Functions Exercises

1. The C++ program below has a function sumup that is defined after the call to sumup in function main. Insert a line after the #include so that it is an appropriate function prototype for function sumup.

```
#include <iostream.h>

void main ()
{
 float val;
 float amount;
 int n;
 cin >> val >> n;
 sumup (amount, val, n);
 cout << amount << endl;
}

void sumup (float& sum, float x, int n)
{
 int i;
 sum = 0.0;
 for (i=1; i<=n; i++)
   {
    sum = sum + i * x;
   }
}
```

2. What is wrong with the following function?

```
float resolve (int a, float b, char c, float d)

{
 char e;
 float c;
 ...
}
```

(a) All function parameters a, b, c, and d must be of the same type
(b) No local variables can be declared in function resolve
(c) The function parameter c cannot also be declared as a local variable
(d) The function parameter c can only be declared as a local variable if type char is used

3. Which of the following is a proper function prototype for function resolve in question 2 above? (Hint -- Look closely.)

(a) float resolve (int, float, char, float)
(b) float resolve (int, float, char, float);
(c) resolve (int, float, char, float)
(d) void resolve (int, float, char, float);

4. What return type is used if a function returns no value?

(a) float
(b) int
(c) prototype
(d) void

5. Actual arguments and function parameters must be...?
(a) same number
(b) same number and same types
(c) same types
(d) same number and different types

6. Which of the following best describes what this function does?

```
void increase (int number)
{
 ++number;
 return;
}
```

(a) changes both the local variable number AND the
        value of the actual parameter in the calling program
(b) changes the local variable number, but NOT the
        value of the actual parameter in the calling program
(c) gets an error message from the C++ compiler since it really
        should start with
        void increase (int& number)
(d) is an example of call-by-reference

——SOLUTION——

1.

```
#include <iostream.h>

void sumup (float&, float, int);

void main ()
...
```

2. (c)

3. (b)

4. (d)

5. (b)

6. (b)

Function Defaults, Recursion, Overloading Exercises

1. Create a function prototype for sumup that specifies the default 1.0 for x and 1 for n.

```
#include <iostream.h>

void main ()
{
 float val;
 float amount;
 int n;
 cin >> val >> n;
 sumup (amount, val, n);
 cout << amount << endl;
}

void sumup (float& sum, float x, int n)
{
 int i;
 sum = 0.0;
 for (i=1; i<=n; i++)
   {
    sum = sum + i * x;
   }
}
```

2. Overload the function name manip in the program below. The new function manip has two float parameters num1 and num2 that are both passed by value. This manip function returns the quotient num1 / num2 of those two float values. Don't forget to include a function prototype.

```
#include <iostream.h>

int manip (int, int);

void main ()
{
 int x = 3;
 int y = 7;
 int z;
 float a = 6.8;
 float b = 13.6;
 int c;
 z = manip (x,y);
 c = manip (a,b);
 cout << z << c << endl;
}
int manip (int num1, int num2)
{
 return (num1 * num2);
}
```

3. Which of the following best describes the statement
z = sum (5.2); below?

```
float sum (float = 1.0, float = 0.0);
...
z = sum (5.2);
```

(a) equivalent to z = sum (1.0, 0.0);
(b) equivalent to z = sum (5.2, 0.0);
(c) equivalent to z = sum (5.2, 1.0);
(d) leads to a compile-time error -- not enough arguments to function sum

4. What is (are) the major characteristic(s) of a recursive function?
(a) a function that calls itself every time it is invoked, and has NO way to end without calling itself
(b) a function that has a loop and does NOT call itself
(c) a function that has no loops
(d) a function that usually calls itself, but has a way to end without calling itself

5. What is required in Function Overloading?
(a) Functions must have different number of parameters
(b) Functions must have different number of parameters or at least one parameter must be different type
(c) Functions must have different return types
(d) Functions must have same number of parameters but at least one parameter must be different type

6. Given the following function prototypes...

```
int suspend (int, int);
float suspend (int, int=30, int=25, int=53);
```

...which of the following function calls leads to an ambiguity?
(a) suspend (3);
(b) suspend (3,5);
(c) suspend (3,5,7);
(d) suspend (3,5,7,9);

——SOLUTION——

1.

```
#include <iostream.h>

void sumup (float&, float = 1.0, int = 1);

void main ()
...
```

2.

```
#include <iostream.h>

int manip (int, int);

float manip (float, float);
```

```
void main ()
{
 int x = 3;
 int y = 7;
 int z;
 float a = 6.8;
 float b = 13.6;
 int c;
 z = manip (x,y);
 c = manip (a,b);
 cout << z << c << endl;
}
int manip (int num1, int num2)
{
 return (num1 * num2);
}

float manip (float num1, float num2)
{
 return (num1 / num2);
}
```

3. (b)

4. (d)

5. (b)

6. (b)

Subscripted Variables and Pointers Exercises

1. Given the definition

```
int box[10];
```

what is wrong with the following assignment statement? If there is nothing wrong, say so.

```
box[10] = 20;
```

2. Is the following statement correct? If correct, answer ``Correct''. If not correct, answer ``Wrong'' and give a reason:

In any one array there can be values of different types.

