Main

In this exercise you will perform the following operations in order. If one of the operations is a function call, then you will fill in the function call card that matches the operation with the appropriate argument values and continue to the named function sheet. Then you will wait until you get a value back from the function before doing your next operation.

```
int u=2, w=5;

int computed = sum(u,w);
int exact = (w * (w + 1))/2 - (u * (u-1))/2;

u = 1;
w = 4;

computed = sum(u,w);
exact = (w * (w + 1))/2;
```

Function trace (fill in the values as you read through the code line by line. If a value changes put the new value in cell on the next row down, that is directly underneath the variable name

Line	u	w	computed	exact	exact == computed
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

Main function call cards

_____	sum(_____ ,____)

_____	sum(_____ ,____)

_____	sum(_____ ,____)

_____	sum(_____ ,____)

_____	sum(_____ ,____)

sum

In this exercise you will perform the operations of the following function. When you are called, you will copy arguments from your call into the corresponding parameters. If one of the operations is a function call, then you will fill out the function call card that matches the operation to the named function. When you get to a return statement you will return the value specified to the caller (the sheet that gave you the function call) of your function.
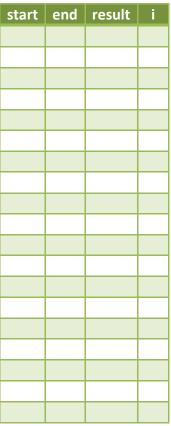
```
int sum (int start, int end) {
    int result = 0; // nothing has been summed yet.
    for (int i = start; i <= end; ++i) {
        result = result + i; // accumulate the sum
    }
    return result;
}
```

Function trace (fill in the values as you read through the code line by line. If a value changes put the new value in cell on the next row down, that is directly underneath the variable name.

First Call

start	end	result	i

Second Call

start	end	result	i

Third Call

start	end	result	i

Main

In this exercise you will perform the following operations in order. If one of the operations is a function call, then you will fill in the function call card that matches the operation with the appropriate argument values and continue to the named function sheet. Then you will wait until you get a value back from the function before doing your next operation.

```
int u=20, w=3;

alphaBeta(u,w);

u = 1;
w = sum(w,u);

alphaBeta(w,u);
```

Function trace (fill in the values as you read through the code line by line. If a value changes put the new value in cell on the next row down, that is directly underneath the variable name

Line	u	w
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		

Main function call cards

_____ sum(_____ ,_____)	_____ alphaBeta(_____ ,_____)
_____ sum(_____ ,_____)	_____ alphaBeta(_____ ,_____)
_____ sum(_____ ,_____)	_____ alphaBeta(_____ ,_____)
_____ sum(_____ ,_____)	_____ alphaBeta(_____ ,_____)
_____ sum(_____ ,_____)	_____ alphaBeta(_____ ,_____)

AlphaBeta

In this exercise you will perform the operations of the following function. When you are called, you will copy arguments from your call into the corresponding parameters. If one of the operations is a function call, then you will fill out the function call card that matches the operation to the named function. When you get to a return statement you will return the value specified to the caller (the sheet that gave you the function call) of your function.

```
void alphaBeta(int u, int w) {
    if (alpha(u, w) || beta(u, w)) {
        println(alpha(u, w));
    }
    else {
        println(beta(u, w));
    }
}
```

Function trace (fill in the values as you read through the code line by line. If a value changes put the new value in cell on the next row down, that is directly underneath the variable name.

Call	Line	u	w	alpha(u,w)	beta(u,w)	Printed output
1	0					
	1 (if)					
	2 (println)					
	3 (println)					
2	0					
	1 (if)					
	2 (println)					
	3 (println)					
3	0					
	1 (if)					
	2 (println)					
	3 (println)					
4	0					
	1 (if)					
	2 (println)					
	3 (println)					

AlphaBeta function call cards

_____ alpha(____ ,___)	_____ beta(____ ,___)
_____ alpha(____ ,___)	_____ beta(____ ,___)
_____ alpha(____ ,___)	_____ beta(____ ,___)
_____ alpha(____ ,___)	_____ beta(____ ,___)
_____ alpha(___ ,___)	_____ beta(____ ,___)

Alpha

In this exercise you will perform the operations of the following function. When you are called, you will copy arguments from your call into the corresponding parameters. If one of the operations is a function call, then you will fill out the function call card that matches the operation to the named function. When you get to a return statement you will return the value specified to the caller (the sheet that gave you the function call) of your function.

```
boolean alpha(int x, int y) {
    x = x/2;
    y = y*2;
    print(x);

    return x>y;
}
```

Function trace (fill in the values as you read through the code line by line. If a value changes put the new value in cell on the next row down, that is directly underneath the variable name

Call	Line	x	y	x/y	return	Printed output
1	0					
	1					
	2					
	3					
	return					
2	0					
	1					
	2					
	3					
	return					
3	0					
	1					
	2					
	3					
	return					
4	0					
	1					
	2					
	3					
	return					

Beta

In this exercise you will perform the operations of the following function. When you are called, you will copy arguments from your call into the corresponding parameters. If one of the operations is a function call, then you will fill out the function call card that matches the operation to the named. When you get to a return statement you will return the value specified to the caller (the sheet that gave you the function call) of your function.

```
boolean beta(int x, int y) {
    print(x);

    return x/y > 10;
}
```

Function trace (fill in the values as you read through the code line by line. If a value changes put the new value in cell on the next row down, that is directly underneath the variable name.

Call	Line	x	y	x/y	return	Printed output
1	0					
	1					
	return					
2	0					
	1					
	return					
3	0					
	1					
	return					
4	0					
	1					
	return					