Functions

Programming in C

Shirley B. Chu
`shirley.chu@delasalle.ph`

College of Computer Studies
De La Salle University

October 15, 2020

Recall: Why functions?

- reduce redundancy

Recall: Why functions?

- reduce redundancy
- improve readability and understandability

Recall: Why functions?

- reduce redundancy
- improve readability and understandability
- reduce complexity

Recall: Why functions?

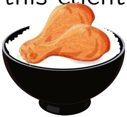
- reduce redundancy
- improve readability and understandability
- reduce complexity
- improve performance

Recall: Why functions?

- reduce redundancy
- improve readability and understandability
- reduce complexity
- improve performance
- isolate complex processes

Example

A certain restaurant MyKitchen serves only Fried Chicken rice. A client dines in this restaurant, and specifies the number of Fried Chicken rice he wishes to order. Write a program that generates the receipt for this client.

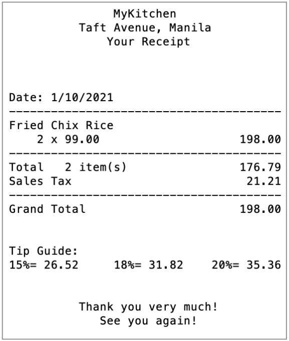


```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 10/10/2020
----------------------------------------
Fried Chix Rice
   10 x 99.00                     990.00
----------------------------------------
Total  10 item(s)                 883.93
Sales Tax                         106.07
----------------------------------------
Grand Total                       990.00


Tip Guide:
15%=132.59     18%=159.11     20%=176.79


          Thank you very much!
             See you again!
```

Sales Tax is 12%, and tip is calculated based on the gross amount (before tax). Total number of characters per line is 40.

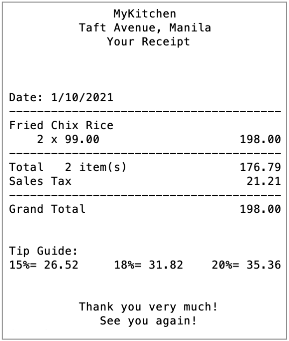
Example



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

Example

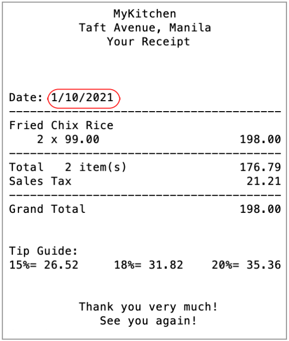


```
            MyKitchen
       Taft Avenue, Manila
          Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs

Example

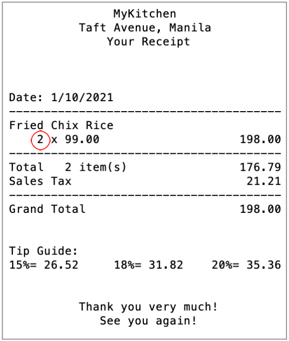


```
              MyKitchen
         Taft Avenue, Manila
            Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
   - date

Example

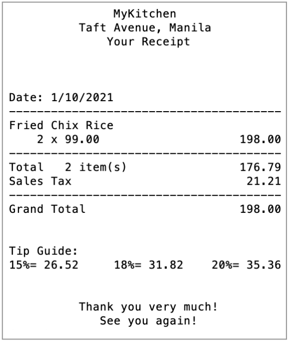


```
              MyKitchen
         Taft Avenue, Manila
             Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
   - date
   - number of orders

Example



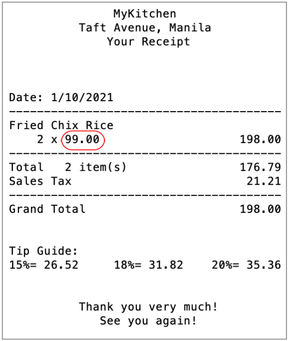
```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
   - date
   - number of orders

Known data:

Example



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52    18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

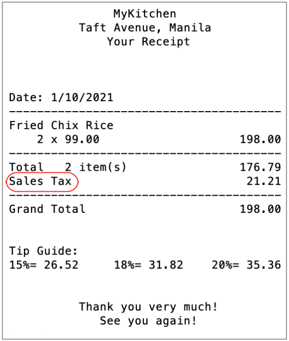
Tasks:

1. Ask for inputs
   - date
   - number of orders

Known data:

- amount per order

Example



```
               MyKitchen
          Taft Avenue, Manila
             Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total    2 item(s)               176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

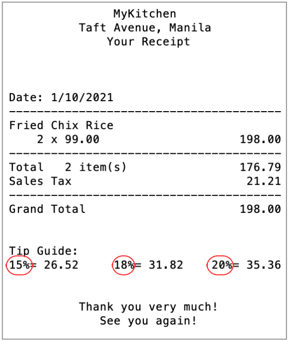
Tasks:

1. Ask for inputs
   - date
   - number of orders

Known data:

- amount per order
- sales tax is 12%

Example



```
              MyKitchen
         Taft Avenue, Manila
            Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52      18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

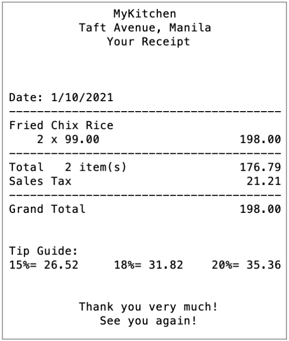
Tasks:

1. Ask for inputs
   - date
   - number of orders

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Example



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

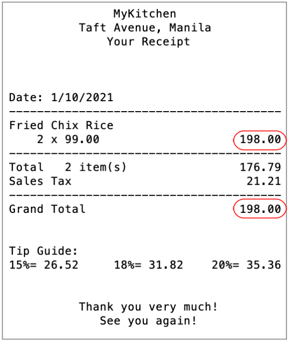
Tasks:

1. Ask for inputs
   - date
   - number of orders
2. Compute

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Example



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

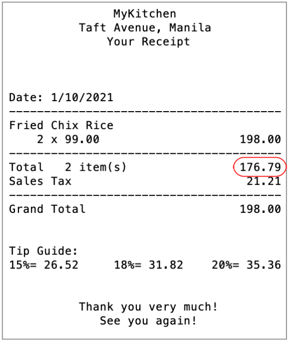
Tasks:

1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Example



```
              MyKitchen
         Taft Avenue, Manila
             Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

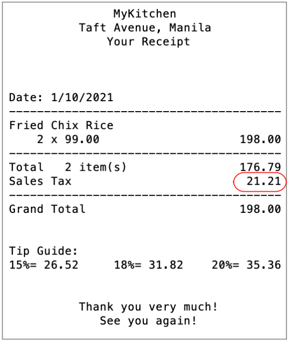
Tasks:

1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount
     (amount before tax)

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Example



```
              MyKitchen
         Taft Avenue, Manila
             Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

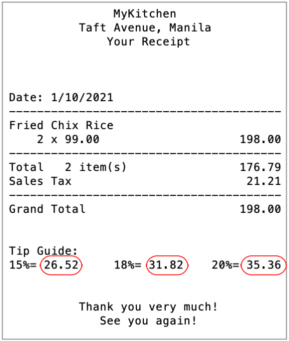
Tasks:

1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount (amount before tax)
   - tax

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Example



```
                MyKitchen
           Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total    2 item(s)               176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

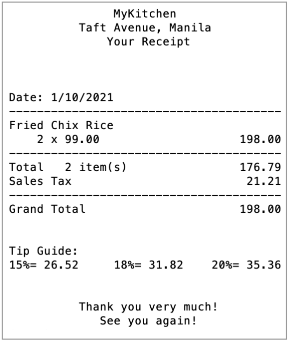
Tasks:

1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount
     (amount before tax)
   - tax
   - tips

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Example



```
                 MyKitchen
             Taft Avenue, Manila
                Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


            Thank you very much!
               See you again!
```

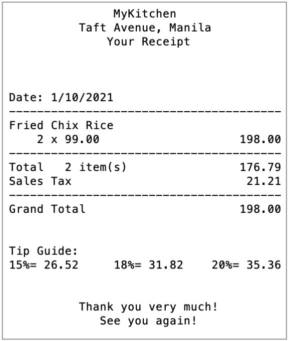
Tasks:

1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount
     (amount before tax)
   - tax
   - tips
3. Display receipt

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Example



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

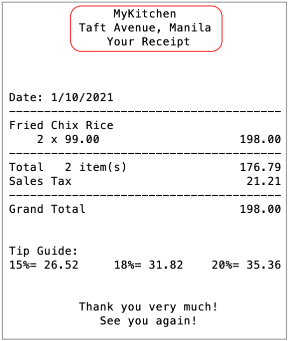
1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount
     (amount before tax)
   - tax
   - tips
3. Display receipt

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Characters per line: 40

Example



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

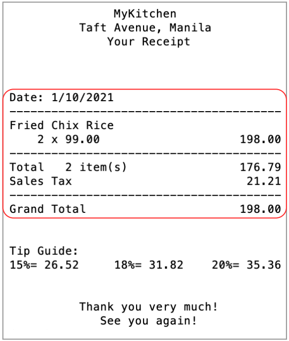
1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount (amount before tax)
   - tax
   - tips
3. Display receipt
   - header

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Characters per line: 40

Example



```
               MyKitchen
          Taft Avenue, Manila
             Your Receipt


Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

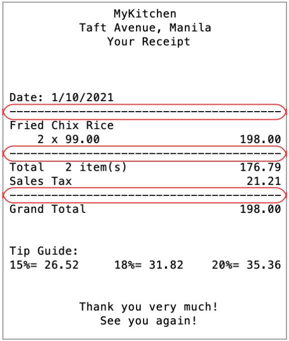
1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount
     (amount before tax)
   - tax
   - tips
3. Display receipt
   - header
   - details

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Characters per line: 40

Example



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

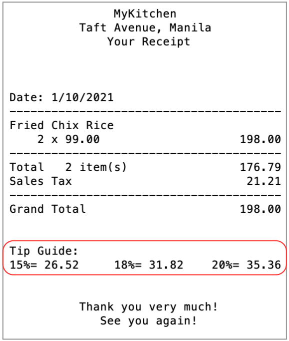
1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount
(amount before tax)
   - tax
   - tips
3. Display receipt
   - header
   - details
   - divider

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Characters per line: 40

Example



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

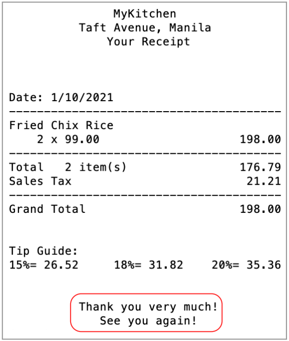
1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount
     (amount before tax)
   - tax
   - tips
3. Display receipt
   - header
   - details
   - divider
   - tip guide

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Characters per line: 40

Example



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

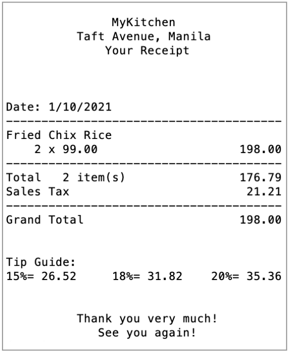
1. Ask for inputs
   - date
   - number of orders
2. Compute
   - total price
   - gross amount
     (amount before tax)
   - tax
   - tips
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

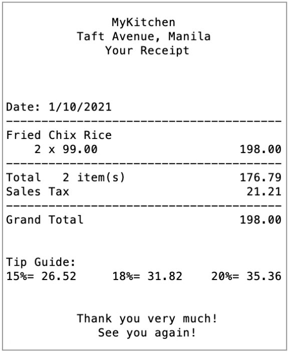
Characters per line: 40

Let’s code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

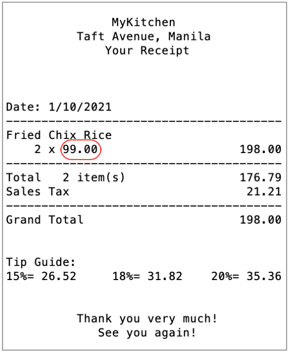
Let’s code



```
          MyKitchen
     Taft Avenue, Manila
        Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

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

Let’s code



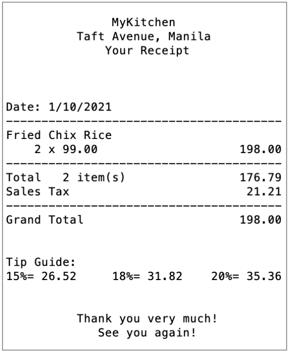
```
                 MyKitchen
            Taft Avenue, Manila
               Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total    2 item(s)               176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


            Thank you very much!
               See you again!
```

```c
#include<stdio.h>

#define AMOUNT 99.00
```

Known data:

- amount per order

Let's code



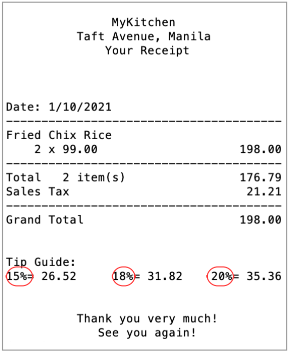
```
              MyKitchen
         Taft Avenue, Manila
            Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
```

Known data:

- amount per order
- sales tax is 12%

Let’s code



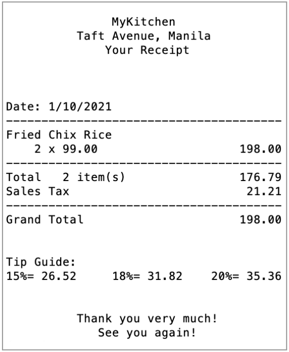
```
              MyKitchen
         Taft Avenue, Manila
            Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total    2 item(s)               176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52      18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20
```

Known data:

- amount per order
- sales tax is 12%
- tips: 15%, 18%, 20%

Let’s code

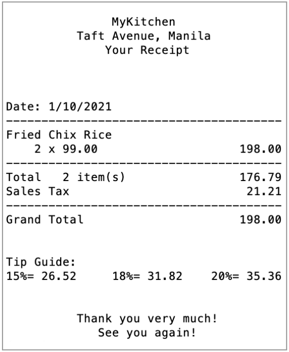


```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20
```

Tasks:

Let’s code



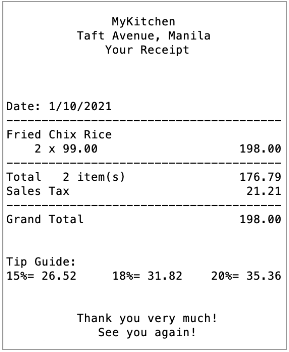
```
               MyKitchen
         Taft Avenue, Manila
             Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

int main ()
{









    return 0;
}
```

Tasks:

1. Ask for inputs

Let’s code



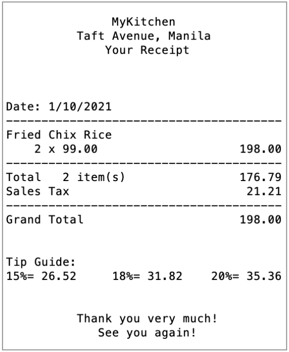
```
              MyKitchen
         Taft Avenue, Manila
            Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total    2 item(s)               176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

int main ()
{
    int nDate;









    return 0;
}
```

Tasks:

1. Ask for inputs
   - date

Let's code



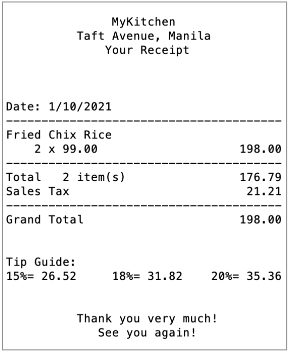
```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

int main ()
{
    int nDate;


    printf ("Date:  ");
    scanf ("%d", &nDate);




    return 0;
}
```

Tasks:

1. Ask for inputs
   - date

Let’s code



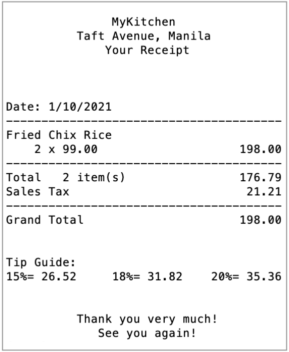
```
                MyKitchen
           Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

int main ()
{
    int nDate;
    int nOrders;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    return 0;
}
```

Tasks:

1. Ask for inputs
   - date
   - number of orders

Let’s code



```
                MyKitchen
           Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total    2 item(s)               176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52      18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

int main ()
{
    int nDate;
    int nOrders;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    return 0;
}
```

Tasks:

1. Ask for inputs
2. Compute
   - total price
   - gross amount (amount before tax)
   - tax
   - tips

General syntax of a function declaration

returnType functionName (*type1 param1, type2 param2, ...*)

General syntax of a function declaration

- specifies the name of the function
- is an identifier, usually follows verb-noun format e.g. `computeAverage()`, `multiplyXY()`

returnType *functionName* (*type1 param1*, *type2 param2*, ...)

General syntax of a function declaration

returnType functionName (*type1 param1, type2 param2, ...*)

e.g. `a` is a variable; `a()` is a function

General syntax of a function declaration

returnType functionName (*type1 param1, type2 param2, ...*)

formal parameters

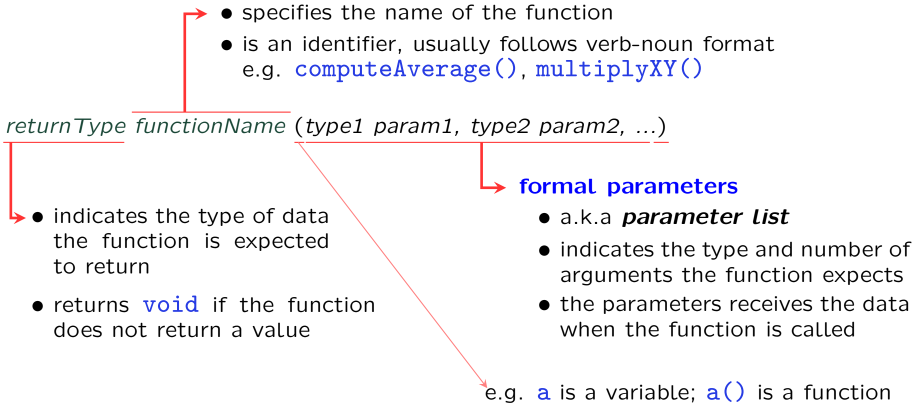
- a.k.a ***parameter list***
- indicates the type and number of arguments the function expects
- the parameters receives the data when the function is called

General syntax of a function declaration

returnType functionName (*type1 param1*, *type2 param2*, ...)

- indicates the type of data the function is expected to return
- returns `void` if the function does not return a value

General syntax of a function declaration



returnType functionName (type1 param1, type2 param2, ...)

- specifies the name of the function
- is an identifier, usually follows verb-noun format e.g. `computeAverage()`, `multiplyXY()`

- indicates the type of data the function is expected to return
- returns `void` if the function does not return a value

formal parameters

- a.k.a ***parameter list***
- indicates the type and number of arguments the function expects
- the parameters receives the data when the function is called

e.g. `a` is a variable; `a()` is a function

User-defined functions

- created by the programmer
- does a specific task
- convention for names: verb-noun pair,
  e.g. `getAverage()`, `displayTitle()`

User-defined functions

Recall: *returnType functionName* (*type1 param1, type2 param2, ...*)

User-defined functions

Recall: *returnType functionName* (*type1 param1, type2 param2, ...*)

Task: Compute total price

User-defined functions

Recall: *returnType* *functionName* (*type1 param1*, *type2 param2*, *...*)

Task: Compute total price

User-defined functions

Recall: *returnType functionName* (*type1 param1*, *type2 param2*, *...*)

Task: Compute total price

