```
#include <stdint.h>
#include <stdio.h>
#include <stdbool.h>
#include <assert.h>
#include <stdlib.h>
typedef struct
{                       // Total: 54 bytes
  uint16_t type;           // Magic identifier: 0x4d42
  uint32_t size;           // File size in bytes
  uint16_t reserved1;      // Not used
  uint16_t reserved2;      // Not used
  uint32_t offset;         // Offset to image data in bytes from beginning of file (54 bytes)
  uint32_t dib_header_size;  // DIB Header size in bytes (40 bytes)
  int32_t width_px;        // Width of the image
  int32_t height_px;       // Height of image
  uint16_t num_planes;     // Number of color planes
  uint16_t bits_per_pixel;   // Bits per pixel
  uint32_t compression;     // Compression type
  uint32_t image_size_bytes; // Image size in bytes
  int32_t x_resolution_ppm;  // Pixels per meter
  int32_t y_resolution_ppm;  // Pixels per meter
  uint32_t num_colors;      // Number of colors
  uint32_t important_colors; // Important colors
} BMPHeader;

typedef struct
{
  BMPHeader header;
  unsigned char *data;
} BMPImage;
```

```
BMPImage *read_bmp(FILE *fp, char **error);
bool write_bmp(BMPImage✶ image, FILE✶ fp, char✶✶ error);
bool check_bmp_header(BMPHeader *bmp_hdr, FILE *fp);
void free_bmp(BMPImage *image);
```