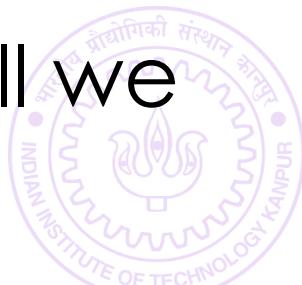
HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

97

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

Format string

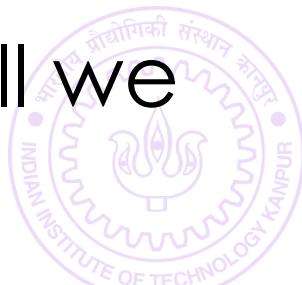
HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



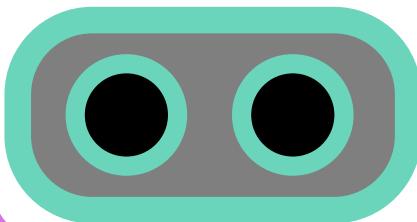
What is going on with scanf?

97

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

Format string



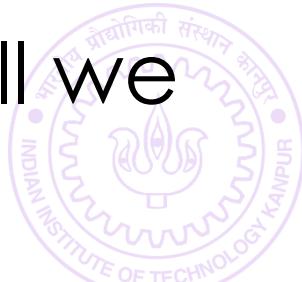
HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces, tabs, newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



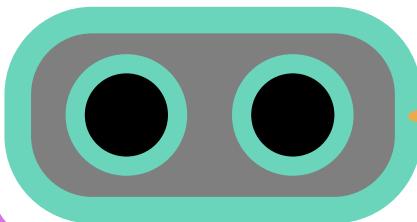
What is going on with scanf?

97

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

Format string



Format string tells me **how** you will write things, and then I am told **where** to store what I have read

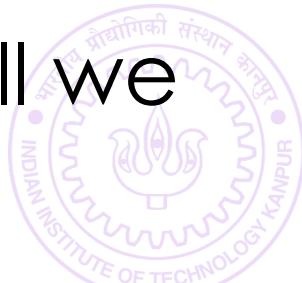
HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

Scanf follows this exact same rule while telling Mr. C how to read



What is going on with scanf?

113



ESC101: Fundamentals
of Computing

What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```



What is going on with scanf?

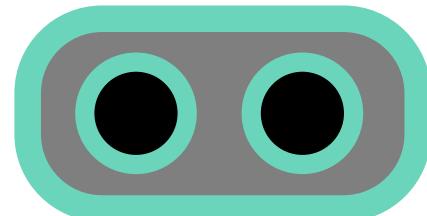
113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

scanf("Hello %d",&a);



What is going on with scanf?

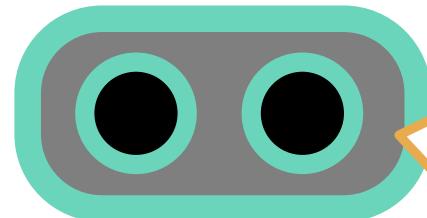
113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

`scanf("Hello %d",&a);`



Hmm ... you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a



What is going on with scanf?

113

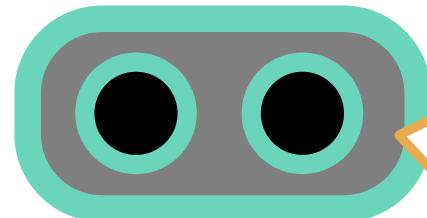
Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

scanf("Hello %d",&a);

Use printf to print and scanf to read



Hmm ... you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

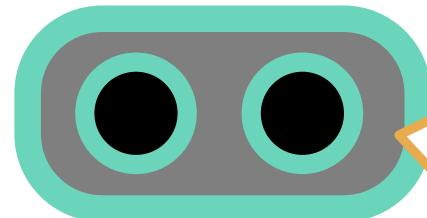
He treats all whitespace characters the same when integers are being input

scanf will never print anything

scanf("Hello %d",&a);

