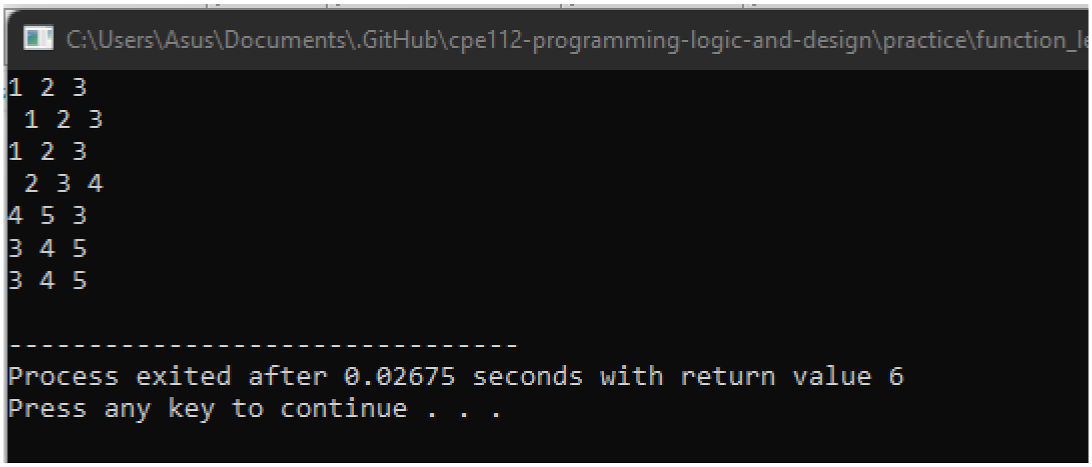
Timajo, Kurt Vincent O.
CpE 1b

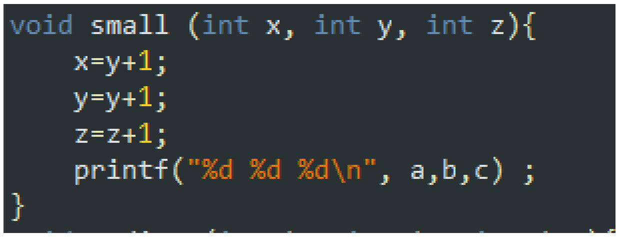
Assignment 9

Exercise #1


C:\Users\Asus\Documents\.GitHub\cpe112-programming-logic-and-design\practice\function_l

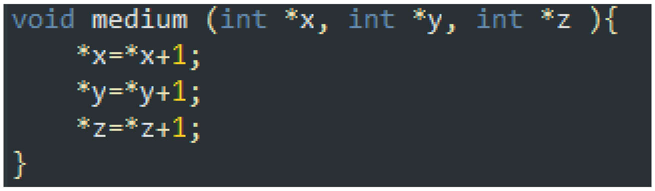
```
1 2 3
 1 2 3
1 2 3
 2 3 4
4 5 3
3 4 5
3 4 5

--------------------------------
Process exited after 0.02675 seconds with return value 6
Press any key to continue . . .
```

Explanations



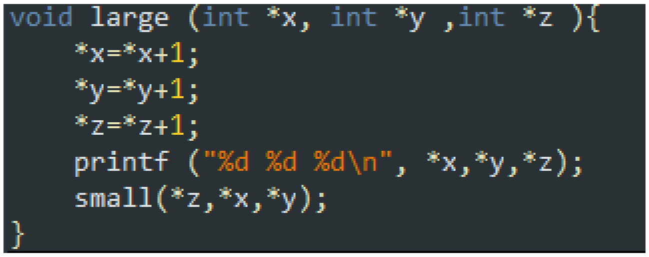
```
void small (int x, int y, int z){
    x=y+1;
    y=y+1;
    z=z+1;
    printf("%d %d %d\n", a,b,c) ;
}
```

- The small function adds 1 to each integer value of x, y, and z passed to it. On the last line, inside the small function is a print function with parameters, a, b, and c. It does not print the values of x, y, and z but it only prints the values of a, b, and c which are accessible globally.



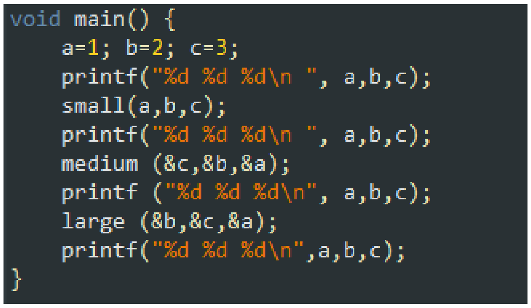
```
void medium (int *x, int *y, int *z ){
    *x=*x+1;
    *y=*y+1;
    *z=*z+1;
}
```

- The medium function accepts 3 parameters *x, *y, and *z which uses the pointer operator. This function executes the adding of 1 to x, y, and z which can affect the actual value of the variables passed. Changes that took effect on formal parameters will take effect also on the actual parameters a, b, and c.



```
void large (int *x, int *y ,int *z ){
    *x=*x+1;
    *y=*y+1;
    *z=*z+1;
    printf ("%d %d %d\n", *x,*y,*z);
    small(*z,*x,*y);
}
```

