

Taking Inputs (scanf)

ESC101: Fundamentals of Computing

Nisheeth

Announcements

- Graded labs starting today
- Prutor accounts: Hopefully everyone now has a working Prutor account (accessible via your CC email id and CC password)
 - If not, please arrive at the lab early (by 1:45pm) and we will create your account on the spot



Recap

Every C program's **entry point** (program's **execution** starts here) is the **main** function with **return type integer**

Tells C compiler to include the **standard input/output library `stdio.h`** (collection of **functions** such as `printf`, `scanf`, etc)

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

```
int main() {
```

main function must open with **left curly brace** {

printf function prints a user specified output

```
printf("Welcome to ESC101");
```

```
return 0;
```

main function must close with **right curly brace** }

The main function must return an integer (return 0 means **successful execution** of program)

Every statement in a C program must end with **semi-colon** ;

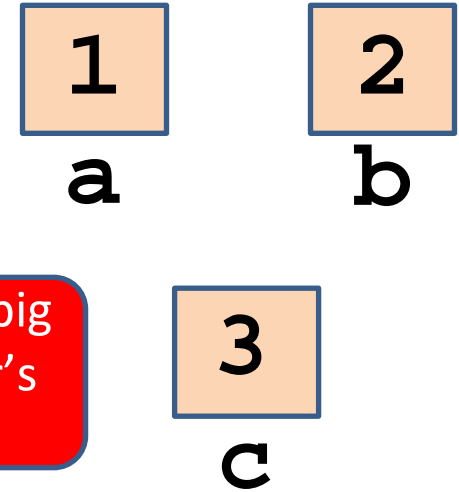
`printf("Welcome to ESC101")` and `return 0` are '**statements**' in the above code. Each C statement must end with a semi-colon ;

Recap

```
# include <stdio.h>
int main () {
    int a = 1;
    int b = 2;
    int c;
    c = a + b;
    printf("Result is %d", c);
    return 0;
}
```

Each variable's **declaration** creates a "box" big enough to store it at a **location** in computer's main memory (RAM)

Assigning a value to the variable writes that value in the box



= and + are "operators"

= is **assignment** operator

+ is **addition** operator

a+b is an "expression"

The program prints the message "Result is 3"

What's Wrong Here?

```
# include <stdio.h>
int main () {
    int a = 1;
    int b = 2;
    c = a + b;
    printf("Result is %d", c);
    return 0;
}
```



Can't assign a value to c since it has not been declared yet

Will NOT Compile



What's Wrong Here?

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



Can't use variables a and b in this assignment operation since a and b have not been assigned a value (**initialized**) in this program

Will Compile but
will print garbage



What's Wrong Here?

```
# include <stdio.h>
int main () {
    int a = 1;
    int b = 2;
    int c;
    printf("Result is %d", c);
    return 0;
}
```

Will print some garbage value since c has not been assigned a value (**initialized**) yet



Will Compile but will print garbage



What About This?

```
# include <stdio.h>
int main () {
    int a;
    int b;
    int c;
    c = a + b;
    a = 2;
    b = 1;
    printf("Result is %d", c);
    return 0;
}
```



Can't use variables a and b in this assignment operation since a and b have not been assigned a value (**initialized**) in this program **yet**. Assigning a and b values **later** does NOT solve this problem

Will Compile but will print garbage

Lesson Learned

Declare and initialize (or assign values to) your variables properly before their use



Recap: Alphabet and Keywords of C

- C programs can be written using the following alphabet

➤ A B Z

➤ a b z

➤ 0 1 9

➤ Space . , : ; ‘ \$ “

➤ # % & ! _ { } [] () |


➤ + - * / =

Keywords in C

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



Naming Convention for Variables and Functions

- We have seen variables and their usage in programs
- We have seen the main and printf function (and will see various other standard functions and user-defined function later)
- Need to follow some rules for naming of variables and functions
- Names can only contain **A-Z, a-z, 0-9**, and underscore **_**
- **Can't begin** a variable's or function's name with a **number**
- A_3, abcDS2, this_variable are some valid names 
- 321, 5_r, dfd@dhr, this variable, no-entry are some not valid names