Use printf to print and scanf to read

Try out what happens with the following



Hmm ... you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

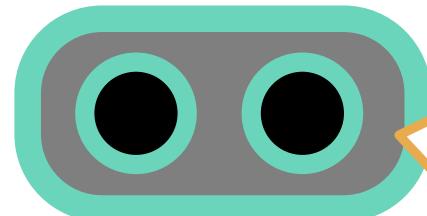
scanf will never print anything

`scanf("Hello %d",&a);`

Use printf to print and scanf to read

Try out what happens with the following

`scanf("%d %d",&a,&b);`



Hmm ... you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

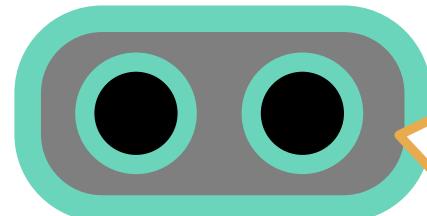
`scanf("Hello %d",&a);`

Use printf to print and scanf to read

Try out what happens with the following

`scanf("%d %d",&a,&b);`

`scanf("%d,%d",&a,&b);`



Hmm ... you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```

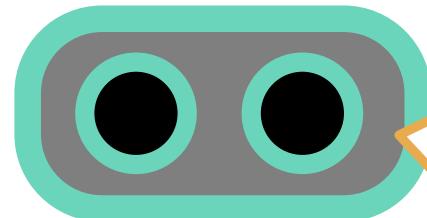
Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
```

```
scanf("%d,%d",&a,&b);
```

```
scanf("%d\n%d",&a,&b);
```



Hmm ... you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```

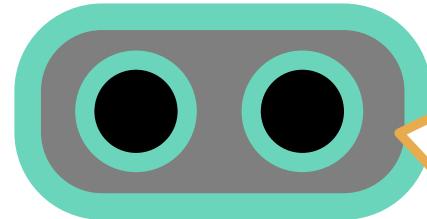
Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
```

```
scanf("%d,%d",&a,&b);
```

```
scanf("%d\n%d",&a,&b);    scanf("%d\t%d",&a,&b);
```



Hmm ... you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```

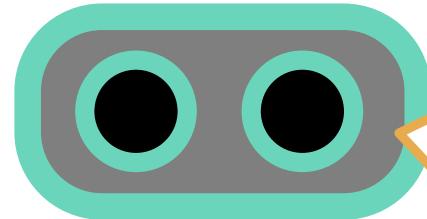
Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
```

```
scanf("%d,%d",&a,&b);
```

```
scanf("%d\n%d",&a,&b);
```



Hmm ... you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

```
scanf("\ "%d%d\ ",&a,&b);
```

```
scanf("%d\t%d",&a,&b);
```



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```

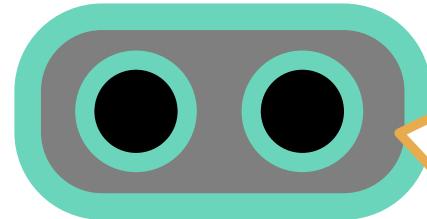
Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
```

```
scanf("%d,%d",&a,&b);
```

```
scanf("%d\n%d",&a,&b);
```

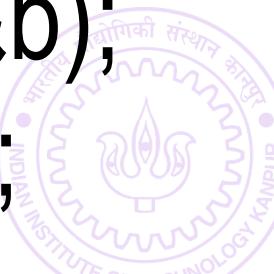


Hmm ... you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

```
scanf("%dHello%d",&a,&b);
```

```
scanf("\\"%d%d\\",&a,&b);
```

```
scanf("%d\\t%d",&a,&b);
```



What is going on with scanf?

113

Be a bit careful since Mr C is a bit careless in this matter

He tried to
integer
scanf v
scanf("Hello %d",&a);

My advice to you is to
take input one at a time in
the beginning ☺ Try out
acrobatics in free time

Use printf to print and scanf to read

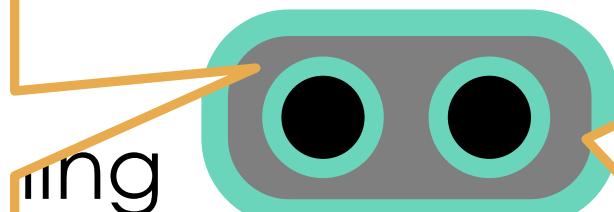
Try out what happens with the following

scanf("%d %d",&a,&b);

scanf("%d,%d",&a,&b);

scanf("%d\n%d",&a,&b);

haracters the same when

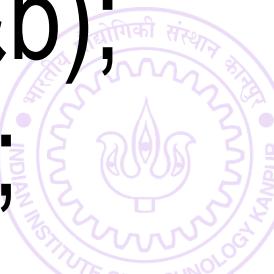


Hmm ... you are going
to write the English
word Hello followed by
space followed by an
integer. I will store the
value of that integer in a

scanf("%dHello%d",&a,&b);

scanf("\\""%d%d\\\"",&a,&b);

scanf("%d\\t%d",&a,&b);

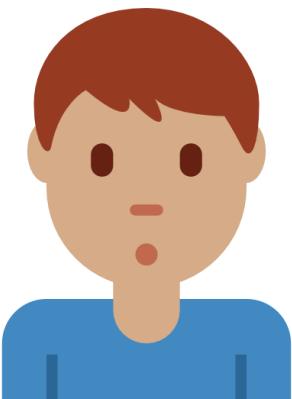


What's in a Name?