```
computeTotal (            )
```

User-defined functions

Recall: *returnType functionName* (*type1 param1, type2 param2, ...*)

Task: Compute total price

```
computeTotal (          )
```

User-defined functions

Recall: *returnType functionName* (*type1 param1, type2 param2, ...*)

Task: Compute total price

```
computeTotal (int nOrders)
```

User-defined functions

Recall: *returnType functionName* (*type1 param1, type2 param2, ...*)

Task: Compute total price

```
computeTotal (int nOrders)
```

User-defined functions

Recall: *returnType functionName* (*type1 param1, type2 param2, ...*)

Task: Compute total price

```
float computeTotal (int nOrders)
```

User-defined functions

Recall: *returnType functionName* (*type1 param1, type2 param2, ...*)

Task: Compute total price

```
float computeTotal (int nOrders)
{




}
```

User-defined functions

Recall: *returnType functionName* (*type1 param1*, *type2 param2*, *...*)

Task: Compute total price

```
float computeTotal (int nOrders)
{
    float fTotal;




}
```

User-defined functions

Recall: *returnType functionName* (*type1 param1, type2 param2, ...*)

Task: Compute total price

```
float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;

}
```

User-defined functions

Recall: *returnType functionName* (*type1 param1, type2 param2, ...*)

Task: Compute total price

```
float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}
```

User-defined functions

Recall: *returnType functionName* (*type1 param1*, *type2 param2*, *...*)

Task: Compute total price

```
float computeTotal (int nOrders)
{
    float fTotal;

    fTotal = nOrders * AMOUNT;

    return fTotal;
}
```

or you may also write this as...

```
float computeTotal (int nOrders)
{
    return nOrders * AMOUNT;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

int main ()
{
    int nDate;
    int nOrders;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    computeTotal ();
    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    computeTotal ();
    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    computeTotal (nOrders);
    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    computeTotal (nOrders);
    return 0;
}
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    computeTotal (nOrders);
    return 0;
}
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ?

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ?

nOrders = ?

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():
nDate = ?
nOrders = ?
fPrice = ?

Let’s trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():
nDate = ?
nOrders = ?
fPrice = ?

```
Date: _
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ~~?~~ 10102020

nOrders = ?

fPrice = ?

```
Date:  10102020
_
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ~~?~~ 10102020

nOrders = ?

fPrice = ?

```
Date:  10102020
Orders:  _
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ~~?~~ 10102020

nOrders = ~~?~~ 5

fPrice = ?

```
Date:  10102020
Orders:  5

_
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ~~?~~ 10102020

nOrders = ~~?~~ 5

fPrice = ?

```
Date:  10102020
Orders:  5
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():
nDate = ~~?~~ 10102020
nOrders = ~~?~~ 5
fPrice = ?

computeTotal():
nOrders =

```
Date:  10102020
Orders:  5
```

Let’s trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ~~?~~ 10102020

nOrders = ~~?~~ 5

fPrice = ?

computeTotal():

nOrders = 5

```
Date:  10102020
Orders:  5
```

Let's trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ~~?~~ 10102020

nOrders = ~~?~~ 5

fPrice = ?

computeTotal():

nOrders = 5

fTotal = ?

```
Date:  10102020
Orders:  5
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ~~?~~ 10102020

nOrders = ~~?~~ 5

fPrice = ?

computeTotal():

nOrders = 5

fTotal = ?

```
Date:  10102020
Orders:  5
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():
nDate = ~~?~~ 10102020
nOrders = ~~?~~ 5
fPrice = ?

computeTotal():
nOrders = 5
fTotal = ~~?~~ 495

```
Date:  10102020
Orders:  5
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():
nDate = ~~?~~ 10102020
nOrders = ~~?~~ 5
fPrice =

computeTotal():
nOrders = 5
fTotal = ~~?~~ 495

```
Date:  10102020
Orders:  5
```

Let’s trace

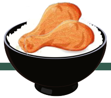
```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():
nDate = ~~?~~ 10102020
nOrders = ~~?~~ 5
fPrice = ~~?~~ 495

computeTotal():
nOrders = 5
fTotal = ~~?~~ 495

```
Date:  10102020
Orders:  5
```

Let’s trace



```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ~~?~~ 10102020

nOrders = ~~?~~ 5

fPrice = ~~?~~ 495

```
Date: 10102020
Orders: 5
Total price = 495.00
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

main():

nDate = ~~?~~ 10102020

nOrders = ~~?~~ 5

fPrice = ~~?~~ 495

```
Date: 10102020
Orders: 5
Total price = 495.00
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

Screen output:

```
Date:  10102020
Orders:  5
Total price = 495.00
```

Your turn to code

Write the function/s to compute for the gross amount and the tax.

$$total\_price = gross \times 1.12$$

$$tax = gross \times 0.12$$

Note that the `return` statement can return one value only.

Compute gross amount and tax

```
float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}
```

Compute gross amount and tax

```c
float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;

    fTotal = nOrders * AMOUNT;

    return fTotal;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    return 0;
}
```

Let’s trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;

    fTotal = nOrders * AMOUNT;

    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}
```

```c
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    return 0;
}
```

Let’s trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;

    fTotal = nOrders * AMOUNT;

    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}
```

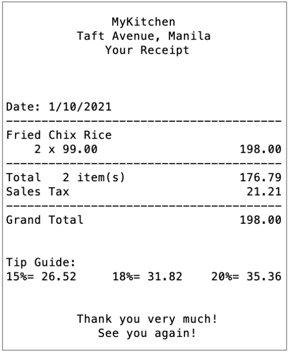
```c
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    printf ("Date:  ");
    scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    return 0;
}
```

Let’s code

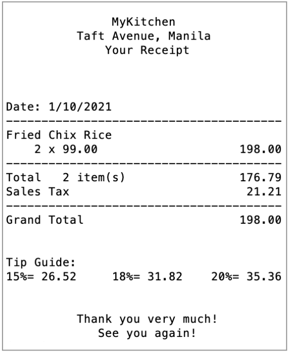


```
              MyKitchen
        Taft Avenue, Manila
            Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let’s code

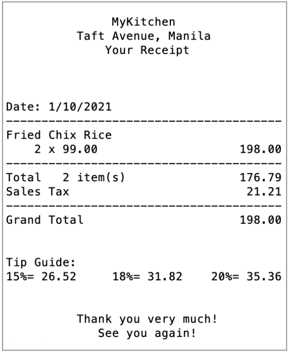


```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let's code



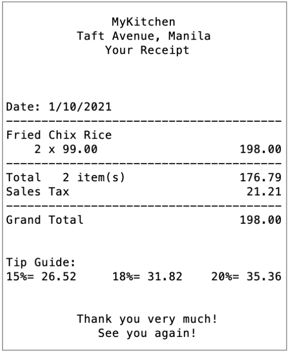
```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Recall: *returnType functionName* (*type1 param1, ...*)

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let's code



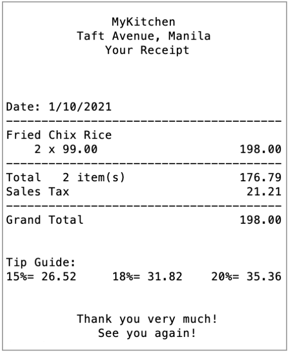
```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Recall: *returnType* *functionName* (*type1 param1*, ...)

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let’s code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

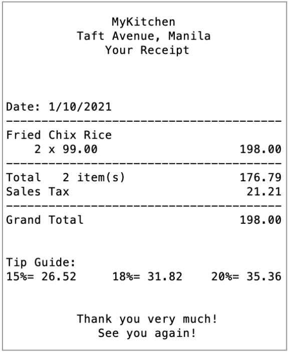
Recall: *returnType functionName* (*type1 param1, ...*)

```
displayDivider ()
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let’s code



```
                 MyKitchen
           Taft Avenue, Manila
               Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

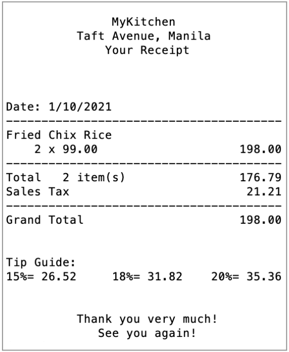
Recall: *returnType functionName* (*type1 param1, ...*)

```
displayDivider ()
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let’s code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

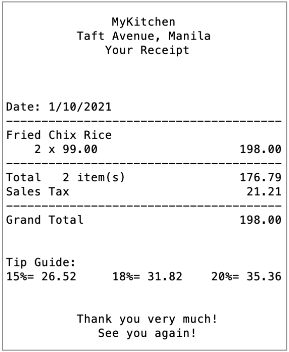
Recall: *returnType functionName* (*type1 param1*, ...)

`displayDivider ()`

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let's code



```
            MyKitchen
       Taft Avenue, Manila
           Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

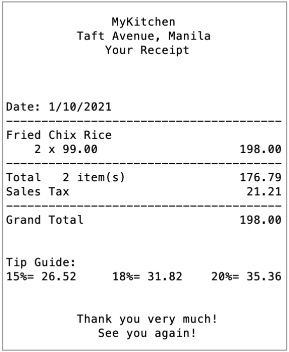
Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Recall: *returnType functionName* (*type1 param1, ...*)

```
void displayDivider ()
```

Let’s code



```
              MyKitchen
        Taft Avenue, Manila
            Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

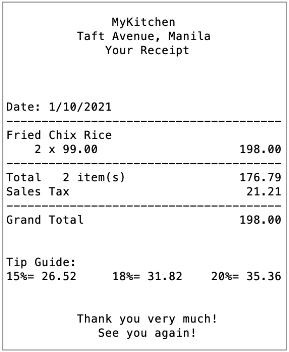
Recall: *returnType functionName* (*type1 param1*, ...)

