EECS 337 Compiler Design 2009 Fall Semester

C: examples

Example 1. Implement a bag of doubles. The bag has fixed maximum size and it supports only insertion and search.

```
Define the type:
```

Write the following functions:

- double_bag_t *double_bag_init(size_t n); which allocates, initializes, and returns a double bag of size n. The bag should be empty (i.e., contain no element). If an error occurs, the double_bag_init function returns NULL.
- int double_bag_insert(double_bag_t *, double value); which adds a value to the bag. It returns 1 if all went well or 0 in case of error.
- int double_bag_search(double_bag_t *, double value); returns 1 if value is in the bag, and 0 otherwise.
- void double_bag_delete(double_bag_t *); de-allocates the double bag as well as its double array.

Example 2. Write a main program that tests the functions in the double bag.

Example 3. Write a bag of points in the plane. Repeat examples 1 and 2, but store pointers to points in the plane (plane_point_t *) instead of double, where a point in the plane is:

```
typedef struct {
    double x;
    double y;
} plane_point_t;
```

Example 4. Generalize examples 1, 2, and 3 by writing a bag "template": the bag stores pointers to void, and the bag type stores a pointer to a comparison function.