Start with number

Contains special symbol @

Contains space character

Contains hyphen ("dash") character -

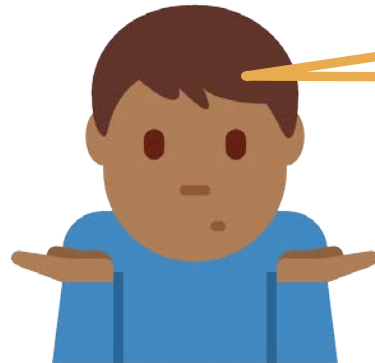


Variables and Function Names: Some Suggestions

- Should prefer short, meaningful names. Don't use C keywords.
- Multi-word name allows, e.g., firstNumber, first_number
- Advice: Use capital letters for constants (e.g., NUMBER_DAYS_JAN)
- Advice: Use small letters for variables (e.g., radius, volume)

```
#include <stdio.h>

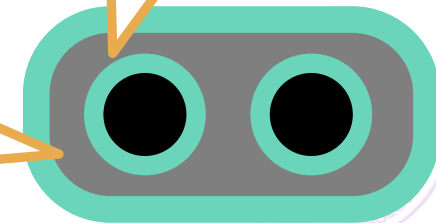
int main(){
int temp, TEMP, Temp, TeMp;
return 0;
}
```



So this program is fine?

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

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



Recap: printf and its use

Note: In some cases,
there will be no such list.
Example: printf("Hello");

```
printf(format string, list of things to print);
```

```
printf("Hello %d %d", a,b);
```

Printing some characters, such as “, new-line, %, \
requires special care (need to use escape sequences)



Reading Inputs: The **scanf** function

- Programs that don't take inputs from user can be boring
 - We saw programs to add two numbers but both had to be written into code
 - Also called “hardcoding” the inputs
 - A bit like a calculator which can only add 5 and 4
 - To add 6 and 9, write a new calculator
- Can't we ask Mr C to request us for the numbers when he is executing our requests i.e. *at runtime*?
- YES, by taking input from the user using **scanf function**



Example: Adding Two **User-provided** Numbers

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

```
int main(){
```

```
int a, b;
```

3

8

b

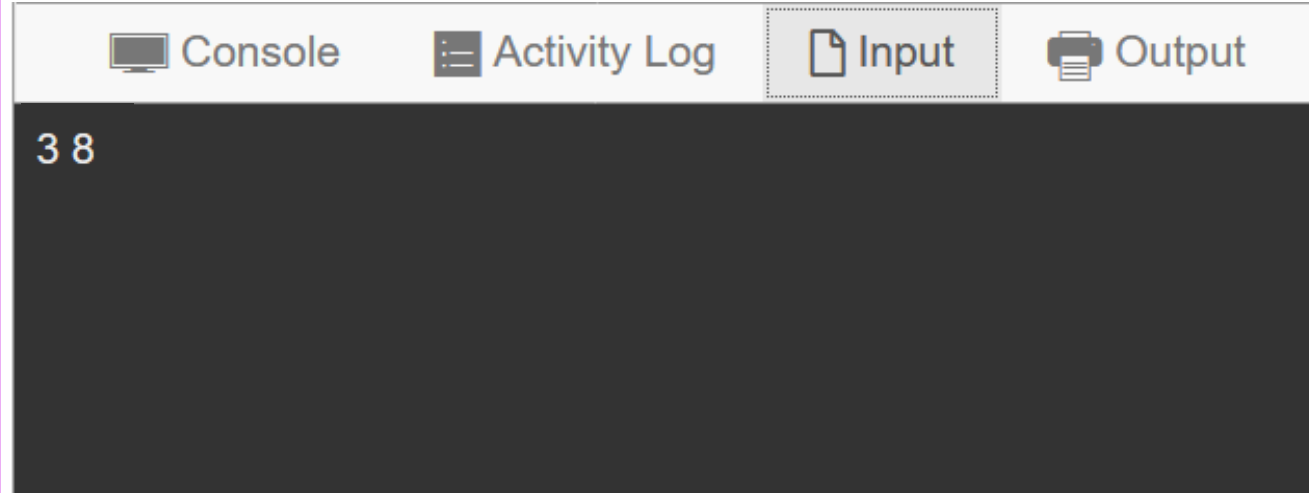
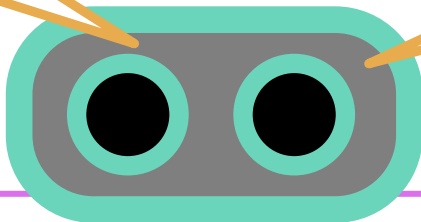
Thanks. Let me get back to work