```
void displayDivider ()
{



}
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let’s code



```
              MyKitchen
        Taft Avenue, Manila
            Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52    18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Recall: *returnType functionName* (*type1 param1, ...*)

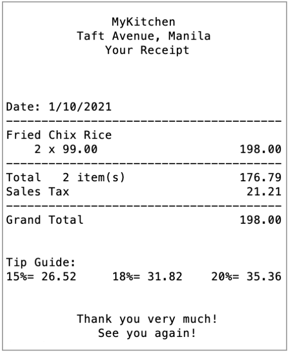
```
void displayDivider ()
{



}
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
    - header
    - details
    - divider
    - tip guide
    - footer

No. of characters per line: 40

Let’s code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Recall: *returnType functionName* (*type1 param1*, ...)

```
void displayDivider ()
{
    printf ("%s%s%s%s\n", "----------",
                 "----------", "----------",
                 "----------");
}
```

No. of characters per line: 40

Let’s trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}
```

```c
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;

    fTotal = nOrders * AMOUNT;

    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void displayDivider ()
{
    printf ("%s%s%s%s\n", "----------",
            "----------", "----------",
            "----------");
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    return 0;
}
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void displayDivider ()
{
    printf ("%s%s%s%s\n", "----------",
            "----------", "----------",
            "----------");
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);
    /*
    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);
    */
    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;

    fTotal = nOrders * AMOUNT;

    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void displayDivider ()
{
    printf ("%s%s%s%s\n", "----------",
            "----------", "----------",
            "----------");
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);
    /*
    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);
    */
    return 0;
}
```

Let’s trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void displayDivider ()
{
    printf ("%s%s%s%s\n", "----------",
            "----------", "----------",
            "----------");
}
```

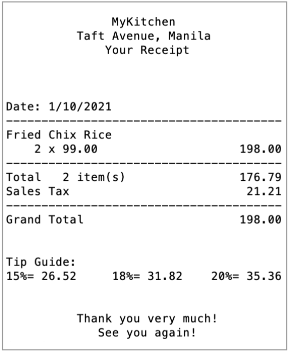
```c
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);
    /*
    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);
    */

    displayDivider ();

    return 0;
}
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void displayDivider ()
{
    printf ("%s%s%s%s\n", "----------",
            "----------", "----------",
            "----------");
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);
    /*
    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);
    */

    displayDivider ();

    return 0;
}
```

Let's code

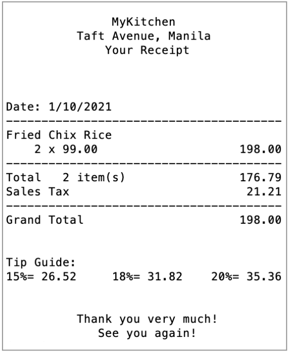


```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
    - header
    - details
    - divider
    - tip guide
    - footer

Let’s code

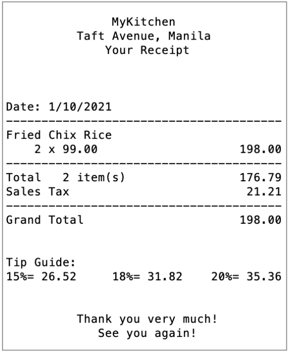


```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let's code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Recall: What is the screen output of

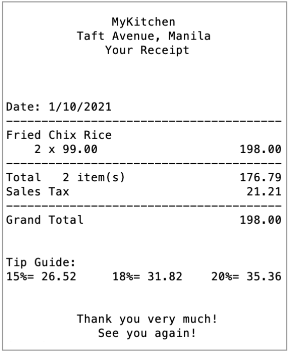
```
printf ("%10s", "hello");
```

[redacted]

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

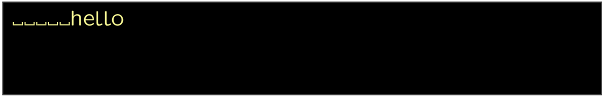
Let’s code



```
              MyKitchen
         Taft Avenue, Manila
            Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


         Thank you very much!
            See you again!
```

Recall: What is the screen output of

```
printf ("%10s", "hello");
```

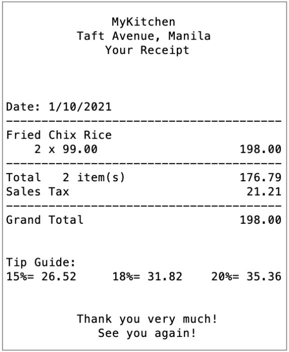


```
␣␣␣␣␣hello
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

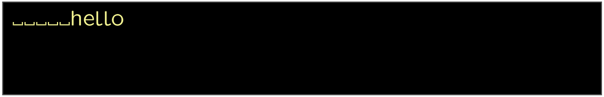
Let’s code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Recall: What is the screen output of

```
printf ("%10s", "hello");
```



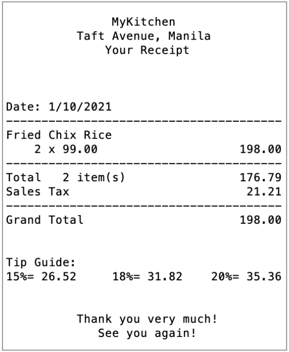
```
␣␣␣␣␣hello
```

Note that each line has 40 characters (or spaces).

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let’s code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Recall: What is the screen output of

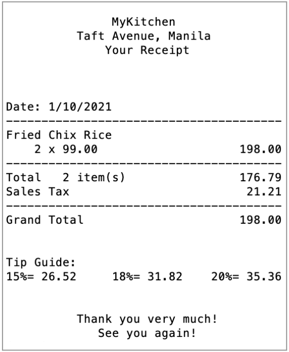
```
printf ("%10s", "hello");
```

```
␣␣␣␣␣hello
```

Note that each line has 40 characters (or spaces).

```
void displayHeader ()
{


}
```

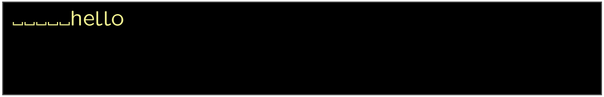
Let's code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Recall: What is the screen output of

```
printf ("%10s", "hello");
```



```
␣␣␣␣␣hello
```

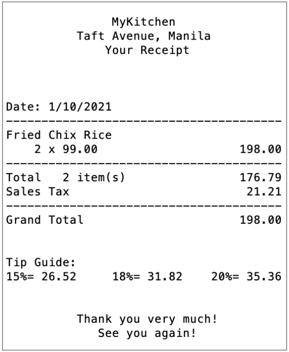
Note that each line has 40 characters (or spaces).

```
void displayHeader ()
{
    printf ("%s\n", "MyKitchen");
    printf ("%s\n", "Taft Avenue, Manila");
    printf ("%s\n", "Your Receipt");
}
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Let’s code



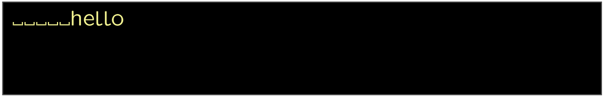
```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
    - header
    - details
    - divider
    - tip guide
    - footer

Recall: What is the screen output of

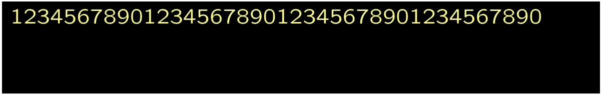
```
printf ("%10s", "hello");
```



```
␣␣␣␣␣hello
```

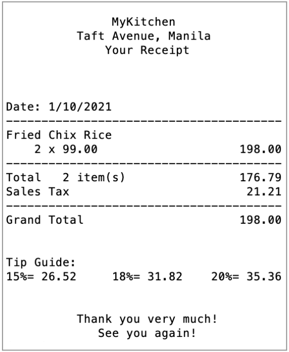
Note that each line has 40 characters (or spaces).

```
void displayHeader ()
{
    printf ("%s\n", "MyKitchen");
    printf ("%s\n", "Taft Avenue, Manila");
    printf ("%s\n", "Your Receipt");
}
```



```
1234567890123456789012345678901234567890
```

Let’s code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
    - header
    - details
    - divider
    - tip guide
    - footer

Recall: What is the screen output of

```
printf ("%10s", "hello");
```

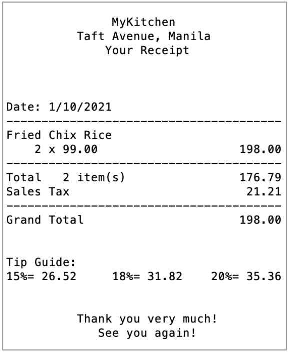
```
␣␣␣␣␣hello
```

Note that each line has 40 characters (or spaces).