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names

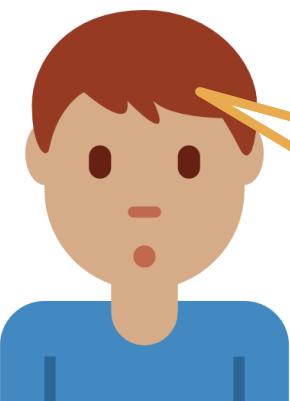


There are
illegal names?



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names



There are
illegal names?

Imagine if your parents
had named you “Hello”



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names



Imagine if your parents
had named you “Hello”



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names



My friends would greet
me as Hello Hello?

Imagine if your parents
had named you “Hello”



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names



My friends would greet
me as Hello Hello?

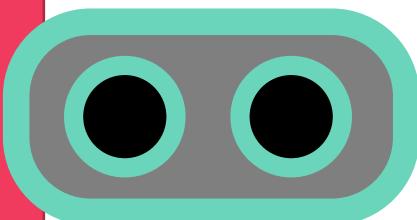
Wouldn't be nice, right?
Mr C thinks so too.

Imagine if your parents
had named you "Hello"



What's in a Name?

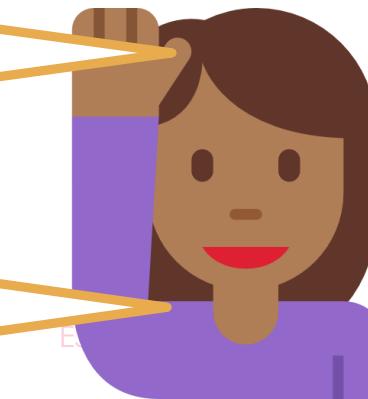
- Mr. C really does not care what we name our variables so long as they are legal names



My friends would greet
me as Hello Hello?

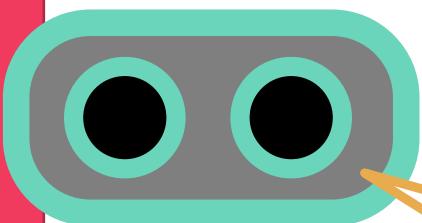
Wouldn't be nice, right?
Mr C thinks so too.

Imagine if your parents
had named you "Hello"



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names

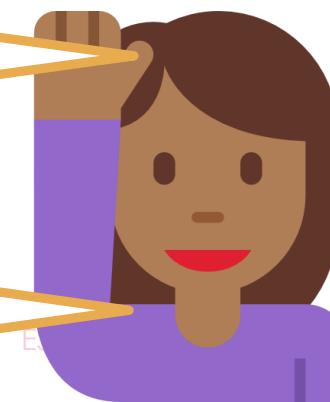


She's right. Some variable names are illegal

My friends would greet me as Hello Hello?

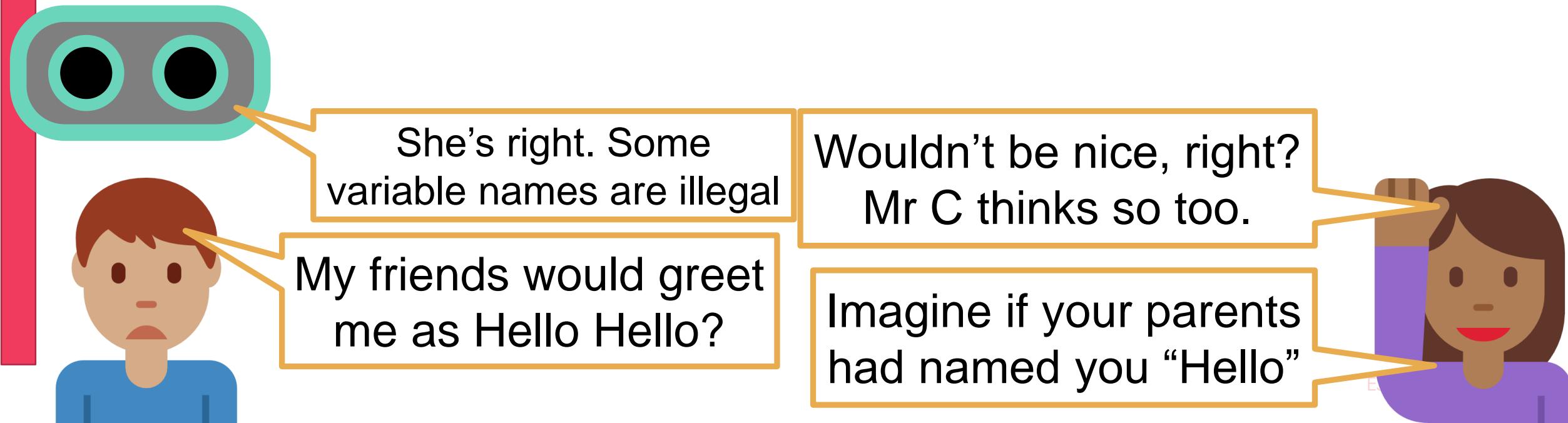
Wouldn't be nice, right? Mr C thinks so too.