11

Please give me input

11

```
}
```



scanf: Some Words of Caution

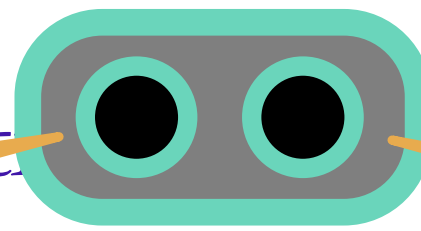
In Prutor, input has to be specified before "Execute"

The screenshot shows the Prutor IDE interface. The top menu bar includes 'Prutor ESC101', 'File', 'Run', and 'Help'. On the right, there are links for 'CodeBook', 'Practice', 'Scratchpad', and a user profile 'Piyush Rai'. The left sidebar shows 'My Works' with two files: 'test.c' and 'test2.c'. A context menu is open over 'test2.c', with options: 'Compile', 'Execute', 'Visualise', and 'Evaluate' (which is checked). The main editor displays a C program in 'test2.c':

```
1 #include <stdio.h>
2 int main () {
3     int a;
4     int b;
5     scanf("%d%d", &a,&b);
6     printf("Sum of a and b = %d",a + b);
7     return 0;
8 }
9
```

The bottom status bar shows 'Console', 'Activity Log', 'Input', and 'Output' tabs. The 'Input' tab is active, showing '2 3'. The 'Output' tab shows '[test2.c] INVALID DATE'.

scanf: Some Words of Caution



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

specified before "E"

Both work!
Experiment!

