Write a program that computes the average of two integers. The integers can be hard-coded in the program. The program should correctly give the average of an odd and an even; in particular it should say that “The average is 28 1/2” when this is in fact the case.

The program should contain a function that computes the average; it should take its input values from the stack, and should have a well-written prologue and epilogue.

Some additional useful commands:

- **sal.** The command `sal destination` shifts all of the bits in `destination` left by one bit. This effectively multiplies `destination` by 2. The bits that are emptied are replaced by zero; the bit shifted out is stored in the carry flag.
  
  One also has `sal %cl, destination` which repeats `sal destination` the number of times specified in the CL register.

  The `sal` command needs a suffix to denote the destination size, so we have the three commands `salb`, `salw` and `sall`.

- **shr.** The command `shr destination` shifts all of the bits in `destination` to the right by one bit. This has the effect of dividing an unsigned integer by 2. The bit shifted out is kept in the carry flag, and the bit that is emptied is replaced by zero.

  As before, there are three versions, `shrb`, `shrw`, and `shrl` depending on the size of the destination.

- **sal.** The `shr` command is not appropriate for division of signed integers, because the added bit should be either zero or one depending on the sign of the integer. The command `sar` is the same as `shr`, save that the added bit will be zero if the signed integer is positive, and one if the signed integer is negative, which makes it suitable for the division of signed integers by 2.

  Once again, there are three versions `salb`, `salw` and `sall` depending on the size of the destination.