

water

TODO

- prompt and validate user input
- calculate equivalent bottles
- print equivalent bottles

TODO

- prompt and validate user input
- calculate equivalent bottles
- print equivalent bottles

get_int

- CS50 Library function
- ensures that the user inputs an integer
 - positive integers
 - negative integers
 - 0
 - otherwise, user is prompted to "Ret ry:".

```
#include <cs50.h>
```

```
int main(void)
```

```
{
```

```
    get_int();
```

```
}
```

```
#include <cs50.h>
```

```
int main(void)
{
    get_int();
}
```



get_int is a
CS50 Library function;
we must #include
its declaration

```
#include <cs50.h>
```


```
int main(void)
```

```
{
```

```
    get_int();
```

```
}
```

how can we keep track
of the integer inputted
by the user?



```
#include <cs50.h>
```

```
int main(void)
```

```
{
```

```
    int n = get_int();
```

```
}
```



store the value
in variable n

while loops

```
while (condition)
{
    // something happens
}
```

do-while loops

- executes the body *at least once*

```
do
{
    // something happens
}
while (condition);
```

do-while loops

```
int n;
```



declared outside do-while scope

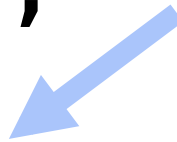
```
do
```

```
{
```

```
    n = get_int();
```

```
}
```


what should this
condition be?



```
while (n is invalid);
```

read the spec carefully!

prompt the user with
this exact string



```
Minutes: 10  
Bottles: 120
```

TODO

- prompt and validate user input
- calculate equivalent bottles**
- print equivalent bottles

detect pattern / formula

- per the spec: 1 min = 12 bottles of water
2 min = 24 bottles of water
...
5 min = 60 bottles of water

 n min = ?? bottles of water

arithmetic operations

- add: $\text{num1} + \text{num2}$
- subtract: $\text{num1} - \text{num2}$
- multiply: $\text{num1} * \text{num2}$
- divide: $\text{num1} / \text{num2}$

TODO

- prompt and validate user input
- calculate equivalent bottles
- print equivalent bottles**


```
printf("I have 3 pets\n");
```

placeholder for an integer

```
printf("I have %i pets\n", n);
```

variable to be printed

placeholder for a float

```
printf("I have %f dollars\n", n);
```

variable to be printed

placeholders

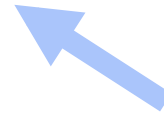
```
printf("I have %i dogs and %i cats\n", dogs, cats);
```

variables

read the spec carefully!

Minutes: 10

Bottles: 120



print the value
in this exact format

TODO

- ☑ prompt and validate user input
- ☑ calculate equivalent bottles
- ☑ print equivalent bottles

this was water