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

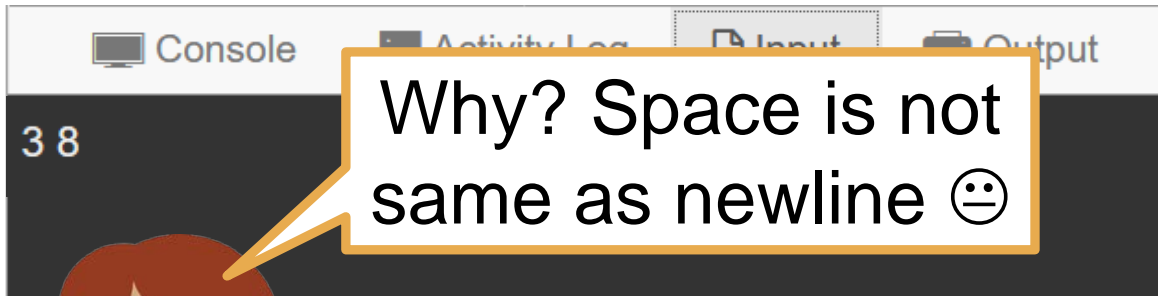
```
3 int 0a, 0b, 0c, 0d; 0
```

TAB SPACE NEWLINE

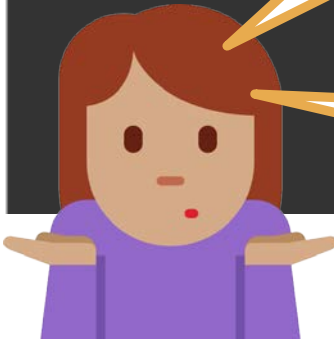
```
scanf("%d", &a);
```



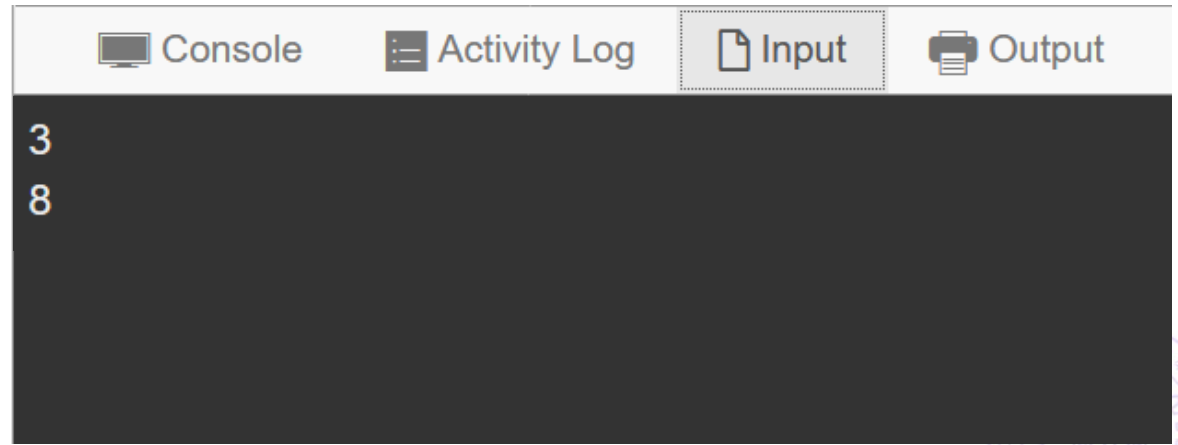
will explain what this & symbol means, in a few weeks



Why? Space is not same as newline 😊



Huh! What is a whitespace?



Taking Multiple Inputs using a Single scanf

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

3 8

a b

11

11

All look the same to me

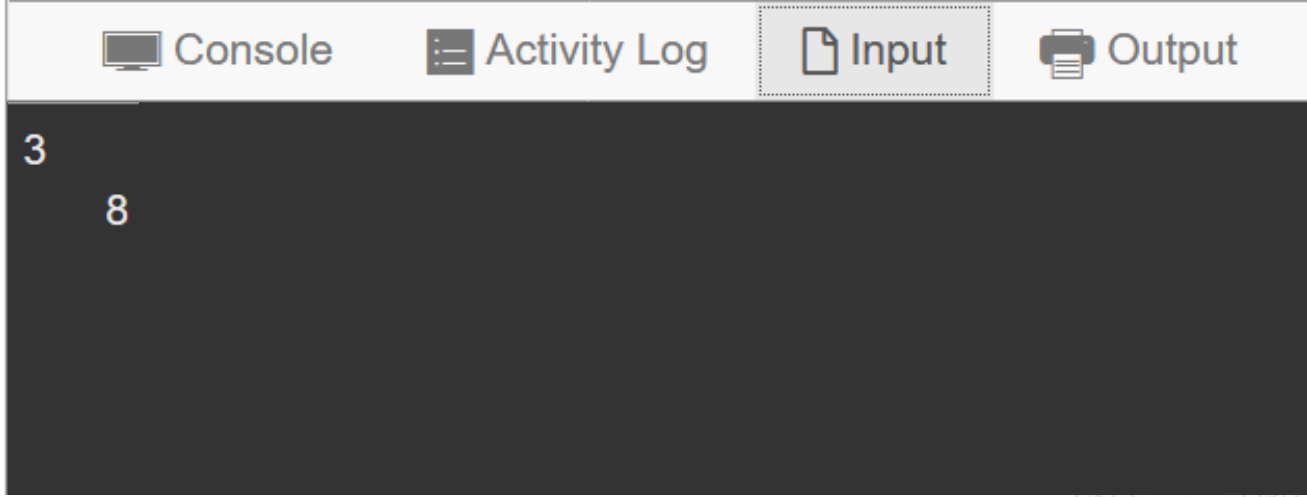
Input please

Thanks

Okay okay



Help!!!



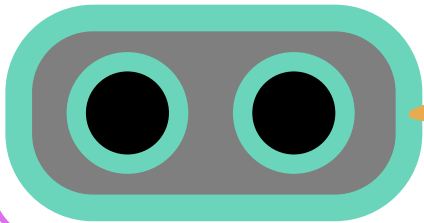
How does scanf work ?

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



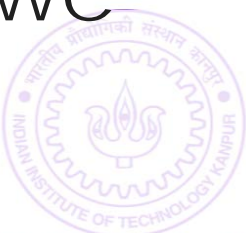
HOW WE USUALLY SPEAK TO A HUMAN

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

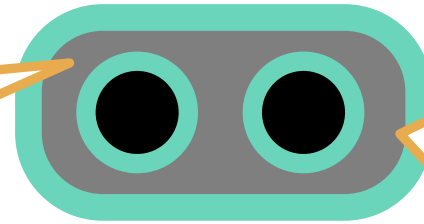


How does scanf work ?