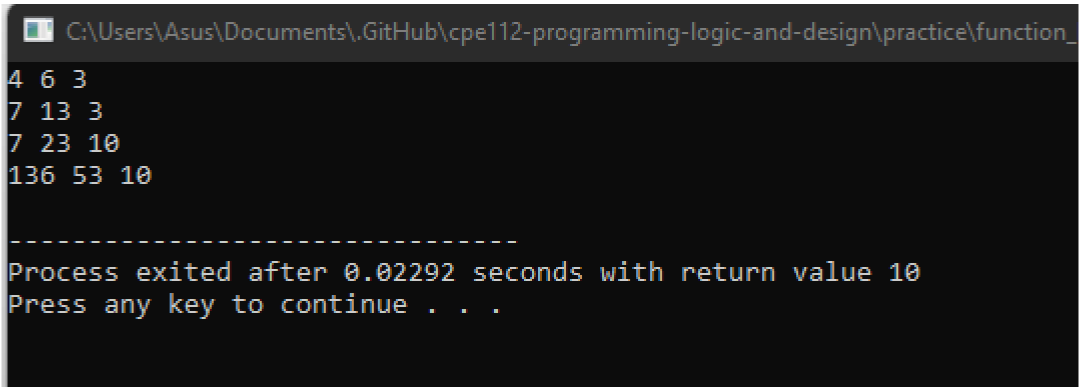
- Large function does the same thing as medium in which it modifies the actual value of the variable integers passed to it. But unlike medium function, it prints out the new values of the passed variable once it is called. After printing x, y, and z, it calls the small function by passing the values of z, x, and y.



```
void main() {
    a=1; b=2; c=3;
    printf("%d %d %d\n ", a,b,c);
    small(a,b,c);
    printf("%d %d %d\n ", a,b,c);
    medium (&c,&b,&a);
    printf ("%d %d %d\n", a,b,c);
    large (&b,&c,&a);
    printf("%d %d %d\n",a,b,c);
}
```

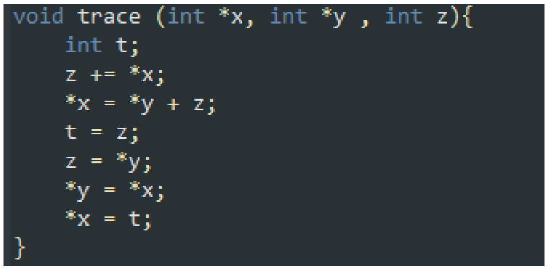
- In the main function, the values of global variables a, b, and c are declared inside. Then these are passed to the printf function for displaying its value. Next, the small function is called through call by value method passing parameters a, b, and c. The medium and large function are then called by reference method passing the memory addresses of a, b, and c in which as a result, the new values will appear in the output when printed.

Exercise #2



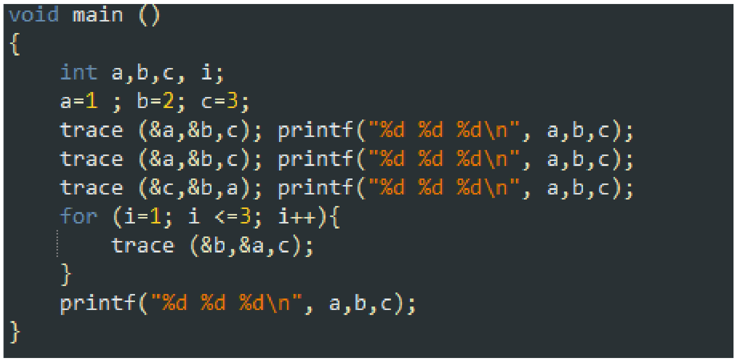
```
C:\Users\Asus\Documents\.GitHub\cpe112-programming-logic-and-design\practice\function_
4 6 3
7 13 3
7 23 10
136 53 10

--------------------------------
Process exited after 0.02292 seconds with return value 10
Press any key to continue . . .
```

Explanations



```c
void trace (int *x, int *y , int z){
    int t;
    z += *x;
    *x = *y + z;
    t = z;
    z = *y;
    *y = *x;
    *x = t;
}
```