Imagine if your parents had named you "Hello"



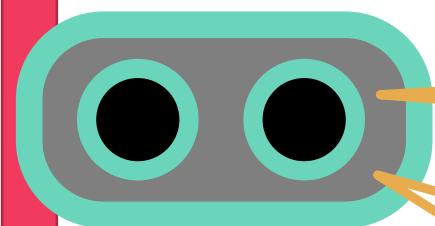
What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names
- Certain names are called “reserved keywords”



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names
- Certain names are called “reserved keywords”



Cannot use reserved
keywords as variable names

She's right. Some
variable names are illegal

My friends would greet
me as Hello Hello?

Wouldn't be nice, right?
Mr C thinks so too.

Imagine if your parents
had named you “Hello”



Reserved Keywords

140



ESC101: Fundamentals
of Computing

Reserved Keywords

auto, break, case, char, const, continue, default, do,
double, else, enum, extern, float, for, goto, if, int, long,
register, return, short, signed, sizeof, static, struct, switch,
typedef, union, unsigned, void, volatile, while



Reserved Keywords

auto, break, case, char, const, continue, default, do,
double, else, enum, extern, float, for, goto, if, int, long,
register, return, short, signed, sizeof, static, struct, switch,
typedef, union, unsigned, void, volatile, while

Prutor shows keywords in a different color ☺



Reserved Keywords

auto, break, case, char, const, continue, default, do,
double, else, enum, extern, float, for, goto, if, int, long,
register, return, short, signed, sizeof, static, struct, switch,
typedef, union, unsigned, void, volatile, while

Prutor shows keywords in a different color ☺

Have already seen int and return in use



Reserved Keywords

auto, break, case, char, const, continue, default, do,
double, else, enum, extern, float, for, goto, if, int, long,
register, return, short, signed, sizeof, static, struct, switch,
typedef, union, unsigned, void, volatile, while

Prutor shows keywords in a different color 😊

Have already seen int and return in use

Will soon see use for long, float



Reserved Keywords

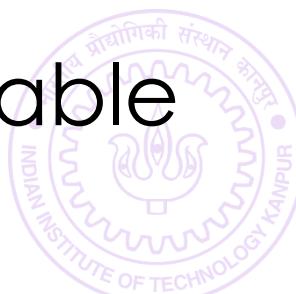
auto, break, case, char, const, continue, default, do,
double, else, enum, extern, float, for, goto, if, int, long,
register, return, short, signed, sizeof, static, struct, switch,
typedef, union, unsigned, void, volatile, while

Prutor shows keywords in a different color ☺

Have already seen int and return in use

Will soon see use for long, float

By the end of course, you will become very comfortable
with most of these keywords ☺



How to give Names in C

146



ESC101: Fundamentals
of Computing

How to give Names in C

146

Mr C calls names we give to our variables as *identifiers*



How to give Names in C

146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

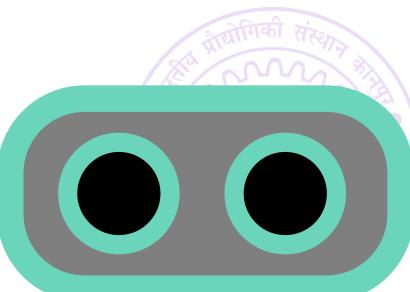


How to give Names in C

146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive



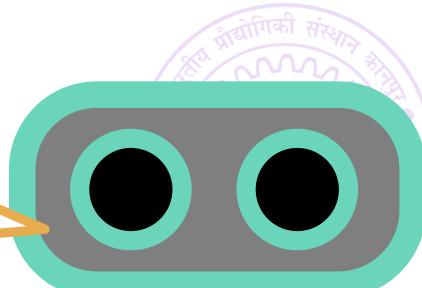
How to give Names in C

146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

For me, the names temp,
Temp, TEMP, TeMp are all
different variable names



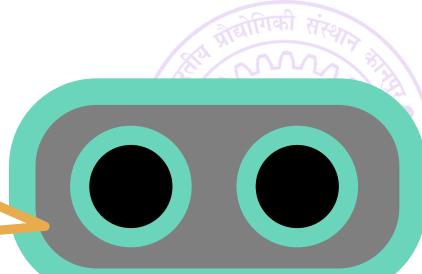
How to give Names in C

146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

For me, the names temp,
Temp, TEMP, TeMp are all
different variable names



How to give Names in C

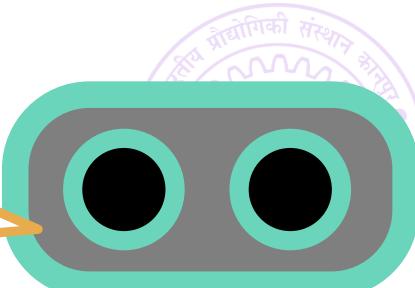
146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