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

He treats characters the same when

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



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

Use printf to print and scanf to read

Try out what happens with the following

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

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

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



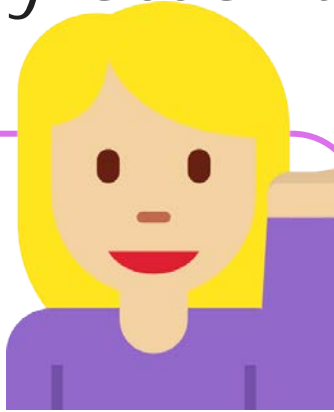
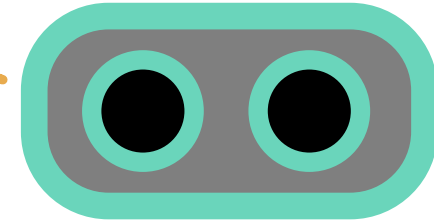
Commenting Your Code

Last week we learned about "indexing"

Let us learn about "comments" today

Absolutely essential in industry, even self projects

Okay. I will add your two numbers



How we write commented code What Mr C sees

```
int main(){
  int a;
  int b;
  a = 5, b = 4;
  int c = a + b;
  printf("c = %d",c);
  return 0;
}
```

```
int main(){
  int a; // My first int
  int b; // The other int
  // Assign them values
  a = 5, b = 4;
  int c = a + b;
  printf("c = %d",c);
  return 0;
}
```

Only humans see comments

```
int main(){
  int a;
  int b;

  a = 5, b = 4;
  int c = a + b;
  printf("c = %d",c);
  return 0;
}
```

Several Ways of Writing Comments

Since it is an art form, artists differ on what is more pretty

```
int main(){  
    int a; // My first int  
    int b; // The other int  
    // Assign them values  
    a = 5, b = 4;  
    a + b;  
    return 0;  
}
```

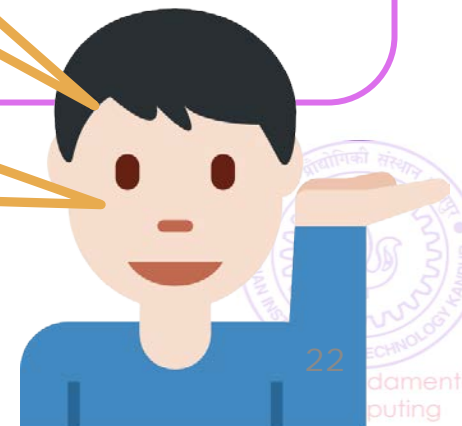
So I can mix and match?

```
int main(){  
    int a; /* My first int */  
    int b; /* The other int */  
    /* Assign them values */  
    a = 5, b = 4;  
    a + b;  
    return 0;  
}
```

Yes. In fact /* */ is used to comment several lines at once – shortcut!

Just be a bit careful. Some compilers don't understand // comments

```
int main(){  
    int a; // My first int  
    int b; // The other int  
    /* Assign them values */  
    a = 5, b = 4;  
    a + b;  
    return 0;  
}
```



More on Comments

Use comments to describe why you defined each variable and what each step of your code is doing

You will thank yourself for doing this when you are looking at your own code before the endsem exams 😊 😊

Your team members in your company/research group will also thank you

Multiline comments very handy. No need to write // on every line 😊

```
int main(){
    int a; // My first int
    int b; // The other int
    // Assign them values
    // so that I can add
    // them later on
    a = 5, b = 4;
    a + b;
    return 0;
}
```