3. Is the following statement correct? If correct, answer ``Correct''. If not correct, answer ``Wrong'' and give a reason:

Passing large arrays into functions is a bad idea because it takes a long time to copy an array from a calling function into a called function.

4. Assume the following definition and initialization:

```
int link[5] = { 2, 3, 4, 0, 1} ;
```

What output does each of the following statements produce:

(a) `cout << link[0];`

(b) `cout << link[link[2]];`

(c) Given any integer k (0<=k<=4), what is the value of the following expression

```
link[link[link[link[link[k]]]]]
```

5. A Fibonacci sequence of numbers is defined as follows: The first two numbers in the sequence are 0 and 1. Then, each additional Fibonacci number is the sum of the two previous numbers in the sequence. Thus, the first ten Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8, 13, 21, and 34.

Complete the following function so that the function generates the first fib_num (fib_num > 2) numbers and puts them in the array F. Do NOT write a recursive function. Declare any local variable(s) you need.

```
void Fib_general (int F[], int fib_num)
{
 // Declare any local variable(s)


 // Initialize the first two numbers in the array F
```

```
    // Use a loop to generate the rest




   }
```

6. Complete the following recursive binary search function which searches a sorted (in ascending order) array A for a given value Element. The parameter First is the first index of the array being searched, while the parameter Last is the last index of the array being searched.

```
   int BinSearch (int A[], int Element, int First, int Last)
   {
    int Mid;

    if(________________)
    {
      return(-1);       // not found;
    }
    else
    {
      Mid = (First + Last) / 2;
      if(________________)
          return(Mid);
      else
      {
          if (Element < A[Mid])
             return(________________);
          else
             return(________________);
      }
    }
   }
```

7. Write 3 lines of C++ code below to (1) declare the variable zolish to be a pointer to an integer, (2) make zolish point to the location where xerxes is, and (3) store the number 45 in the location where zolish is pointing.

```
  int xerxes;
  xerxes = 37;
```

8. Now what will

```
  cout << xerxes << endl;
```

print?

9. Write 3 lines of C++ code below to (1) declare the variable salary to be a pointer to a float type value, (2) dynamically allocate a float location and make salary point to it, and (3) store the value 150.75 in the location where salary is pointing.

10. Write a line of C++ code to de-allocate the float location that you dynamically allocated in part 3.

11. Assume the following declarations:

```
int pixel;
int pad;
int *plane;
int *dots;
```

What is stored in pixel, pad, plane, and dots after the following statements?

```
pixel = 24;
pad = 57;
plane = &pixel;
dots = &pad;
```

——SOLUTION——

1. Can't assign using index 10; valid indices are 0 to 9.

2. Wrong. Every item in an array must be of the same type.

3. Wrong. Arrays are always passed by reference.

4. (a) 2, (b) 1, (c) k

5.

```
// Declare any local variable
int i;

// Initialize the first two numbers in the array F
F[0] = 0;
F[1] = 1;

// Use a loop to generate the rest
for (i=2; i < fib_num; i++)
{
   F[i] = F[i-1] + F[i-2];
}
```

6.

```
int BinSearch (int A[], int Element, int First, int Last)
{
 int Mid;

 if(First > Last)
 {
   return(-1);       // not found;
 }
 else
 {
   Mid = (First + Last) / 2;
   if(Element == A[Mid])
       return(Mid);
   else
   {
       if (Element < A[Mid])
          return(BinSearch(A,Element,First,Mid-1));
       else
          return(BinSearch(A,Element,Mid+1,Last);
```

```
      }
    }
   }
```

7.
```
  int* zolish;
  zolish = &xerxes;
  *zolish = 45;
```

8. 45

9.
```
  float* salary;
  salary = new float;
  *salary = 150.75;
```

10.
```
  delete salary;
```

11. pixel = 24; pad = 57; lane = pointer to pixel; dots = pointer to pad