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

For me, the names temp,
Temp, TEMP, TeMp are all
different variable names



How to give Names in C

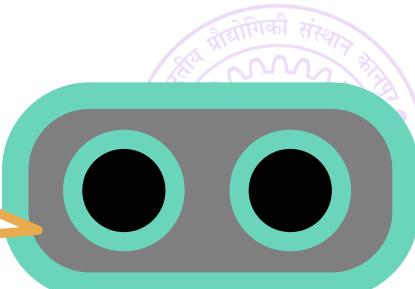
146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

```
#include <stdio.h>  
int main(){
```

For me, the names temp,
Temp, TEMP, TeMp are all
different variable names



How to give Names in C

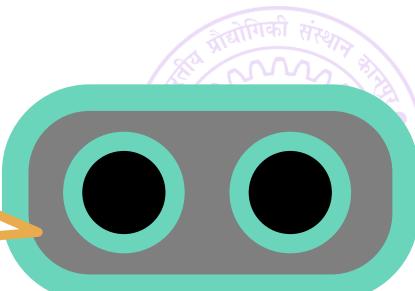
146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

```
#include <stdio.h>  
  
int main(){  
    int temp, TEMP, Temp, TeMp;
```

For me, the names temp,
Temp, TEMP, TeMp are all
different variable names



How to give Names in C

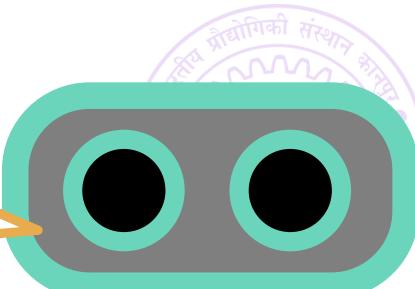
146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

```
#include <stdio.h>  
  
int main(){  
    int temp, TEMP, Temp, TeMp;  
    return 0;
```

For me, the names temp, Temp, TEMP, TeMp are all different variable names



How to give Names in C

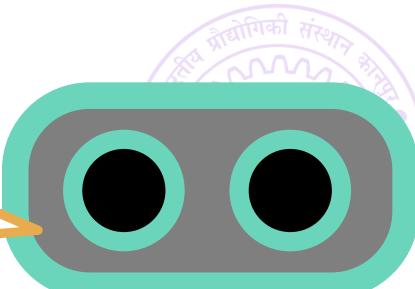
146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

```
#include <stdio.h>  
  
int main(){  
    int temp, TEMP, Temp, TeMp;  
    return 0;  
}
```

For me, the names temp, Temp, TEMP, TeMp are all different variable names



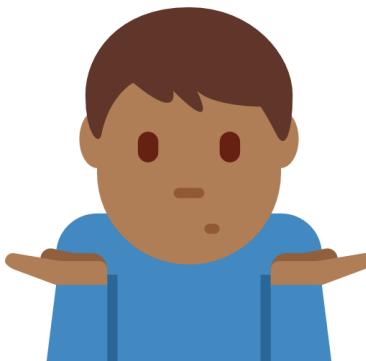
How to give Names in C

146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

```
#include <stdio.h>  
  
int main(){  
    int temp, TEMP, Temp, TeMp;  
    return 0;  
}
```



For me, the names temp, Temp, TEMP, TeMp are all different variable names



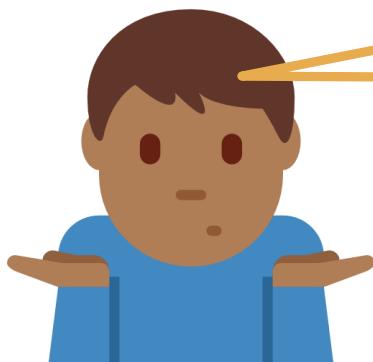
How to give Names in C

146

Mr C calls names we give to our variables as *identifiers*

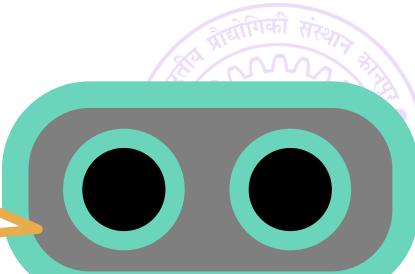
Names are case-sensitive

```
#include <stdio.h>  
  
int main(){  
    int temp, TEMP, Temp, TeMp;  
    return 0;  
}
```



So this program is fine?