```
void displayHeader ()
{
    printf ("%s\n", "MyKitchen");
    printf ("%s\n", "Taft Avenue, Manila");
    printf ("%s\n", "Your Receipt");
}
```

```
1234567890123456789012345678901234567890
```

Let’s code



```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Recall: What is the screen output of

```
printf ("%10s", "hello");
```

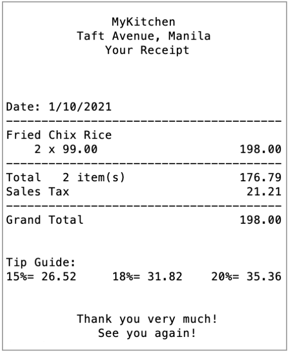
```
␣␣␣␣␣hello
```

Note that each line has 40 characters (or spaces).

```
void displayHeader ()
{
    printf ("%s\n", "MyKitchen");
    printf ("%s\n", "Taft Avenue, Manila");
    printf ("%s\n", "Your Receipt");
}
```

```
1234567890123456789012345678901234567890
```

Let’s code



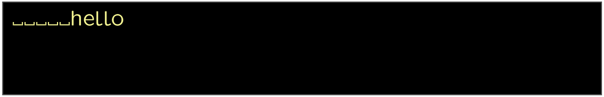
```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt


Date: 1/10/2021
----------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
----------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
----------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82     20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Recall: What is the screen output of

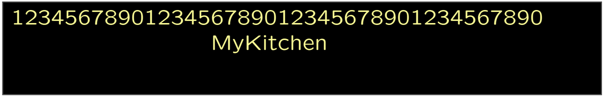
```
printf ("%10s", "hello");
```



```
␣␣␣␣␣hello
```

Note that each line has 40 characters (or spaces).

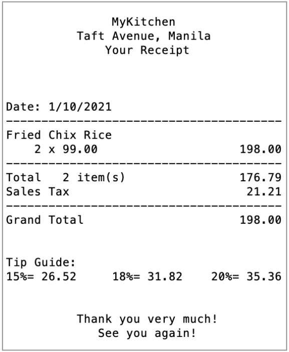
```
void displayHeader ()
{
    printf ("%s\n", "MyKitchen");
    printf ("%s\n", "Taft Avenue, Manila");
    printf ("%s\n", "Your Receipt");
}
```



```
1234567890123456789012345678901234567890
               MyKitchen
```

- half of 40 is 20
- MyKitchen has 9 characters
- $9/2 = 4$; 5^{th} character should be at 20^{th} space

Let’s code



```
              MyKitchen
         Taft Avenue, Manila
            Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                    198.00
---------------------------------------
Total   2 item(s)                176.79
Sales Tax                         21.21
---------------------------------------
Grand Total                      198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Recall: What is the screen output of

```
printf ("%10s", "hello");
```

```
␣␣␣␣␣hello
```

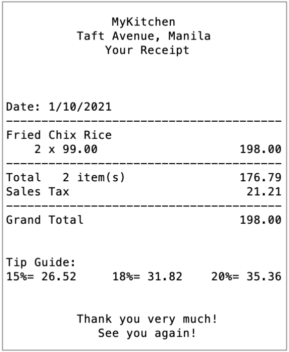
Note that each line has 40 characters (or spaces).

```
void displayHeader ()
{
    printf ("%s\n", "MyKitchen");
    printf ("%s\n", "Taft Avenue, Manila");
    printf ("%s\n", "Your Receipt");
}
```

```
1234567890123456789012345678901234567890
                   MyKitchen
```

- half of 40 is 20
- MyKitchen has 9 characters
- $9/2 = 4$; 5^{th} character should be at 20^{th} space
- last character of MyKitchen is at 24^{th} space

Let’s code



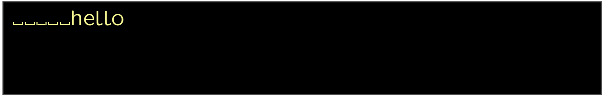
```
               MyKitchen
          Taft Avenue, Manila
              Your Receipt



Date: 1/10/2021
---------------------------------------
Fried Chix Rice
    2 x 99.00                     198.00
---------------------------------------
Total   2 item(s)                 176.79
Sales Tax                          21.21
---------------------------------------
Grand Total                       198.00


Tip Guide:
15%= 26.52     18%= 31.82    20%= 35.36


          Thank you very much!
             See you again!
```

Tasks:

1. Ask for inputs
2. Compute
3. Display receipt
   - header
   - details
   - divider
   - tip guide
   - footer

Recall: What is the screen output of

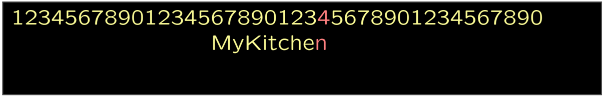
```
printf ("%10s", "hello");
```



```
␣␣␣␣␣hello
```

Note that each line has 40 characters (or spaces).

```
void displayHeader ()
{
    printf ("%24s\n", "MyKitchen");
    printf ("%s\n", "Taft Avenue, Manila");
    printf ("%s\n", "Your Receipt");
}
```



```
1234567890123456789012345678901234567890
               MyKitchen
```

- half of 40 is 20
- MyKitchen has 9 characters
- $9/2 = 4$; 5^{th} character should be at 20^{th} space
- last character of MyKitchen is at 24^{th} space

HW: Your tasks

Write the function/s to

- display details
- display footer

Complete `main()` to generate the receipt

Your turn to code

Write the function/s to compute for tip amounts.

$$tip_1 = gross * 0.15$$

$$tip_2 = gross * 0.18$$

$$tip_3 = gross * 0.20$$

Note that the `return` statement can return one value only. You may however make **3 functions** to compute for each of the tip amounts.

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
void computeTips (float fGross, float* tip1, float* tip2, float* tip3)
{
}
```

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
void computeTips (float fGross, float* tip1, float* tip2, float* tip3)
{
}
```

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
void computeTips (float fGross, float* tip1, float* tip2, float* tip3)
{
}
```

The data type of `tip1` is `float*`.

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
void computeTips (float fGross, float* tip1, float* tip2, float* tip3)
{
}
```

The data type of `tip1` is `float*`.

- `tip1` is a *pointer variable*.

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
void computeTips (float fGross, float* tip1, float* tip2, float* tip3)
{
}
```

The data type of `tip1` is `float*`.

- `tip1` is a *pointer variable*.

Pointer variable

- *pointer* in short
- holds a *memory address*, where the value is stored.

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
void computeTips (float fGross, float* tip1, float* tip2, float* tip3)
{
}
```

The data type of `tip1` is `float*`.

- `tip1` is a *pointer variable*.

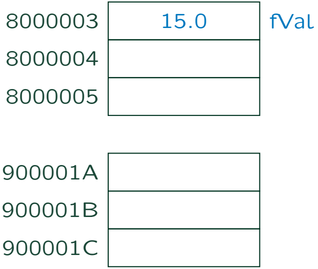
Pointer variable

- *pointer* in short
- holds a *memory address*, where the value is stored.

- The data type of `tip` is `float*`, *pointer to a float*.

Pointers

```
float fVal;
    ⋮
fVal = 15;
```



8000003 15.0 fVal

8000004

8000005

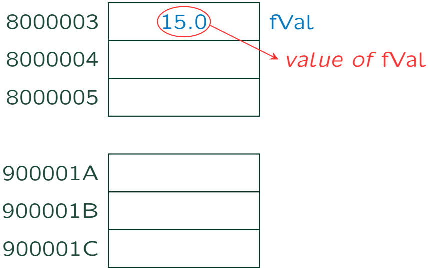
900001A

900001B

900001C

Pointers

```
float fVal;
    ⋮
fVal = 15;
```



8000003 | 15.0 | fVal

value of fVal

8000004

8000005

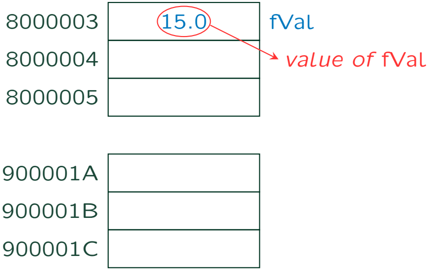
900001A

900001B

900001C

Pointers

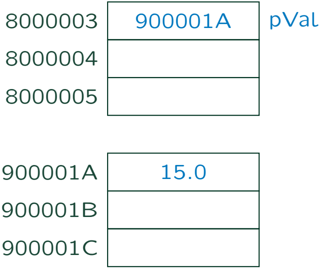
```
float fVal;
    ⋮
fVal = 15;
```



8000003	15.0	fVal
8000004		
8000005		
900001A		
900001B		
900001C		

value of fVal

```
float* pVal
    ⋮
*pVal = 15;
```

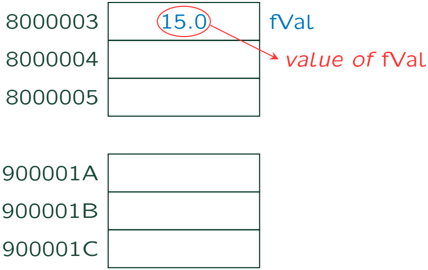


8000003	900001A	pVal
8000004		
8000005		
900001A	15.0	
900001B		
900001C		

DO NOT declare pointer variables for now. You MAY use pointers as parameters.

