Review Question

1. A C program contains one or more functions. And this function is defined as a group of statements that perform a well-defined task. The main() functions is most important function and is a part of every C program. Statements are executed in a logical sequence that we written.
2. Declaration gives the compiler name of the variable, type of value it holds and the initial value if any it takes. On the other hand, Definition is where the variable gets stored and make memory that is allocated for the variable.
3. Memory is reserved using data type in the variable declaration. A programming language implementation has predefined sizes for its data types.
4. Identifiers are names given to program elements such as variables, arrays and function. And Keywords are a set of reserved word in C that cannot be used as an identifier.
5. A variable is defined as a meaningful name given to a data storage location in the computer memory. And C language supports two kind of variables. One is Numeric Variables and the other is Character variables. Constant is not same as variable. Constants are identifiers whose values do not change eternally. While values of variables can be changed at any time, values of constants can never be changed.
6. The data type of a variable signifies the set of values that a data item can take and the operations that can be performed on the item.
7. The char data type is of one byte and is used to store single character. The int data type is of two byte and is used to store integer numbers. The float data type is of four byte and is used to store floating point numbers. The double data type is of eight byte and is used to store big floating point numbers.
8. The file has some in-built functions. By simply including this file in our code, we can use this function directly. It means Standard Input/Output that it has functions for input and output of data like reading values from the keyboard and printing the results on the screen.
9. To use function(we don't know the details of how these functions work) that is reserved in C, the header file should be included and then C compilers automatically can perform their operations.
10. printf function is used to display information required by the user and also prints the values of the variables. scanf function is used to read formatted data from the keyboard.
11. available operators in C are 'Arithmetic operators', 'Equality operators', 'Unary operators', 'Bitwise operators', 'Comma operators', 'Relational operators', 'Logical operators', 'Conditional operator','Assighment operators' and 'Sizeof operator'.
12. operator precedence chart

(), [] , . ,->	1
++(postfix),--(postfix)	2
++(prefix),--(prefix),+(unary), -(unary) *(indirection), &(address), sizeof	3
*, /, %	4
+, -	5
<< >>	6
<, <=, >, >=	7
==, !=	8
&	9
^	10
|	11
&&	12
||	13
?:	14
=	15
,(comma)	16

13. Type conversion or typecasting of variables refers to changing a variable of one data type into another. While type conversion is done implicitly, typecasting has to be done explicitly by the programmer.
14. Decision control statements can alter the flow of a sequence of instructions. There are four kinds of Decision Control Statement. a.if b.if-else c.if-else-if d.switch-case. **a. if**: When the condition defined by if statement is True, the statement in if body will execute. **b. if-else**: When the condition defined by if statement is True, the statement in if body will execute. But When the condition defined by if statement isn't true, the statement in else block will execute. **c. if-else-if**: construct works in the same way as a normal if statement. But if first condition defined by if statement is False, The second condition defined by else-if statement will be checked. **d.switch-case**: switch block defines a condition and a lot of case have statement. According to condition, corresponding case will execute.
15. Iterative Statements are used to repeat the execution of a sequence of statement until the condition becomes false. There are three kinds of iterative statements in C. a. while loop b. do-while loop c. for loop.
16. This is very useful When we need to evaluate multiple conditions. and It would be easy to debug and easy to read and understand.
17. A function is a group of statement that together perform a task. The functions is needed for understanding, coding, and testing multiple separate functions is easier than doing the same for one big function. and programmers can write their own functions and use it everywhere.
18. Function Declaration enables the compiler to make a check on the arguments used while calling that function. Function Declaration concludes function name, return data type, and arguments. Function Definition is a tool that allows the

function to be allocated space in the memory. Function Definition concludes function header and body. Function header is same as Function Declaration and Function header consists of statements.

19. Because it enables the compiler to make a check on the arguments used while calling that function.

20. Function call statement invokes the function. Function call consists of function name and arguments(variables). Arguments may be passed in the form of expressions to the called function. Arguments are first evaluated and converted to the type of formal parameter and the body of the function get executed.

21. In call by value method, the called function creates new variables to store the value of the arguments passed to it. Then the called function uses a copy of the actual arguments to perform its intended task. In call by reference method, a function receives an implicit reference to the argument, rather than a copy of its value. ex) following the code(call by value):

```
#include <stdio.h>
void cal(int a);
int main()
{
        int a=3;
        cal(a);
        printf("%d",a);
}

void cal(int a)
{
        a=a+3;

}
```

In call by value, the output will be 3 because the called function uses a copy of the actual arguments. on the other hand ex) following the code(call by references):

```
#include <stdio.h>
void cal(int* a);
int main()
{
        int a=3;
        cal(&a);
        printf("%d",a);
}

void cal(int *a)
{
        *a=*a+3;
```

}

In call by reference, the output will be 6 because the function can use the reference, do not use a copy of the actual arguments. Hence the function can modify the value of the variable and that change will be reflected in the calling function.

22. A pointer is a variable that contains the memory location of another variable. and It represents the location of a data item such as array element. It used to not only pass arrays or strings as function arguments, but also provide an alterate way to access the individual elements of an array.

23. The NULL pointers is a pointer that does not point to any valid memory address. And the void pointer is a pointer that can point to variables of any data type.

24. void pointer is declared like a normal pointer variable but using the void keyword as the pointer's data type. The void pointer will not point to any data then you need to cast a void pointer to another kind of pointer before using it.

25. When we define pointer to pointers, the first pointer contains the address of second pointer that points to the location that contains actual value. To use pointer to pointers, you just add an asterisk* for each level of references.

Multiple-choice Questions

1.(b)
2.(c)
3.(d)
4.(d)
5.(b)
6.(b)
7.(d)
8.(c)
9.(d)
10.(d)
11.(c)
12.(b)
13.(c)
14.(a)
15.(b)

True or False

1.F
2.T
3.F
4.F
5.T
6.F

7.T
8.F
9.F
10.F
11.T
12.F
13.T
14.T
15.F
16.T
17.T
18.F
19.T
20.F
21.T
22.F
23.T
24.T
25.T

Fill in the Blanks
1. Dennis Ritchie
2. main() function
3. single characters
4. operating system
5. %
6. unary
7. typecasting
8. default
9. printf
10. return 0
11.\n
12. const
13. sizeof
14. %d
15. %x,%X
16. ‘-’
17. calling program
18. calling function
19. argument
20. fuction header, function body
21. call by reference
22. 1byte

23. NULL
24. =
25. *