

Assignment: Hamming Distance

Description:

The “hamming distance” between two strings of equal length is defined as the number of positions at which the corresponding characters are different. In other words, how many characters would need to be changed in order to make the strings equal.

For instance...

```
Hamming_Distance( text, test ) = 1
Hamming_Distance( text, tell ) = 2
Hamming_Distance( text, told ) = 3
Hamming_Distance( text, bold ) = 4
```

I have attached a file called hamdist.c which is a complete implementation of the hamming distance function. Feel free to compile and test it if you like. Note that strings of any length can be passed in, since it will continue checking until it encounters a null terminator.

Your assignment is to implement this function in assembly. I have included a second file, hamdist.asm, as a skeleton to get you started if needed. **Please utilize the concepts we have been working on: functions, parameters, local variables, and frame pointers.** Specifically, I will be looking for...

1. A separate, standalone function, accessed with call/ret.
2. Two parameters passed from main() to hamdist() in the proper way.
3. An EBP based stack frame complete with prologue and epilogue.
4. Two local variables: the loop counter and the distance counter.
5. Parameters and local variables accessed properly using the frame pointer.

If your function follows those 5 rules and returns correct hamming distances, you will almost certainly receive full credit! Take your time, triple check it, and feel free to ask questions. :)

Deliverables:

- Your assembly code, as a plaintext (.txt) file.
- A screenshot of your code running, proving it works, or showing how far you got.
- The path on the server where it can be found. (run `pwd` to get this)