Pointers

```
float fVal;
    ⋮
fVal = 15;
```



8000003 15.0 fVal

8000004

8000005

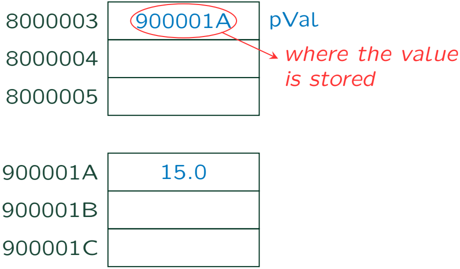
value of fVal

900001A

900001B

900001C

```
float* pVal
    ⋮
*pVal = 15;
```



8000003 900001A pVal

8000004

8000005

where the value is stored

900001A 15.0

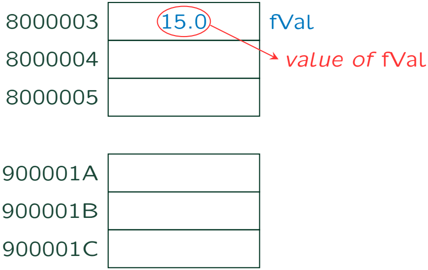
900001B

900001C

DO NOT declare pointer variables for now. You MAY use pointers as parameters.

Pointers

```
float fVal;
    ⋮
fVal = 15;
```



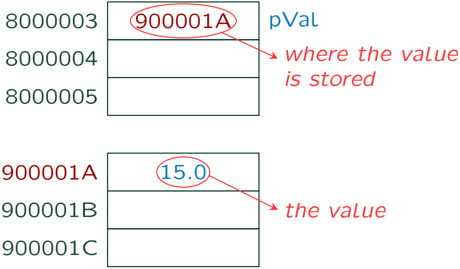
8000003	15.0
8000004	
8000005	

900001A	
900001B	
900001C	

fVal

value of fVal

```
float* pVal
    ⋮
*pVal = 15;
```



8000003	900001A
8000004	
8000005	

900001A	15.0
900001B	
900001C	

pVal

where the value is stored

the value

DO NOT declare pointer variables for now. You MAY use pointers as parameters.

Pointers

```
float* pVal
    ⋮
*pVal = 15;
```

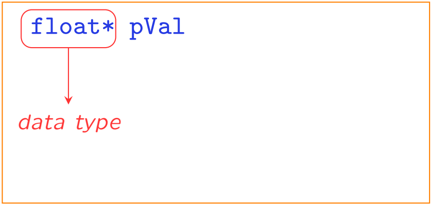
Pointers

```
float* pVal
    ⋮
*pVal = 15;
```

```
float* pVal
```

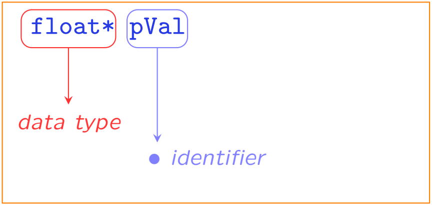
Pointers

```
float* pVal
    ⋮
*pVal = 15;
```



Pointers

```
float* pVal
    ⋮
*pVal = 15;
```



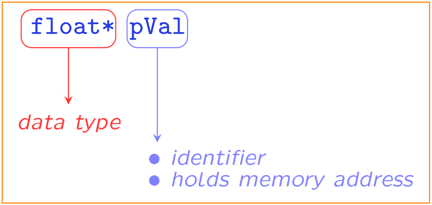
float* pVal

data type

● identifier

Pointers

```
float* pVal
    ⋮
*pVal = 15;
```



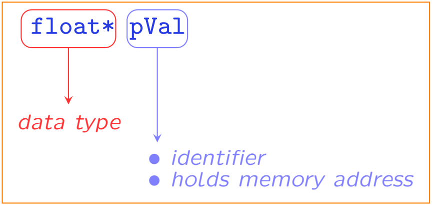
float* pVal

data type

- *identifier*
- *holds memory address*

Pointers

```
float* pValue
   ⋮
*pVal = 15;
```



float* pVal

data type

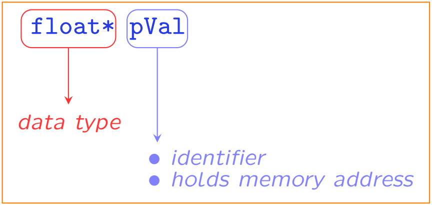
- *identifier*
- *holds memory address*

```
float *pVal
   ⋮
*pVal = 15;
```

Pointers

```
float* pVal
    ⋮
*pVal = 15;
```

```
float *pVal
    ⋮
*pVal = 15;
```



float* pVal

data type

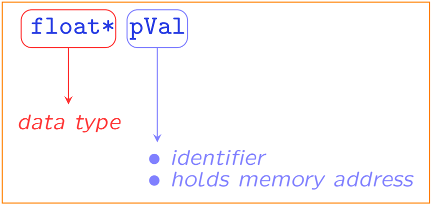
- *identifier*
- *holds memory address*

```
float *pVal
```

Pointers

```
float* pVal
    ⋮
*pVal = 15;
```

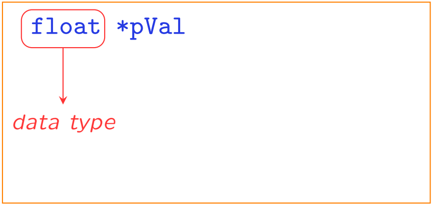
```
float *pVal
    ⋮
*pVal = 15;
```



float* pVal

data type

- *identifier*
- *holds memory address*

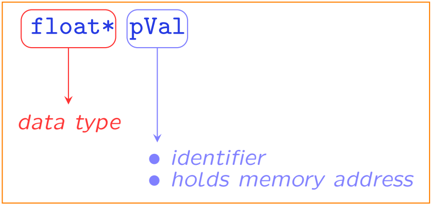


float *pVal

data type

Pointers

```
float* pVal
    ⋮
*pVal = 15;
```

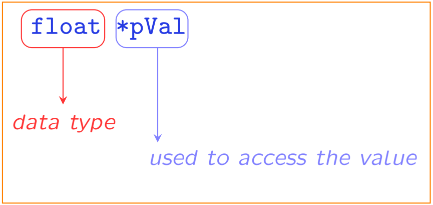


float* pVal

data type

- *identifier*
- *holds memory address*

```
float *pVal
    ⋮
*pVal = 15;
```



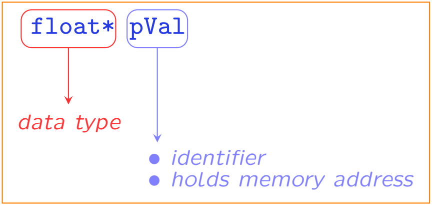
float *pVal

data type

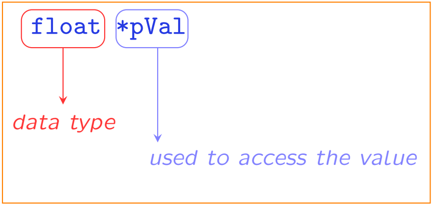
used to access the value

Pointers

```
float* pVal
    ⋮
*pVal = 15;
```



```
float *pVal
    ⋮
*pVal = 15;
```



The * may either be close to the data type (at its left), or close to the identifier (at its right).

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20


void computeTips (float fGross, float* tip1, float* tip2, float* tip3)
{




}
```

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20


void computeTips (float fGross, float * tip1 , float* tip2, float* tip3)
{
    *tip1 = fGross * TIP1;



}
```

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20


void computeTips (float fGross, float* tip1, float * tip2 , float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;

}
```

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20


void computeTips (float fGross, float* tip1, float* tip2, float * tip3 )
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

Another solution

You may also create **one function** that will compute and update the 3 tip amounts.

```
#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20


void computeTips (float fGross, float* tip1, float* tip2, float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                float* tip1, float* tip2,
                float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                float* tip1, float* tip2,
                float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips ( );
    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross,  );
    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                float* tip1, float* tip2,
                float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1,  );
    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                float* tip1, float* tip2,
                float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                &fTip3);
    return 0;
}
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

Let’s trace

```c
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

8881	
8882	
8883	
8884	
8885	
8886	
8887	
8888	
8889	

9020	
9021	
9022	
9023	
9024	

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

8881	?	nDate
8882		
8883		
8884		
8885		
8886		
8887		
8888		
8889		

9020	
9021	
9022	
9023	
9024	

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

8881	?	nDate
8882	?	nOrders
8883		
8884		
8885		
8886		
8887		
8888		
8889		

9020	
9021	
9022	
9023	
9024	

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

8881	?	nDate
8882	?	nOrders
8883	?	fPrice
8884		
8885		
8886		
8887		
8888		
8889		

9020	
9021	
9022	
9023	
9024	

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

8881	?	nDate
8882	?	nOrders
8883	?	fPrice
8884	?	fGross
8885		
8886		
8887		
8888		
8889		

9020	
9021	
9022	
9023	
9024	

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

8881	?	nDate
8882	?	nOrders
8883	?	fPrice
8884	?	fGross
8885	?	fTax
8886		
8887		
8888		
8889		

9020	
9021	
9022	
9023	
9024	

Let's trace

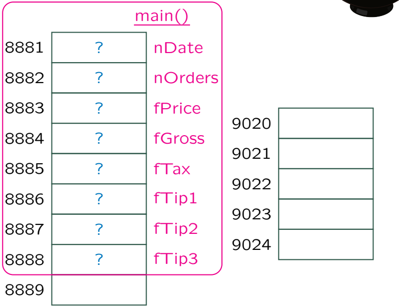
```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

