**[A] Answer the following Questions.**

(a) What do you mean by storage class of a variable?  
- The storage class of a variable defines its scope, lifetime, and location (memory or CPU registers). If not explicitly specified, the compiler assigns a default storage class based on the context. The storage class determines where the variable's value is stored and how it is accessed.

(b) Why register storage class is required?  
- The register storage class is used to suggest that the variable should be stored in a CPU register instead of RAM for faster access. This is particularly useful for variables that are heavily used within loops or critical sections of code, as accessing CPU registers is much quicker than accessing memory.

(c) How many storage classes are there in c and why they are used?  
- There are four storage classes in C: auto, register, static, and extern.

* **auto**: The default storage class for local variables, stored in memory.
* **register**: Suggests storing the variable in a CPU register for faster access.
* **static**: Preserves the variable's value between function calls and limits its scope to the file or function.
* **extern**: Declares a global variable that is defined in another file, allowing for cross-file variable sharing.

(d) Why someone needs static storage variable in a program?  
- A static storage variable is needed to retain its value between function calls and to limit its scope to the file or function where it is declared.

(e) Which storage class is termed as default storage class, if the storage class of a variable is not mentioned in a program? Explain its working with the help of a suitable example.  
- The default storage class is auto if not specified. It applies to local variables within functions, meaning they are stored in memory and have automatic duration (created at function entry and destroyed at exit). For example, if we declared int x = 10, we didn’t declared the storage class, so the compiler automatically add the default storage class which is auto.

(f) Can you store a floating point variable in CPU Registers, Explain?  
- Yes, you can use the register storage class for a float or double variable, but the compiler will treat it as an automatic variable if it can't store it in a register. You won't get an error message; it will just be stored in memory instead.  
  
(g) What do you mean by scope and life of a variable?

- The scope of a variable in C refers to the region of the program where the variable can be accessed or used. The life (or lifetime) of a variable refers to the duration during which the variable exists in memory and holds a value.

(h) Justify where to use which storage class.  
-(a) Use static to keep a variable's value between function calls.  
-(b) Use register for frequently used variables, like loop counters, for faster access.  
-(c) Use extern for variables shared across multiple functions to avoid passing them as arguments.  
-(d) Use auto (default) if none of the above needs apply.

(i) What do you mean by external variable and where it is defined?  
- An external variable is a global variable that can be accessed by multiple files in a program. It is defined outside of any function, usually at the top of a file, and declared with the extern keyword in other files where it is used.

(j) What is the difference between auto and static variables in a program?

- ***auto*** variables are local to a function, created when the function is called, and destroyed when it exits. ***static*** variables are also local to a function but retain their value between function calls and exist for the entire program's duration.

(k) What do you mean by definition and declaration of a variable?

- A variable declaration tells the compiler about the variable's type and name, but doesn't allocate memory. A variable definition allocates memory for the variable and may also initialize it. For example, extern int x; is a declaration, while int x = 10; is a definition.

A screenshot of a black screen

Description automatically generatedA screenshot of a computer

Description automatically generated  
**[B] What is the output of the following programs:  
a.)**

**A screenshot of a computer program

Description automatically generatedA black screen with white text

Description automatically generatedb.)**

**A screenshot of a computer program

Description automatically generated**

A screen shot of a black screen

Description automatically generated**c.)**

**A black screen with white text

Description automatically generatedA screenshot of a computer

Description automatically generatedd.)**

**[C] State whether the following statements are True or False:**

(a) The register storage class variables cannot hold float values. **False**  
(b) The value of an automatic storage class variable persists between various function invocations. **False**  
c) If the CPU registers are not available, the register storage class variables are treated as static storage class variables. **False**  
(d) The default value for automatic variable is zero. **False**  
(e) If a global variable is to be defined, then the extern keyword is necessary in its declaration. **False**  
(f) If we try to use register storage class for a float variable the compiler will flash an error message. **False**  
(g) Storage for a register storage class variable is allocated each time the control reaches the block in which the variable is present. **True**  
(h) An extern storage class variable is not available to the functions that precede its definition, unless the variable is explicitly declared in the above functions. **True**  
(i) The life of static variable is till the control remains within the block in which it is defined. **False**  
(j) The address of a register variable is not accessible. **True**

**A screenshot of a computer program

Description automatically generated[D] Following program calculates the sum of digits of the number 25634. Go through it and find out why is it necessary to declare the storage class of the variable sum as static.**

A screen shot of a black screen

Description automatically generated  
  
  
  
  
  
The variable sum must be static because:

It needs to maintain a running total across all recursive calls.  
Without static, each recursion would create a new sum with garbage value, ***static*** ensures one shared sum variable persists throughout all recursive iterations. That’s why changing static int sum; to int sum; would break the program’s functionality.