```
/* illustration of dynamically allocated 2D array */
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int i; /* general purpose variable used for loop index */
  int j; /* general purpose variable used for loop index */

  int **a;     /* this is the array name */
  int size_x; /* this variable will be used for the first  dimension */
  int size_y; /* this variable will be used for the second dimension */
  int value;
  /* suppose we want an array of int: a[3][3] */
  size_x = 3;
  size_y = 3;
  value = 1;

  /*  allocate storage for an array of pointers */
  a = malloc(size_x * sizeof(int *));

  /* for each pointer, allocate storage for an array of ints */
  for (i = 0; i < size_x; i++) {
    a[i] = malloc(size_y * sizeof(int));
  }
  /* assign an arbitrary value to each element        */
```

```
  for (i = 0; i < size_x; i++) {
    for (j = 0; j < size_y; j++) {
        a[i][j] = value; /* just some unique number for each element */
                value++;
    }
    printf ("\n");
  }

  /* now show the contents that were assigned */
  for (i = 0; i < size_x; i++) {
    for (j = 0; j < size_y; j++) {
      printf("a[%d][%d] = %2d\n", i, j, a[i][j]);
    }
    printf ("\n");
  }

  /* now for each pointer, free its array of ints */
  for (i = 0; i < size_y; i++) {
    free(a[i]);
  }
  /* now free the array of pointers */
  free(a);
  return 0;
}
```