8881	?	nDate
8882	?	nOrders
8883	?	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

9020	
9021	
9022	
9023	
9024	

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```



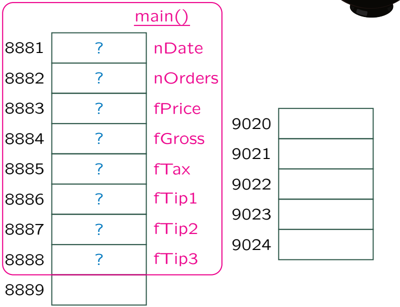
main()

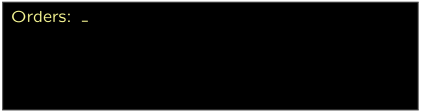
8881	?	nDate
8882	?	nOrders
8883	?	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

9020	
9021	
9022	
9023	
9024	

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```





```
Orders: _
```

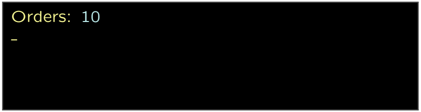
Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	?	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

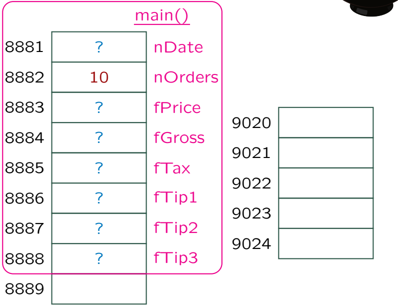
9020	
9021	
9022	
9023	
9024	



```
Orders:  10
_
```

Let’s trace

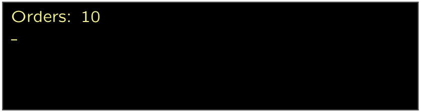
```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```



main()

8881	?	nDate
8882	10	nOrders
8883	?	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

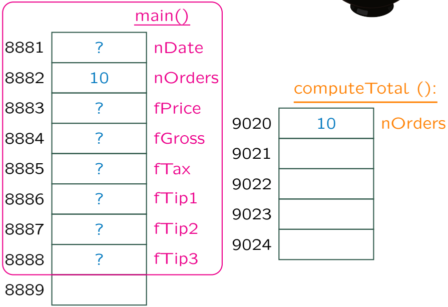
9020	
9021	
9022	
9023	
9024	



```
Orders:  10
_
```

Let's trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

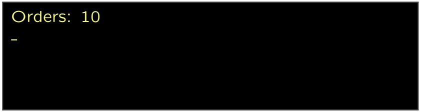


main()

8881	?	nDate
8882	10	nOrders
8883	?	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

computeTotal ():

9020	10	nOrders
9021		
9022		
9023		
9024		



```
Orders: 10
_
```

Let’s trace

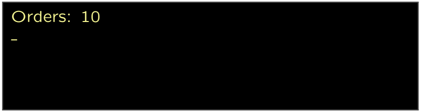
```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	?	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

computeTotal ():

9020	10	nOrders
9021	?	fTotal
9022		
9023		
9024		



```
Orders: 10
_
```

Let’s trace

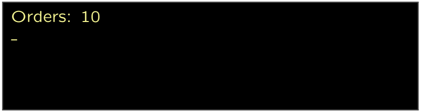
```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	?	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

computeTotal ():

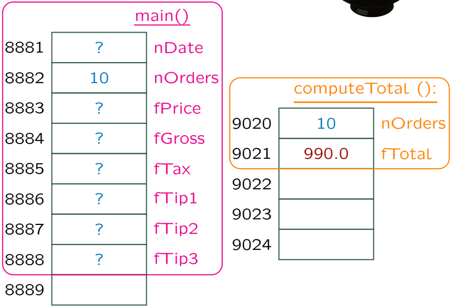
9020	10	nOrders
9021	990.0	fTotal
9022		
9023		
9024		



```
Orders: 10
_
```

Let’s trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```



main()

8881	?	nDate
8882	10	nOrders
8883	?	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

computeTotal ():

9020	10	nOrders
9021	990.0	fTotal
9022		
9023		
9024		

```
Orders: 10
_
```

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

9020	
9021	
9022	
9023	
9024	

```
Orders:  10
_
```

Let's trace

```c
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

9020	
9021	
9022	
9023	
9024	

```
Orders:  10
Total price = 990.00
_
```

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

9020	
9021	
9022	
9023	
9024	

```
Orders:  10
Total price = 990.00
_
```

Let's trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

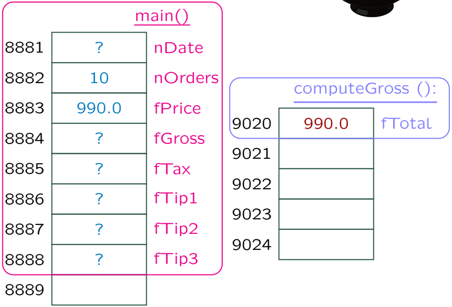
computeGross ():

9020	990.0	fTotal
9021		
9022		
9023		
9024		

```
Orders: 10
Total price = 990.00
_
```

Let’s trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                float* tip1, float* tip2,
                float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

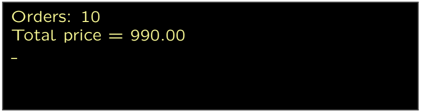


main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

computeGross ():

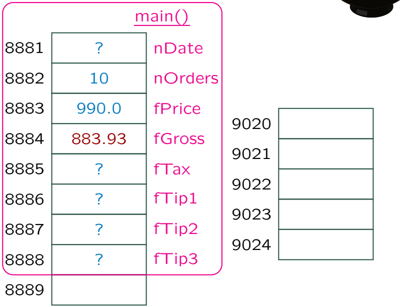
9020	990.0	fTotal
9021		
9022		
9023		
9024		



```
Orders: 10
Total price = 990.00
_
```

Let’s trace

```c
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```



main()

8881 ? nDate

8882 10 nOrders

8883 990.0 fPrice

8884 883.93 fGross

8885 ? fTax

8886 ? fTip1

8887 ? fTip2

8888 ? fTip3

8889

9020

9021

9022

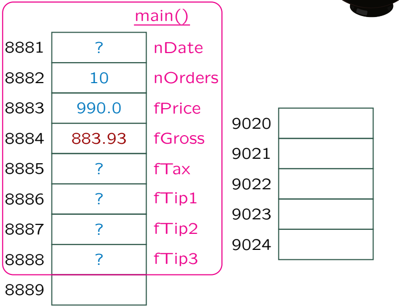
9023

9024

```
Orders:  10
Total price = 990.00
_
```

Let’s trace

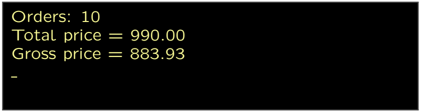
```c
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```



main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

9020	
9021	
9022	
9023	
9024	



```
Orders:  10
Total price = 990.00
Gross price = 883.93
_
```

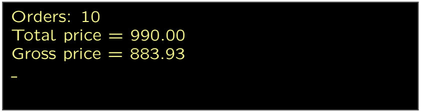
Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

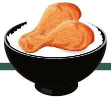
8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

9020	
9021	
9022	
9023	
9024	

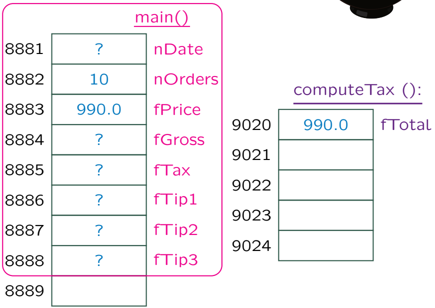


```
Orders:  10
Total price = 990.00
Gross price = 883.93

_
```

Let’s trace



```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```



main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

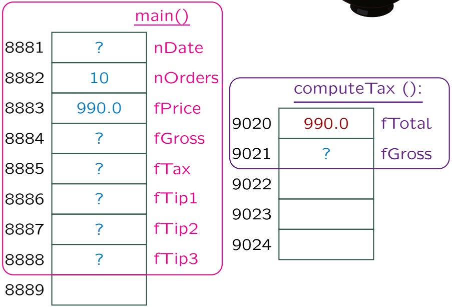
computeTax ():

9020	990.0	fTotal
9021		
9022		
9023		
9024		

```
Orders: 10
Total price = 990.00
Gross price = 883.93

_
```

Let’s trace

```c
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

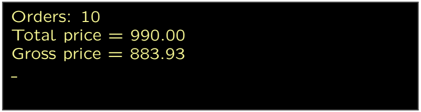


main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

computeTax ():

