```c
/*
 *  uarray2.h
 *  Isabelle Lai(ilai01), Max Anavian(manavi01)
 *  2/2/2020
 *  Homework 2 iii
 *
 */

#include <stdbool.h>

#define T UArray2_T
typedef struct T *T;

/* Function : UArray2_new
 * Arguments: an int width and height of the new UArray2 set
 * Returns  : a new UArray2
 * Details  : Allocates space for and creates a new UArray2 set of the
 *            given width and height using Hanson's UArray interface
 */
T UArray2_new(int width, int height, int size);

/* Function : UArray2_free
 * Arguments: a pointer to a UArray2 set
 * Returns  : N/A
 * Details  : Frees all the memory associated with UArray2
 *
 */
void UArray2_free(T *uarray2);


/* Function : UArray2_width
 * Arguments: a UArray2
 * Returns  : the int width
 * Details  : Uses Hanson's UArray interface to find the width of the
 *            UArray2
 */
int UArray2_width(T uarray2);

/* Function : UArray2_height
 * Arguments: a UArray2
 * Returns  : the int height
 * Details  : Uses Hanson's UArray interface to find the height of the
 *            UArray2
 */
int UArray2_height(T uarray2);


```

```
/* Function : UArray2_size
 * Arguments: a UArray2
 * Returns  : the int size
 * Details  : Uses Hanson's UArray interface to find the size of the
 *            UArray2
 */
int UArray2_size(T uarray2);

/* Function : UArray2_at
 * Arguments: a UArray2, int col, int row
 * Returns  : A void pointer to the element at current location
 * Details  : Uses Hanson's UArray interface to find the Current
 *            element at col row
 */
void *UArray2_at(T uarray2, int col, int row);

/* Function : UArray2_map_row_major
 * Arguments: UArray2, *function apply(int, int, T, void *, void *,) and a *bool
 * Returns  : N/A
 * Details  : Calls a given apply function on every UArray2 element in the set
 *            traversing row by row
 */

void UArray2_map_row_major(T uarray2,
                           void (*apply)(int, int, T, void *, void *),
                           bool *OK);

/* Function : UArray2_map_col_major
 * Arguments: UArray2, *function apply(int, int, T, void *, void *,) and a *bool
 * Returns  : N/A
 * Details  : Calls a given apply function on every UArray2 element in the set
              traversing col by col
 */
void UArray2_map_col_major(T uarray2,
                           void (*apply)(int, int, T, void *, void *),
                           bool *OK);

#undef T

/*
 *  bit2.h
 *  Isabelle Lai (ilai01), Max Anavian (manavi01)
 *  2/2/2020
 *  Homework 2 iii
 *
 */

#include "bit.h"

#define T Bit2_T
typedef struct T *T;
```

```
93
94  struct T {
95      Bit_T set;
96      int width;
97      int height;
98  };
99
100 /* Function : bit2_new
101  * Arguments: an int width and height of the new bit set
102  * Returns  : The new bit set
103  * Details  : Allocates space for and creates a new bit set of the
104  *            given width and height using Hanson's bit interface
105  */
106 T Bit2_new(int width, int height);
107
108 /* Function : Bit2_free
109  * Arguments: a pointer to a bit set
110  * Returns  : N/A
111  * Details  : Frees all the memory associated with the bit set
112  *
113  */
114 void Bit2_free(T *bitset);
115
116 /* Function : Bit2_width
117  * Arguments: a bit set
118  * Returns  : the int width
119  * Details  : Uses Hanson's bit interface to find the width of the
120  *            bit set
121  */
122 int Bit2_width(T bitset);
123
124 /* Function : Bit2_height
125  * Arguments: a bit set
126  * Returns  : the int height
127  * Details  : Uses Hanson's bit interface to find the height of the
128  *            bit set
129  */
130 int Bit2_height(T bitset);
131
132
133 /* Function : Bit2_put
134  * Arguments: a bit set, int col, int row, int bit
135  * Returns  : the previous bit's value at the given location
136  * Details  : Uses Hanson's bit interface to find the previous bit's
137  *            value and replace it with the given bit value
138  */
139 int Bit2_put(T bitset, int col, int row, int bit);
140
141
142
143
```

```
/* Function : Bit2_get
 * Arguments: a bit set, int col and int row
 * Returns  : the int value of the bit at the given location
 * Details  : Uses Hanson's bit interface to find the current bit's
 *            value at the current col and row.
 */
int Bit2_get(T bitset, int col, int row);

/* Function : Bit2_map_row_major
 * Arguments: a bit set, *function apply(int, int, T, int, void *,) and a *bool
 * Returns  : N/A
 * Details  : Calls a given apply function on every bit in the set traversing
              row by row
 */
void Bit2_map_row_major(T bitset,
                        void (*apply)(int, int, T, int, void *),
                        bool *OK);


/* Function : Bit2_map_col_major
 * Arguments: a bit set, *function apply(int, int, T, int, void *,) and a *bool
 * Returns  : N/A
 * Details  : Calls a given apply function on every bit in the set traversing
              col by col
 */
void Bit2_map_col_major(T bitset,
                        void (*apply)(int, int, T, int, void *),
                        bool *OK);

#undef T

```