

Homework 7: Functions

Q 1. TRUE/FALSE STATEMENTS

1. A function must either accept one or more arguments or return a value.
2. The compiler compares the definition of a function with its declaration.
3. A function without arguments need not use the () suffix.
4. A function that returns a value can be treated as an expression.
5. A function can't both return void and also use void in the argument list.
6. A function can call other functions except itself.
7. The function body must be enclosed within curly braces even if the body comprises a single statement.
8. It is impossible for a function **f(x)** to change x because x is destroyed after the function terminates.
9. A function can be used for its side effect or return value or both.
10. A **while** loop can run forever, but not a recursive function.
11. A program that uses **int main(void)** can not be run with an argument.
12. Every time the keyword **extern** is used, separate storage is allocated for the variable.

Q 2. FILL IN THE BLANKS

1. The _____ of a function determines its usage while the _____ provides the code that will be executed.
2. A function that doesn't return a value must be declared as _____.
3. A _____ function repeatedly calls itself.
4. The file `stdio.h` contains the _____ of the standard I/O functions.
5. Pass-by-value implies that actual arguments are _____ to the formal arguments of a function.
6. The parameters and local variables of a function are stored in a region of memory called _____.
7. A _____ variable defined inside a function is created only once irrespective of the

number of calls made to the function.

8. The _____ statement terminates the current function, but the _____ function terminates the program no matter where it is invoked from.
9. The **main** function is never _____ in a program.
10. The **extern** keyword is used to _____ a variable. The variable must be _____ elsewhere.
11. By default, a variable defined inside a function belongs to the _____ storage class.
12. The _____ keyword is used to store a variable in high-speed memory directly connected to the CPU.

Q 3. MULTIPLE-CHOICE QUESTIONS

1. The statement **return sum;** in a function
 - (A) terminates the program after transmitting the value of sum to the operating system,
 - (B) transmits the value of sum to the caller of the function
 - (C) terminates the function,
 - (D) B and C, (E) A, B and C.
2. The declaration of a function
 - (A) must precede its usage in **main**,
 - (B) must be followed by its definition,
 - (C) must use the same parameter names as the definition,
 - (D) A and B, (E) None of these.
3. A local variable of a function
 - (A) can be transmitted to the caller using **return**,
 - (B) is destroyed before it can be returned,
 - (C) cannot be transmitted using **return**,
 - (D) none of these.
4. When the name of an array is passed as an argument to a function,
 - (A) the array is copied inside the function,
 - (B) the address of the first array element is copied,

- (C) the function can determine the size of the array using **sizeof**,
 - (D) none of these.
5. A variable declared outside a function is
- (A) visible inside **main**,
 - (B) not visible in any function,
 - (C) visible from the point of its declaration to end of program,
 - (D) none of these.
6. If a variable is declared inside a function as static,
- (A) it is initialized only once,
 - (B) it retains its value between multiple function calls,
 - (C) it is destroyed only on program termination,
 - (D) all of these.

Q 4. Write a program containing a function named volume which returns the volume of a sphere. The function is passed the radius as a single argument.

Q 5. The volume of a cylinder is the product of the area of the base ($\pi * r * r$) and the height. Write a program that accepts the radius and height from the user and invokes a function named **volume** which in turn invokes a function named **area**.

Q 6. Write a program that displays the Fibonacci numbers up to n terms using

- (i) a recursive function named **fibonacci_rec**,
- (ii) an iterative function named **fibonacci_ite**.

The value of n is set by user input. The program must track and print the total number of recursive calls and loop iterations made by the two functions. As you increase the value of n, what observations can you make from the results?