

Name: Harshit Jain User ID: hmj5262
--

Problem 1

The binary value 01010000010100110011001000110011 is equivalent to the decimal number 1347629619 in a 32-bit unsigned integer.

The binary value 01010000010100110011001000110011 is equivalent to the ASCII codes for the "PS23".

The binary value 01010000010100110011001000110011 is equivalent to the floating point number 1.649969458580017 in IEEE 754 single precision format.

The binary value 01010000010100110011001000110011 is equivalent to the array $\text{int8 } t[4] = 80, 83, 50, 51$.

Problem 2

Problem 3

1. Static typing and dynamic typing are two different approaches to type checking in programming languages.

In **static typing**, the type of each variable and expression is declared explicitly by the programmer. This information is then used by the compiler to check for type errors at compile time. Static typing can help to catch errors early and make code more reliable. Some examples of statically typed languages are Java, C++, and C#.

In **dynamic typing**, the type of each variable and expression is inferred from its value at runtime. This means that the programmer does not need to declare the type of each variable explicitly. Dynamic typing can make code more concise and easier to write, but it can also make it more difficult to debug and maintain. Some examples of dynamically typed languages are Python, JavaScript, and Ruby.

2. The choice between static and dynamic typing depends on the requirements of the project and the preferences of the developer.

The advantages of using static typing include:

Early detection of errors: The compiler can detect type errors at compile-time, which can save time and reduce the likelihood of errors occurring during runtime.

Improved performance: Static typing can allow the compiler to generate more efficient code.

Better code documentation: The type declarations in the code can make the code more readable and self-documenting.

The disadvantages of using static typing include:

More verbose code: The type declarations can make the code more verbose and harder to read.

Less flexibility: The type system can be restrictive, and it can be difficult to express certain types of data structures.

The advantages of using dynamic typing include:

More flexibility: The lack of type declarations can allow for more flexibility and expressiveness in the code.

Less verbose code: Without type declarations, the code can be more concise and easier to read.

The disadvantages of using dynamic typing include:

Runtime errors: Since type errors are not detected until runtime, it can be more difficult to find and fix errors.

Reduced performance: Dynamic typing can lead to less efficient code, since the interpreter must dynamically determine the type of each variable at runtime.

3. A **strongly typed language** is a language in which the type of each variable and expression is strictly enforced by the compiler. This means that the compiler will not allow you to assign a value of one type to a variable of another type. Some examples of strongly typed languages are Java, C++, and C#.

A **weakly typed language** is a language in which the type of each variable and expression is not strictly enforced by the compiler. This means that the compiler may allow you to assign a value of one type to a variable of another type, even if this is not semantically correct. Some examples of weakly typed languages are Python, JavaScript, and Ruby.

Problem 4

- 1.
- 2.
- 3.
- 4.

Problem 5