Some more Technical Questions

1) How do you construct an increment statement or decrement statement in C?

Answer: There are actually two ways you can do this. One is to use the increment operator ++ and decrement operator �. For example, the statement �x++� means to increment the value of x by 1. Likewise, the statement �x ��� means to decrement the value of x by 1. Another way of writing increment statements is to use the conventional + plus sign or � minus sign. In the case of �x++�, another way to write it is �x = x +1?.

2) Some coders debug their programs by placing comment symbols on some codes instead of deleting it. How does this aid in debugging?

Answer: Placing comment symbols /* */ around a code, also referred to as �commenting out�, is a way of isolating some codes that you think maybe causing errors in the program, without deleting the code. The idea is that if the code is in fact correct, you simply remove the comment symbols and continue on. It also saves you time and effort on having to retype the codes if you have deleted it in the first place.

3) What is the equivalent code of the following statement in WHILE LOOP format?

[c]
```
for (a=1; a<=100; a++)
printf ("%d\n", a * a);
```
[/c]

Answer: [c]
```
a=1;
while (a<=100) {
printf ("%d\n", a * a);
a++;
}
```
[/c]

4) What is spaghetti programming?

Answer: Spaghetti programming refers to codes that tend to get tangled and overlapped throughout the program. This unstructured approach to coding is usually attributed to lack of experience on the part of the programmer. Spaghetti programing makes a program complex and analyzing the codes difficult, and so must be avoided as much as possible.

5) In C programming, how do you insert quote characters (� and �) into the output screen?

Answer: This is a common problem for beginners because quotes are normally part of a printf statement. To insert the quote character as part of the output, use the format specifiers \� (for single quote), and \� (for double quote).

6) What is the use of a �\0' character?

Answer: It is referred to as a terminating null character, and is used primarily to show the end of a string value.

7) What is the difference between the = symbol and == symbol?

Answer: The = symbol is often used in mathematical operations. It is used to assign a value to a given variable. On the other hand, the == symbol, also known as ⍰equal to⍰ or ⍰equivalent to⍰, is a relational operator that is used to compare two values.

8) Which of the following operators is incorrect and why? (>=, <=, <>, ==)

Answer: <> is incorrect. While this operator is correctly interpreted as ⍰not equal to⍰ in writing conditional statements, it is not the proper operator to be used in C programming. Instead, the operator != must be used to indicate ⍰not equal to⍰ condition.

9) Can the curly brackets { } be used to enclose a single line of code?

Answer: While curly brackets are mainly used to group several lines of codes, it will still work without error if you used it for a single line. Some programmers prefer this method as a way of organizing codes to make it look clearer, especially in conditional statements.

10) What are header files and what are its uses in C programming?

Answer: Header files are also known as library files. They contain two essential things: the definitions and prototypes of functions being used in a program. Simply put, commands that you use in C programming are actually functions that are defined from within each header files. Each header file contains a set of functions. For example: stdio.h is a header file that contains definition and prototypes of commands like printf and scanf.

11) Can I use ⍰int⍰ data type to store the value 32768? Why?

Answer: No. ⍰int⍰ data type is capable of storing values from -32768 to 32767. To store 32768, you can use ⍰long int⍰ instead. You can also use ⍰unsigned int⍰, assuming you don⍰t intend to store negative values.

12) Can two or more operators such as \n and \t be combined in a single line of program code

Answer: Yes, it⍰s perfectly valid to combine operators, especially if the need arises. For example: you can have a code like ⍰ printf (⍰Hello\n\n\⍰World\⍰⍰) ⍰ to output the text ⍰Hello⍰ on the first line and ⍰World⍰ enclosed in single quotes to appear on the next two lines.

13) Why is it that not all header files are declared in every C program?

Answer: The choice of declaring a header file at the top of each C program would depend on what commands/functions you will be using in that program. Since each header file contains different function definitions and prototype, you would be using only those header files that would contain the functions you will need. Declaring all header files in every program would only increase the overall file size and load of the program, and is not considered a good programming style.

14) When is the void keyword used in a function?

Answer: When declaring functions, you will decide whether that function would be returning a value or not. If that function will not return a value, such as when the purpose of a function is to display some outputs on the screen, then void is to be placed at the leftmost part of the function header. When a return value is expected after the function execution, the data type of the return value is placed instead of void.

15) What are compound statements?

Answer: Compound statements are made up of two or more program statements that are executed together. This usually occurs while handling conditions wherein a series of statements are executed when a TRUE or FALSE is evaluated. Compound statements can also be executed within a loop. Curly brackets { } are placed before and after compound statements.

16) Write a loop statement that will show the following output:

```
1
12
123
1234
12345
```

Answer: [c]

```
for (a=1; a<=5; i++) {
for (b=1; b<=a; b++)
printf("%d",b);
printf("\n");
}
```

[/c]

17) What is wrong in this statement? scanf(%d,whatnumber);

Answer: An ampersand & symbol must be placed before the variable name whatnumber. Placing & means whatever integer value is entered by the user is stored at the address of the variable name. This is a common mistake for programmers, often leading to logical errors.

18) How do you generate random numbers in C?

Answer: Random numbers are generated in C using the rand() command. For example: anyNum = rand() will generate any integer number beginning from 0, assuming that anyNum is a variable of type integer.

19) What could possibly be the problem if a valid function name such as tolower() is being reported by the C compiler as undefined?