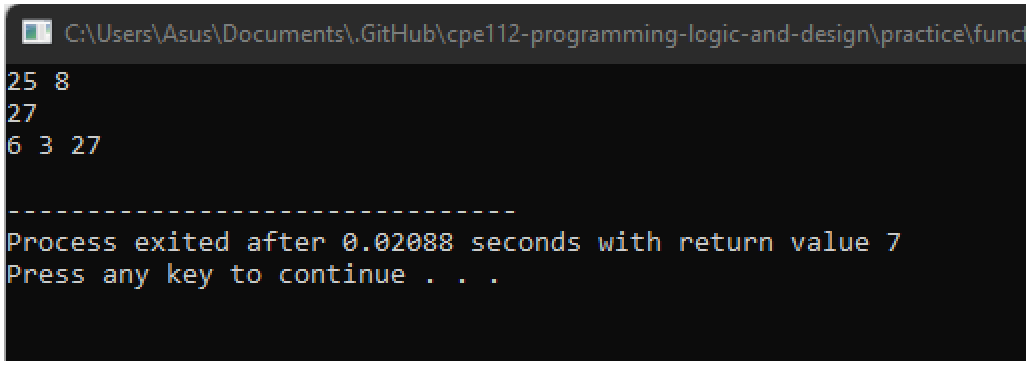
- The trace function accepts three parameters *x, *y, and *z. This function modifies the actual values of the first two variables passed to it. For the value of x, it is set equal to the value of the sum of y and z first then set again to the value of t. For y, it is set equal to the value of x which was the result of adding y and z. Inside the function, z has a new value equal to y, but it does not modify the actual values of the variable declared in the main function.



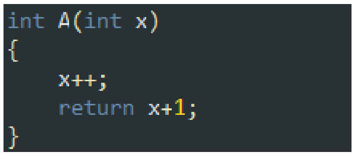
```c
void main ()
{
    int a,b,c, i;
    a=1 ; b=2; c=3;
    trace (&a,&b,c); printf("%d %d %d\n", a,b,c);
    trace (&a,&b,c); printf("%d %d %d\n", a,b,c);
    trace (&c,&b,a); printf("%d %d %d\n", a,b,c);
    for (i=1; i <=3; i++){
        trace (&b,&a,c);
    }
    printf("%d %d %d\n", a,b,c);
}
```

- In the main function, the trace function is called thrice passing inside the variables a, b, and c. After each call the values of a, b, and c are printed. Then there is a for loop that performs two iterations with calling trace function also twice by passing the variables a, b, and c. Lastly, it prints the final values of a, b, and c.

Exercise #3

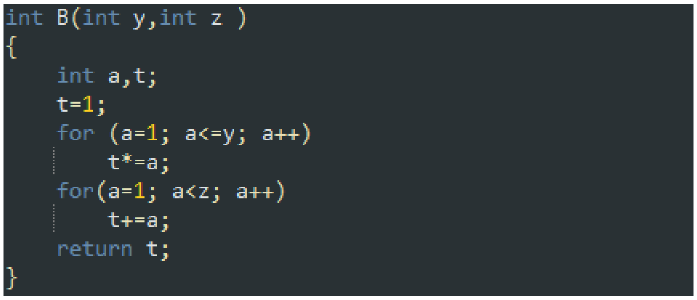


```
C:\Users\Asus\Documents\.GitHub\cpe112-programming-logic-and-design\practice\funct
25 8
27
6 3 27

--------------------------------
Process exited after 0.02088 seconds with return value 7
Press any key to continue . . .
```



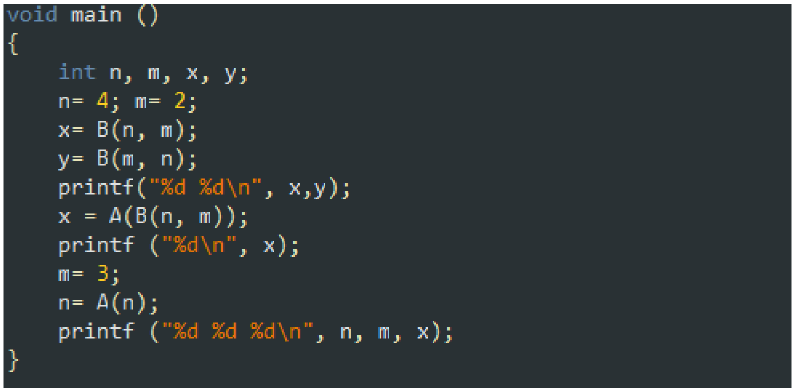
```
int A(int x)
{
    x++;
    return x+1;
}
```

- The A function accepts one integer parameter x. It increments the value first and returns the new value of x that is added with 1.



```
int B(int y,int z )
{
    int a,t;
    t=1;
    for (a=1; a<=y; a++)
        t*=a;
    for(a=1; a<z; a++)
        t+=a;
    return t;
}
```

- The B function receives two integer parameters y and z and returns the value of t. A t variable is declared with a value inside the function which is then modified through the for loop. In the first for loop, t is multiplied with a while a is less than or equal to y. Then, in the second for loop, a is added to t while a is less than z.



```
void main ()
{
    int n, m, x, y;
    n= 4; m= 2;
    x= B(n, m);
    y= B(m, n);
    printf("%d %d\n", x,y);
    x = A(B(n, m));
    printf ("%d\n", x);
    m= 3;
    n= A(n);
    printf ("%d %d %d\n", n, m, x);
}
```

- In the main function, the B function is called twice passing the values of m and n. Its returned value is then assigned to variables x and y which are then printed to the output. The value of x is then modified having assigned the return value of A(B(n, m)). The last function call inside the main is the function A passing to it the value of n and assigned as a new value of n. Lastly, it prints the values of n, m, and x.