For me, the names temp,
Temp, TEMP, TeMp are all
different variable names



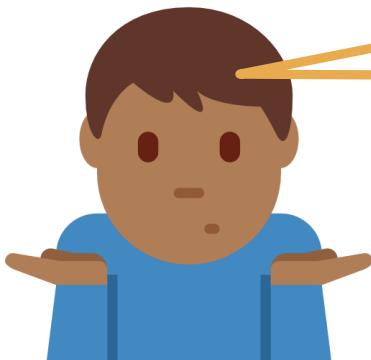
How to give Names in C

146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

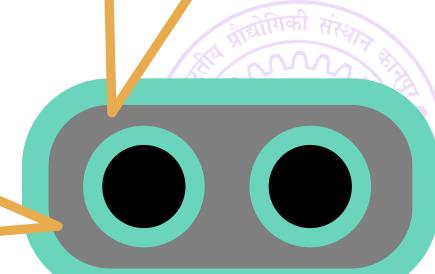
```
#include <stdio.h>  
  
int main(){  
    int temp, TEMP, Temp, TeMp;  
    return 0;  
}
```



Yes, but not advisable.
May make mistakes,
confuse others

So this program is fine?

For me, the names temp,
Temp, TEMP, TeMp are all
different variable names



How to give Names in C

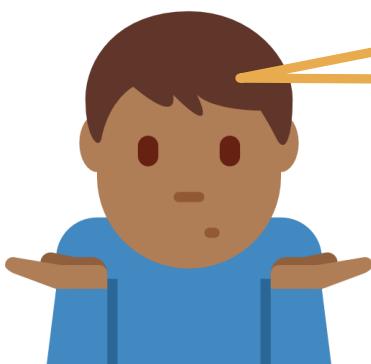
146

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

Names can contain A-Z, a-z, 0-9, underscore _

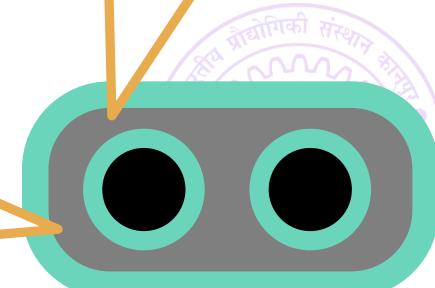
```
#include <stdio.h>  
  
int main(){  
    int temp, TEMP, Temp, TeMp;  
    return 0;  
}
```



Yes, but not advisable.
May make mistakes,
confuse others

So this program is fine?

For me, the names temp,
Temp, TEMP, TeMp are all
different variable names



How to give Names in C

146

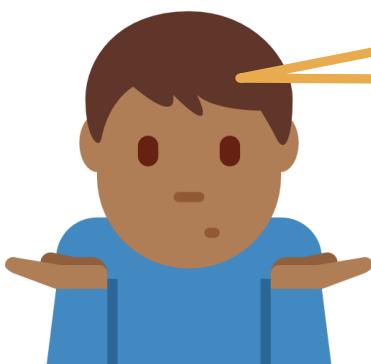
Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

Names can contain A-Z, a-z, 0-9, underscore _

Names cannot start with a digit

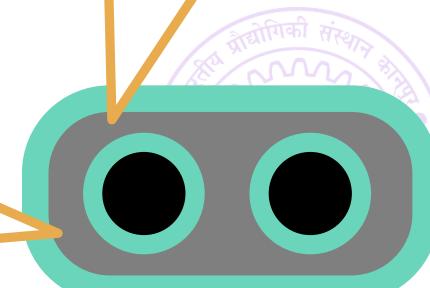
```
#include <stdio.h>  
  
int main(){  
    int temp, TEMP, Temp, TeMp;  
    return 0;  
}
```



Yes, but not advisable.
May make mistakes,
confuse others

So this program is fine?

