CSC 111 - WECS Midterm Review Package

Notes on functions:

- Function Definition:

```
<return type> <name>(<input type> <input name>){
      <action>
      <return statement>
}
```

- 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
    - 0 is False, any other value is true (-5, 999 , 1 are all true)
- **&&** is a **logical 'and'**
    - 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:
    - x = x+1; → use x; [note the embedded assignment statement]
    - Increment x by 1, then use its **new** value
- **x++** is equivalent to the following operations in the given order:
    - use x; → x = x + 1; [note the embedded assignment statement]
    - Use the **current** value of x, then increment by 1

Notes on Loops

For loops

```
for (<starting point>;<terminating condition>;
<incrementation/decrementation>){
     <action>
}
```

While loops

```
<starting point>
While (<termination condition>){
     <action>
     <incrementation/decrementation>
}
```

Question 1

(a) Write a function sumValues( ) which takes as input an integer n and returns the integer sum of all integers 1 to n.
ie. sumValues(5) returns the value 15 (1+2+3+4+5)

(b) Alter your code from the above question to calculate the integer sum of only odd values from 1 to n.
ie. sumValues(5) returns the value 9 (1+3+5)

Question 2

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

```
void F1(int x){
   printf("Lastly...\n");
   if (x%2 == 0)
       printf("Mango\n");
   else
      printf("Strawberry\n");
}

void F2(){
   printf("Apple\n");
   F1(5);
}

double F3(int x){
   printf("Pear\n");
   double d = 2.75;
   F2();
   return d;
}

int main() {
     printf("Let's make fruit
     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>

int main(){
   int a, b, c;
   a = 2;
   b = 2;
   c = 0;
   while( b != 0 ){
      a = a + 3;
      b = a%7;
      int c = a + b;
      printf("%d %d\n", a, b);
   }
   printf("%d %d %d\n", a, b, c);
   return 0;
}
```

(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?

Question 5

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	**Type**	**Value**
(a%b) + (c%b)		
(b%5) * (b/5)		
(a<=b) || (z<a)		
(a<=b) && (z<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)

56

Question 7

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
```