The most probable reason behind this error is that the header file for that function was not indicated at the top of the program. Header files contain the definition and prototype for functions and commands used in a C program. In the case of tolower(), the code #include must be present at the beginning of the program.

20) What does the format %10.2 mean when included in a printf statement?

Answer: This format is used for two things: to set the number of spaces allotted for the output number and to set the number of decimal places. The number before the decimal point is for the allotted space, in this case it would allot 10 spaces for the output number. If the number of space occupied by the output number is less than 10, addition space characters will be inserted before the actual output number. The number after the decimal point sets the number of decimal places, in this case, its 2 decimal spaces.

21) What is wrong with this statement? myName = Robin;

Answer: You cannot use the = sign to assign values to a string variable. Instead, use the strcpy function. The correct statement would be: strcpy(myName, Robin);

22) How do you determine the length of a string value that was stored in a variable?

Answer: To get the length of a string value, use the function strlen(). For example, if you have a variable named FullName, you can get the length of the stored string value by using this statement: I = strlen(FullName); the variable I will now have the character length of the string value.

23) Is it possible to initialize a variable at the time it was declared?

Answer: Yes, you dont have to write a separate assignment statement after the variable declaration, unless you plan to change it later on. For example: char planet[15] = Earth; does two things: it declares a string variable named planet, then initializes it with the value Earth.

24) What are the different file extensions involved when programming in C?

Answer: Source codes in C are saved with .C file extension. Header files or library files have the .H file extension. Every time a program source code is successfully compiled, it creates an .OBJ object file, and an executable .EXE file.

25) What are reserved words?

Answer: Reserved words are words that are part of the standard C language library. This means that reserved words have special meaning and therefore cannot be used for purposes other than what it is originally intended for. Examples of reserved words are int, void, and return.

26) What are linked list?

Answer: A linked list is composed of nodes that are connected with another. In C programming, linked lists are created using pointers. Using linked lists is one efficient way of utilizing memory for storage.

27) What are binary trees?

Answer: Binary trees are actually an extension of the concept of linked lists. A binary tree has two pointers, a left one and a right one. Each side can further branch to form additional nodes, which each node having two pointers as well.

28) Not all reserved words are written in lowercase. TRUE or FALSE?

Answer: FALSE. All reserved words must be written in lowercase; otherwise the C compiler would interpret this as unidentified and invalid.

29) What is wrong with this program statement? void = 10;

Answer: The word void is a reserved word in C language. You cannot use reserved words as a user-defined variable.

30) Is this program statement valid? INT = 10.50;

Answer: Assuming that INT is a variable of type float, this statement is valid. One may think that INT is a reserved word and must not be used for other purposes. However, recall that reserved words are express in lowercase, so the C compiler will not interpret this as a reserved word.

31) What is a newline escape sequence?

Answer: A newline escape sequence is represented by the \n character. This is used to insert a new line when displaying data in the output screen. More spaces can be added by inserting more \n characters. For example, \n\n would insert two spaces. A newline escape sequence can be placed before the actual output expression or after.

32) What is output redirection?

Answer: It is the process of transferring data to an alternative output source other than the display screen. Output redirection allows a program to have its output saved to a file. For example, if you have a program named COMPUTE, typing this on the command line as COMPUTE >DATA can accept input from the user, perform certain computations, then have the output redirected to a file named DATA, instead of showing it on the screen.

33) What is the difference between functions abs() and fabs()?

Answer: These 2 functions basically perform the same action, which is to get the absolute value of the given value. Abs() is used for integer values, while fabs() is used for floating type numbers. Also, the prototype for abs() is under , while fabs() is under .

34) Write a simple code fragment that will check if a number is positive or negative.

Answer: [c]

```
If (num>=0)
printf("number is positive");
else
printf ("number is negative");
[/c]
```

35) What does the function toupper() do?

Answer: It is used to convert any letter to its upper case mode. Toupper() function prototype is declared in . Note that this function will only convert a single character, and not an entire string.

36) Which function in C can be used to append a string to another string?

Answer: The strcat function. It takes two parameters, the source string and the string value to be appended to the source string.

37) Dothese two program statements perform the same output? 1) scanf(⍰%c⍰, &letter); 2) letter=getchar()

Answer: Yes, they both do the exact same thing, which is to accept the next key pressed by the user and assign it to variable named letter.

38) What is the difference between text files and binary files?

Answer: Text files contain data that can easily be understood by humans. It includes letters, numbers and other characters. On the other hand, binary files contain 1s and 0s that only computers can interpret.

39) is it possible to create your own header files?

Answer: Yes, it is possible to create a customized header file. Just include in it the function prototypes that you want to use in your program, and use the #include directive followed by the name of your header file.

40) What is dynamic data structure?

Answer: Dynamic data structure provides a means for storing data more efficiently into memory. Using dynamic memory allocation, your program will access memory spaces as needed. This is in contrast to static data structure, wherein the programmer has to indicate a fix number of memory space to be used in the program.

41) The % symbol has a special use in a printf statement. How would you place this character as part of the output on the screen?

Answer: You can do this by using %% in the printf statement. For example, you can write printf(⍰10%%⍰) to have the output appear as 10% on the screen.

42) What are the advantages and disadvantages of a heap?

Answer: Storing data on the heap is slower than it would take when using the stack. However, the main advantage of using the heap is its flexibility. That�s because memory in this structure can be allocated and remove in any particular order. Slowness in the heap can be compensated if an algorithm was well designed and implemented.