For me, the names temp,
Temp, TEMP, TeMp are all
different variable names



Writing pretty code is an art

162



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



iitk1,iitk_1,me_iitk,
iitk_one, iitK



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



iitk1,iitk_1,me_iitk,
iitk_one, iitK



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



iitk1,iitk_1,me_iitk,
iitk_one, iitK



Make names meaningful i.e. hzoxy is valid name but ☹



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



iitk1,iitk_1,me_iitk,
iitk_one, iitK



Make names meaningful i.e. hzoxy is valid name but ☹

Programmers are artists and differ on what is more pretty



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



iitk1,iitk_1,me_iitk,
iitk_one, iitK



Make names meaningful i.e. hzoxy is valid name but ☹

Programmers are artists and differ on what is more pretty
numFirst, numSecond



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



iitk1,iitk_1,me_iitk,
iitk_one, iitK



Make names meaningful i.e. hzoxy is valid name but ☹

Programmers are artists and differ on what is more pretty

numFirst, numSecond num_first, num_second



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



iitk1,iitk_1,me_iitk,
iitk_one, iitK



Make names meaningful i.e. hzoxy is valid name but ☹

Programmers are artists and differ on what is more pretty

numFirst, numSecond num_first, num_second

Make names not too long e.g. tmp vs temporary



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



iitk1,iitk_1,me_iitk,
iitk_one, iitK



Make names meaningful i.e. hzoxy is valid name but ☹

Programmers are artists and differ on what is more pretty

numFirst, numSecond num_first, num_second

Make names not too long e.g. tmp vs temporary

You will gradually get used to many artistic styles



Writing pretty code is an art

162

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,



iitk1,iitk_1,me_iitk,
iitk_one, iitK



Make names meaningful i.e. hzoxy is valid name but ☹

Programmers are artists and differ on what is more pretty

numFirst, numSecond num_first, num_second

Make names not too long e.g. tmp vs temporary

You will gradually get used to many artistic styles

tmp, iter, var, fun, ref



Beyond Integers



Beyond Integers

- Lots of fun possible with integers alone



Beyond Integers

- Lots of fun possible with integers alone
 - However, the box storing integers is actually not very big



Beyond Integers

- Lots of fun possible with integers alone
 - However, the box storing integers is actually not very big



Beyond Integers

- Lots of fun possible with integers alone
 - However, the box storing integers is actually not very big
 - Can only store integers between -2,147,483,648 and 2,147,483,647



Beyond Integers

- Lots of fun possible with integers alone
 - However, the box storing integers is actually not very big
 - Can only store integers between -2,147,483,648 and 2,147,483,647
 - Some years ago the range was much smaller -32,768 and 32,767



Beyond Integers

- Lots of fun possible with integers alone
 - However, the box storing integers is actually not very big
 - Can only store integers between -2,147,483,648 and 2,147,483,647
 - Some years ago the range was much smaller -32,768 and 32,767
 - Computers have become really powerful in last 5-10 years



Beyond Integers

- Lots of fun possible with integers alone
 - However, the box storing integers is actually not very big
 - Can only store integers between -2,147,483,648 and 2,147,483,647
 - Some years ago the range was much smaller -32,768 and 32,767
 - Computers have become really powerful in last 5-10 years
- Also, what about real numbers (fractions etc)



Beyond Integers

- Lots of fun possible with integers alone
 - However, the box storing integers is actually not very big
 - Can only store integers between -2,147,483,648 and 2,147,483,647
 - Some years ago the range was much smaller -32,768 and 32,767
 - Computers have become really powerful in last 5-10 years
- Also, what about real numbers (fractions etc)
 - How to ask Mr C to work with a real number?



Beyond Integers

- Lots of fun possible with integers alone
 - However, the box storing integers is actually not very big
 - Can only store integers between -2,147,483,648 and 2,147,483,647
 - Some years ago the range was much smaller -32,768 and 32,767
 - Computers have become really powerful in last 5-10 years
- Also, what about real numbers (fractions etc)
 - How to ask Mr C to work with a real number?
 - How to ask Mr C to print a real number?



Long integers

184



ESC101: Fundamentals
of Computing

Long integers

Really long – can store integers between



Long integers

184

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

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



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
```



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
```



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
```



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
    printf("My first long int %ld", a);
```



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
    printf("My first long int %ld", a);
    return 0;
```



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
    printf("My first long int %ld", a);
    return 0;
}
```



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
    printf("My first long int %ld", a);
    return 0;
}
```

Comment



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
    printf("My first long int %ld", a);
    return 0;
}
```

Comment

%ld



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
    printf("My first long int %ld", a);
    return 0;
}
```

Comment

%ld

Integer arithmetic applies to
long int as well +, -, /, *, %, ()



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
    printf("My first long int %ld", a);
    return 0;
}
```

Comment

%ld

Integer arithmetic applies to
long int as well +, -, /, *, %, ()

Try them out on Prutor



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
    printf("My first long int %ld", a);
    return 0;
}
```

Comment

%ld

Integer arithmetic applies to
long int as well +, -, /, *, %, ()

Try them out on Prutor

How does long int interact
with int – int + long, int * long?



Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
    long a; // long int a; works too!
    scanf("%ld", &a);
    printf("My first long int %ld", a);
    return 0;
}
```

Comment

%ld

Integer arithmetic applies to
long int as well +, -, /, *, %, ()

Try them out on Prutor

How does long int interact
with int – int + long, int * long?

