**Keywords, Identifier, Literals, Operators and Expression Assignment**

**Choose all valid identifiers**

1. int int

**Invalid**: int is a keyword in C, and cannot be used as an identifier.

1. int \_numvalue

**Valid**: Identifiers can start with an underscore (\_) and can include letters and numbers, so \_numvalue is a valid identifier.

1. float price\_money

**Valid**: Spaces are not allowed in identifiers, so this is **invalid** because it has a space.

1. char name1234567890123456789012345678901234567890

**Invalid**: C allows long identifiers (within the limits of the implementation). This is a invalid identifier because it is a very long one (more than necessary).

1. char name value

**Invalid**: Identifiers cannot contain spaces. So, name value is **invalid**

1. char $name

**Invalid**: The rules for valid identifiers in C specify that they must begin with a letter (uppercase or lowercase) or an underscore (\_), followed by letters, digits, or underscores.

So, $name would not be a valid variable name in C.

**2. What is the meaning of the following keywords, show the usage**

**a. auto**

Auto keyword automatically refers to the type of variable without declaration and allocates memory

**b. extern**

Extern keyword is used to refer to functions or variables that are defined in another file

**c. volatile**

Volatile keyword describes that the value of variable can be changed at any time

**d. sizeof**

sizeof keyword gives the size of datatype or variable

**e. const**

const keyword is used for the variables that cannot be altered after initialization.

**3. Explain the difference between the following variables.**

a. char \*ptr = “ABC”;

b. char arr[]=”ABC”;

**Can you manipulate the contents of ptr? Why?**

echnically, you can manipulate ptr (since it’s a pointer) to point to another string or another location, but you cannot modify the content of the string "ABC" itself (this is undefined behavior in C).

**Can you manipulate the contents of arr? Why?**

Yes, you can modify the contents of arr, as it is a mutable array of characters.

**Which one of the above is a string literal?**

"ABC" in both cases is a **string literal** because it’s a constant sequence of characters enclosed in double quotes.

**4. Predict the output of the following code .**

A screenshot of a computer code

Description automatically generated

5 4

4 4

**5. Refer the code snippet. It fails with error. Fix it.**

A white rectangular object with a black border

Description automatically generated with medium confidence

A white background with black text

Description automatically generated

const int num; was not initialized, which caused the error. The line num = num + k; tries to modify a constant, which is illegal in C.

**6. Consider the following code snippet. Evaluate the value of f1, f2 and f3.**

A white rectangular object with a gray background

Description automatically generated

**f1 = 3.00**: The division i / j is integer division, so the result is truncated to 3.

**f2 = 3.33**: The division (float) i / j ensures floating-point division, so the result is 3.3333....

**f3 = 3.00**: First, the integer division i / j is performed, resulting in 3, and then it’s cast to float, giving 3.0.