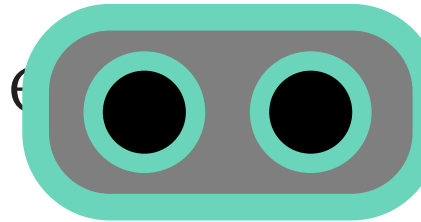
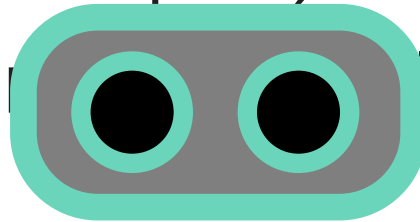
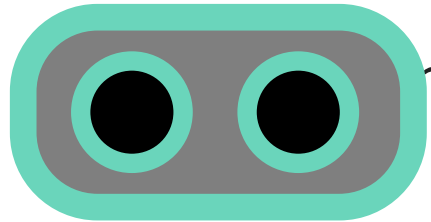
A Useful Tip While Problem-Solving

Comments can be also used to identify where is error. Mr C will tell you (compile) where he thinks the error is.

Error!

Okay!

Okay!



```
int main(){
  int a, b;
  c = a + b;
  a = 5;
  b = 4;
  return 0;
}
```

```
int main(){
  int a, b;
  // c = a + b;
  a = 5;
  b = 4;
  return 0;
}
```



```
int main(){
  // c = a + b;
  a = 5;
  b = 4;
  return 0;
}
```

Aha! I forgot to declare c



Take Care with Formulae: Using Brackets Help

Operation	C Code	a	b	c
Addition	<code>c = a + b;</code>	5	4	9
Subtraction	<code>c = a - b;</code>	4	5	-1
Multiplication	<code>c = a * b;</code>	-2	-4	8
Division	<code>c = a / b;</code>	7	2	3
Remainder	<code>c = a % b;</code>	7	2	1

Bracket, Of, Division,
Multiplication, Addition, Subtraction

Recall your BODMAS order rules from high school

Mr. C follows similar rules – will see in detail soon

Good practice to bracket your formulae

Minimize confusion as well as chances of error

Play with brackets in lab to practice

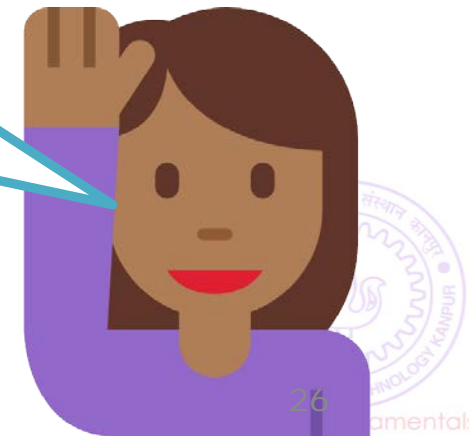
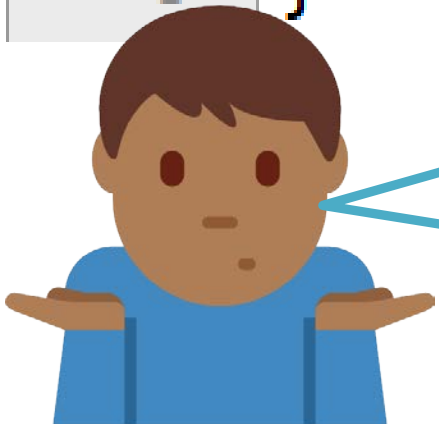
A Useful Tip While Solving Problems

```
1 #include<stdio.h>-  
2 int main(){-  
3     int x = 3;-  
4     int result;-  
5     result = 2/3*x*x*x + 2*x*x*x;-  
6     printf("The area under the curve is %d", result);-  
7     return 0;-  
8 }
```

Print your solutions to each one of these pieces to see where going wrong

Try breaking up the problem into smaller pieces

I have no idea what is going wrong here!



A Useful Tip While Solving Problems

```
1 #include<stdio.h>-  
2 int main(){-  
3     int x = 3;-  
4     int result; Equals 0  
5     result = 2/3*x*x*x + 2*x*x + 9*x;-  
6     printf("The area under the curve is %d",result);-  
7     return 0;-  
8 }
```



A Useful Tip While Solving Problems

```
1 #include<stdio.h>-  
2 int main(){-  
3     int x = 3;-  
4     int result;-  
5     result = 2/3*x*x*x + 2*x*x + 9*x;-  
6     printf("The area under the curve is %d", result);-  
7     return 0;-  
8 }
```

Replace this part by $(2*x*x*x)/3$