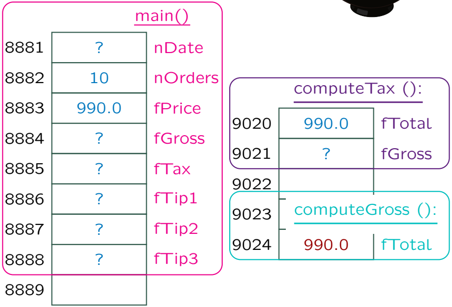
9020	990.0	fTotal
9021	?	fGross
9022		
9023		
9024		



```
Orders: 10
Total price = 990.00
Gross price = 883.93

_
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```



main()

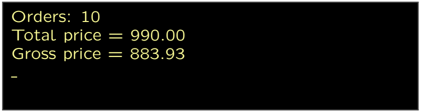
8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

computeTax ():

9020	990.0	fTotal
9021	?	fGross
9022		

computeGross ():

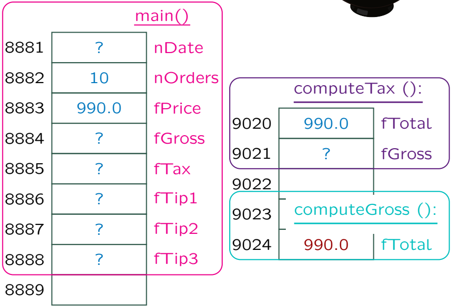
9023		
9024	990.0	fTotal



```
Orders: 10
Total price = 990.00
Gross price = 883.93
_
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                float* tip1, float* tip2,
                float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```



main()

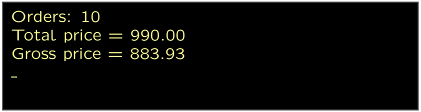
8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

computeTax ():

9020	990.0	fTotal
9021	?	fGross
9022		

computeGross ():

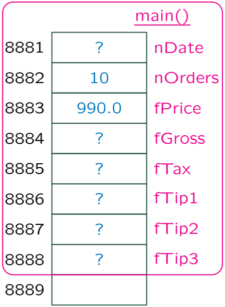
9023		
9024	990.0	fTotal



```
Orders: 10
Total price = 990.00
Gross price = 883.93

_
```

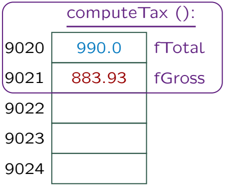
Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                float* tip1, float* tip2,
                float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```



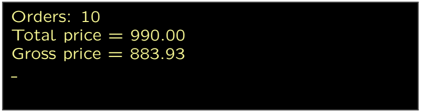
main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		



computeTax ():

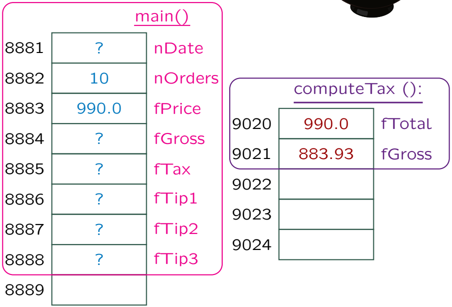
9020	990.0	fTotal
9021	883.93	fGross
9022		
9023		
9024		



```
Orders:  10
Total price = 990.00
Gross price = 883.93

_
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

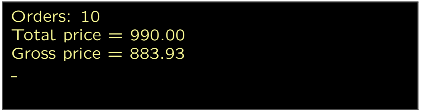


main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	?	fGross
8885	?	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3
8889		

computeTax ():

9020	990.0	fTotal
9021	883.93	fGross
9022		
9023		
9024		



```
Orders: 10
Total price = 990.00
Gross price = 883.93

_
```

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

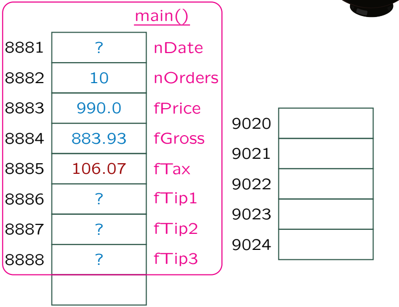
8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3

9020	
9021	
9022	
9023	
9024	

```
Orders: 10
Total price = 990.00
Gross price = 883.93
_
```

Let's trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```



main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3

9020	
9021	
9022	
9023	
9024	

```
Orders: 10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07
_
```

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

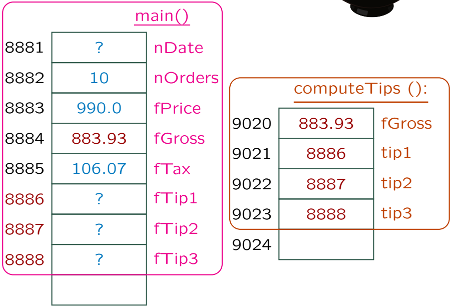
8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3

9020	
9021	
9022	
9023	
9024	

```
Orders:  10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07

_
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```



main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	?	fTip1
8887	?	fTip2
8888	?	fTip3

computeTips ():

9020	883.93	fGross
9021	8886	tip1
9022	8887	tip2
9023	8888	tip3
9024		

```
Orders: 10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07
_
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	132.59	fTip1
8887	?	fTip2
8888	?	fTip3

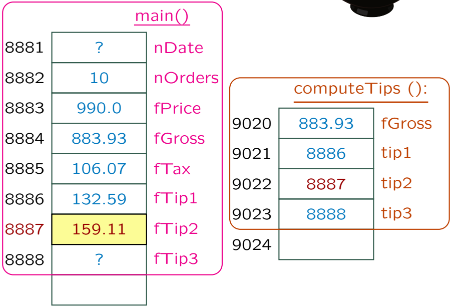
computeTips ():

9020	883.93	fGross
9021	8886	tip1
9022	8887	tip2
9023	8888	tip3
9024		

```
Orders: 10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07
_
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```



main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	132.59	fTip1
8887	159.11	fTip2
8888	?	fTip3

computeTips ():

9020	883.93	fGross
9021	8886	tip1
9022	8887	tip2
9023	8888	tip3
9024		

```
Orders: 10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07
_
```

Let’s trace

```
#include<stdio.h>

#define AMOUNT 99.00
#define TAX 0.12
#define TIP1 0.15
#define TIP2 0.18
#define TIP3 0.20

float computeTotal (int nOrders)
{
    float fTotal;
    fTotal = nOrders * AMOUNT;
    return fTotal;
}

float computeGross (float fTotal)
{
    return fTotal / (1 + TAX);
}

float computeTax (float fTotal)
{
    float fGross = computeGross (fTotal);
    return fTotal - fGross;
}

void computeTips (float fGross,
                  float* tip1, float* tip2,
                  float* tip3)
{
    *tip1 = fGross * TIP1;
    *tip2 = fGross * TIP2;
    *tip3 = fGross * TIP3;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	132.59	fTip1
8887	159.11	fTip2
8888	179.79	fTip3

computeTips ():

9020	883.93	fGross
9021	8886	tip1
9022	8887	tip2
9023	8888	tip3
9024		

```
Orders:  10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07
_
```

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	132.59	fTip1
8887	159.11	fTip2
8888	179.79	fTip3
8889		

9020	
9021	
9022	
9023	
9024	

```
Orders:  10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07
```

Let’s trace

```c
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	132.59	fTip1
8887	159.11	fTip2
8888	179.79	fTip3
8889		

9020	
9021	
9022	
9023	
9024	

```
Orders:  10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07
Tips:  132.59, 159.11, 179.79
```

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                 &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```

main()

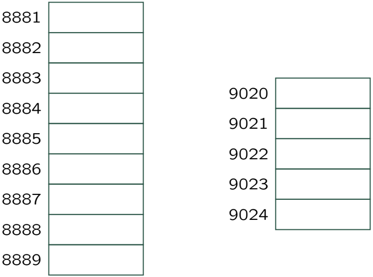
8881	?	nDate
8882	10	nOrders
8883	990.0	fPrice
8884	883.93	fGross
8885	106.07	fTax
8886	132.59	fTip1
8887	159.11	fTip2
8888	179.79	fTip3
8889		

9020	
9021	
9022	
9023	
9024	

```
Orders:  10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07
Tips:  132.59, 159.11, 179.79
```

Let’s trace

```
int main ()
{
    int nDate;
    int nOrders;
    float fPrice;
    float fGross;
    float fTax;
    float fTip1, fTip2, fTip3;

    // printf ("Date:  ");
    // scanf ("%d", &nDate);

    printf ("Orders:  ");
    scanf ("%d", &nOrders);

    fPrice = computeTotal (nOrders);
    printf ("Total price = %.2f\n", fPrice);

    fGross = computeGross (fPrice);
    printf ("Gross price = %.2f\n", fGross);

    fTax = computeTax (fPrice);
    printf ("Sales tax = %.2f\n", fTax);

    computeTips (fGross, &fTip1, &fTip2,
                &fTip3);
    printf ("Tips:  %.2f, %.2f, %.2f\n",
            fTip1, fTip2, fTip3);

    return 0;
}
```



Screen Output:

```
Orders:  10
Total price = 990.00
Gross price = 883.93
Sales tax = 106.07
Tips:  132.59, 159.11, 179.79
```

User-defined functions

returnType functionName (*type1 param1*, *type2 param2*, ...)

User-defined functions

returnType functionName (*type1 param1, type2 param2, ...*)

- `void displayDividers ()`

User-defined functions

returnType functionName (*type1 param1*, *type2 param2*, ...)

- `void displayDividers ()`
- `float computeTax (float fTotal)`

User-defined functions

returnType functionName (*type1 param1*, *type2 param2*, *...*)

- `void displayDividers ()`
- `float computeTax (float fTotal)`
- `void displayDetails (int nDate, int nOrders)`

User-defined functions

returnType functionName (*type1 param1*, *type2 param2*, *...*)

- `void displayDividers ()`
- `float computeTax (float fTotal)`
- `void displayDetails (int nDate, int nOrders)`

-
  ```
  void computeTips (float fGross, float* tip1,
                    float* tip2, float* tip3)
  ```

☺ Thank you! ☺