Next class ☺



How about Real numbers?

201



How about Real numbers?

201

int allows us to store, do math formulae with integers



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

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



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
```



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>  
  
int main(){  
    float a;
```



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
    float a;
    scanf("%f", &a);
```



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
    float a;
    scanf("%f", &a);
    printf("My first real %f", a);
```



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
    float a;
    scanf("%f", &a);
    printf("My first real %f", a);
    return 0;
```



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
    float a;
    scanf("%f", &a);
    printf("My first real %f", a);
    return 0;
}
```



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
    float a;
    scanf("%f", &a);
    printf("My first real %f", a);
    return 0;
}
```



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
    float a;
    scanf("%f", &a);
    printf("My first real %f", a);
    return 0;
}
```

%f

Very large range 3.4e+38



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
    float a;
    scanf("%f", &a);
    printf("My first real %f", a);
    return 0;
}
```

%f

Very large range 3.4e+38

Arithmetic operations apply
to float as well +, -, /, *, ()



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
    float a;
    scanf("%f", &a);
    printf("My first real %f", a);
    return 0;
}
```



Very large range 3.4e+38
Arithmetic operations apply
to float as well +, -, /, *, ()



How about Real numbers?

201

int allows us to store, do math formulae with integers

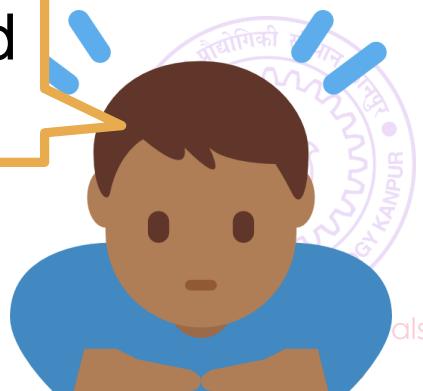
float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
    float a;
    scanf("%f", &a);
    printf("My first real %f", a);
    return 0;
}
```

%f

Very large range 3.4e+38
Arithmetic operations apply
to float as well +, -, /, *, ()

What happened
to %?



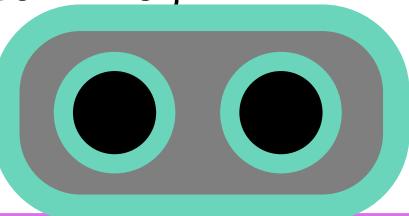
How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>  
  
int main(){  
    float a;  
  
    scanf("%f", &a);  
    printf("My first real %f", a);  
  
    return 0;  
}
```



%f

Very large range 3.4e+38
Arithmetic operations apply
to float as well +, -, /, *, ()

What happened
to %?



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>  
  
int main(){  
    float a;  
  
    scanf("%f", &a);  
  
    printf("My first real number is %f", a);  
  
    return 0;  
}
```

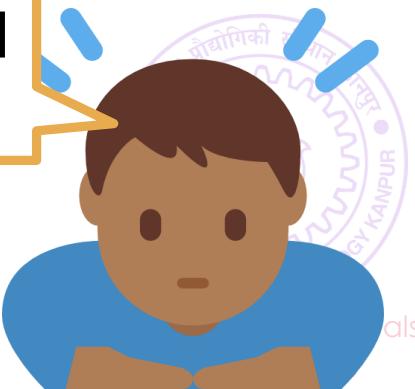
%f

☺ Did you ever do
remainders with real
numbers in school?

Very large range 3.4e+38

Arithmetic operations apply
to float as well +, -, /, *, ()

What happened
to %?



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>  
  
int main(){  
    float a;  
  
    scanf("%f", &a);  
  
    printf("My first real number is %f", a);  
  
    return 0;  
}
```

%f

☺ Did you ever do
remainders with real
numbers in school?

Very large range 3.4e+38
Arithmetic operations apply
to float as well +, -, /, *, ()



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>  
  
int main(){  
    float a;  
  
    scanf("%f", &a);  
  
    printf("My first real number is %f", a);  
  
    return 0;  
}
```

%f

☺ Did you ever do
remainders with real
numbers in school?

Very large range 3.4e+38
Arithmetic operations apply
to float as well +, -, /, *, ()

I remember. Remainders
make sense for integers,
not for real numbers



How about Real numbers?

201

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>  
  
int main(){  
    float a;  
    scanf("%f", &a);  
    printf("My first real number is %f", a);  
    return 0;  
}
```

%f

☺ Did you ever do
remainders with real
numbers in school?

Very large range 3.4e+38

Arithmetic operations apply
to float as well +, -, /, *, ()

Try them out on Prutor

I remember. Remainders
make sense for integers,
not for real numbers

