#### **CSC 111 - WECS Midterm Review Package**

#### Notes on functions:

• Function Definition:

• Function Declaration:

```
<return type> <name>(<input type> <input name>);
```

• Function Call:

```
<name>(<input type> <input name>);
```

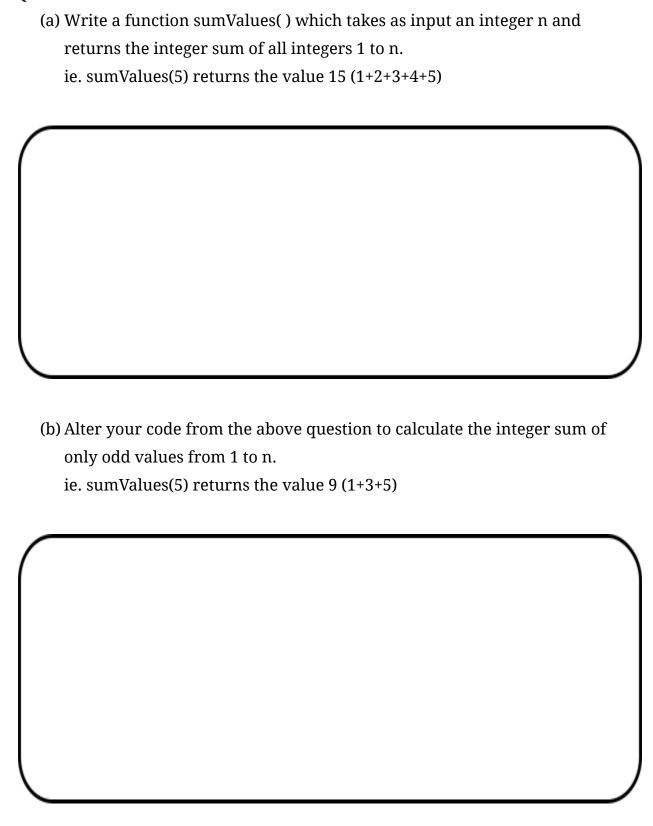
#### **Notes on Expressions**

- Operations (+ \* /) involving **only** integer elements yield an integer result
- Operations(+ \* /) involving at least one floating point value will yield a
  floating point value
- Boolean (true or false) expressions are expressed as integer values in C
  - $\circ$  0 is False, any other value is true ( -5, 999 , 1 are all true)
- && is a logical 'and'
  - o Returns True if both sides are True, otherwise returns false.
- || is a logical 'or'
  - Returns True if at least one side is True, otherwise returns false.
- ++x is equivalent to the following operations in the given order:
  - $\circ$  x = x+1; → use x; [note the embedded assignment statement]
  - o Increment x by 1, then use its **new** value
- $\bullet$  x++ is equivalent to the following operations in the given order:
  - o use x;  $\rightarrow$  x = x + 1; [note the embedded assignment statement]
  - $\circ$  Use the **current** value of x, then increment by 1

## **Notes on Loops**

#### For loops

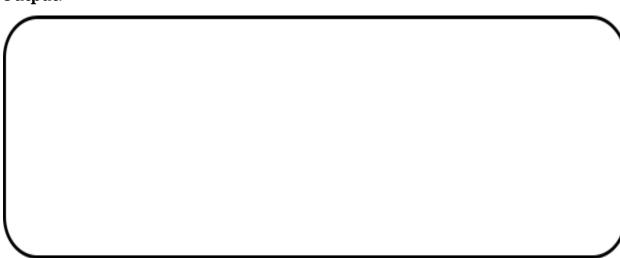
# While loops



What is the output of the syntactically correct C program below?

```
void F1(int x) {
                                    double F3(int x){
  printf("Lastly...\n");
                                       printf("Pear\n");
   if (x\%2 == 0)
                                       double d = 2.75;
       printf("Mango\n");
                                       F2();
   else
                                       return d;
      printf("Strawberry\n");
}
                                    int main() {
void F2(){
  printf("Apple\n");
                                         printf("Let's make fruit
   F1(5);
                                         salad\n");
                                       int x = F3(3);
}
                                         printf("We need %d
                                         orders", x);
                                       return 0;
                                    }
```

### **Output:**



Question 3 (There can be multiple correct answers for parts (a) and (b))

- (a) Which choice is a variable declaration?
  - (i) int a:
  - (ii) float b = 2.5;
  - (iii) c = 17;
- (b) Which of the following is an assignment statement?
  - (i) int a:
  - (ii) float b = 2.5;
  - (iii) c = 17;
- (c) Is the following line of code a function definition or a function call? void myfunc (int x) { <function body> }

#### **Question 4**

```
#include <stdio.h>
#include <stdlib.h>
4 int main(){
   int a, b, c;
    a = 2;
    b = 2;
    c = 0;
    while( b != 0 ){
     a = a + 3;
      b = a\%7;
11
     int c = a + b;
12
       printf("%d %d\n", a, b);
13
   printf("%d %d %d\n", a, b, c);
15
    return 0;
16
17 }
```

- (a) How many **function calls** appear in the code above?
- (b) How many **function definitions** appear in the code above?
- (c) How many assignment statements appear in the code above?
- (d) How many variable declarations appear in the code above?

What is the type and value of each result given that:

```
int a = 6;
int b = 10;
int c = 17;
int d = 187;
float x = 2.0;
float y = 11.1;
float z = 11.6;
```

Expression	Туре	Value
(a%b) + (c%b)		
(b%5) * (b/5)		
(a<=b)     (z <a)< td=""><td></td><td></td></a)<>		
(a<=b) && (z <a)< td=""><td></td><td></td></a)<>		
(b+x) / a		
(b/x) + a		

### Question 6

(a) Convert the following positive binary integer (base-2) to a decimal (base-10)  $1\,1\,0\,0\,1\,0\,1$ 

(b) Convert the following decimal (base-10) to a binary integer (base-2)  $\,$ 

Write a function named **type()** which produces the given output when called from main:

```
int main() {
    type();
    return 0;
}
```

X

-X

- - X

---X

- - X

-X

X