

## Bit Manipulations

Your program will need code that extracts numbers from the bit fields it reads in. This is the code in the input data.

The bits shall be stored in an **unsigned short**. This gives us the 16 bits we need. Assume each letter below stands for a single bit and that the A is most significant and P is least significant.

ABCD EFGH IJKL MNOP

- The NOP bits are the color, which ranges from 0 to 7.
- The KL bits are the type which ranges from 0 to 3. 0 implies a smile and 1 implies a projectile.
- The EFGH bits are the speed which ranges from 0-15. Note that 0 is not a valid value, even though it can be coded.
- The ABCD bits are the checksum, ranges 0-15.

Note that the ranges given mean the value are unsigned (never negative).

Bits.c

**Assignment Project Exam Help**  
All code that knows the layout of the bits **must** reside in a separate source code file, bits.c. Code that doesn't directly have anything to do with these bit manipulations should not be in this file.

<https://tutorcs.com>  
All numeric constants used to get the numbers out of the fields or packed together **must** be #define symbols inside the .C code file. They won't be in a separate header file because only one file would ever include them.

**WeChat: cstutorcs**  
The functions in bits.c should all have meaningful names and all outside code asks for functions that make sense to it and not for particular bits. You will need the following capabilities:

- A function that returns the color as an integer value
- A function that returns the speed as an integer value
- A Boolean function that tells if the bits are for a smile
- A Boolean function that tells if the bits are for a projectile
- A Boolean function that tells if the bits are valid

## Valid bits

The bits are valid if the number of 1 bits in the low-order twelve bits of the code is the same value as the checksum value contained in the upper four bits and the speed is non-zero.

## Working with the bits

See "Masking.pdf" in the lab 2 area of piazza.