KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
COMPUTER SCIENCE DEPARTMENT

CSC215 - Procedural Programming
Lab Exam 2 - Fall 2016
Monday - Dec. 05, 2016

Consider the following header file **prod.h**

```
#if ! defined PROD
#define PROD

/* struct Product has the following members:
    - serial as integer
    - name as string of maximum length of 24 letters
    - price as floating point number */
typedef struct Product Product;

/* function read_product reads product's information from
    the Keyboard and fills it in the parameter p */
void read_product(Product* p);

/* function print_product prints the information of
    the product p to the screen */
void print_product(Product p);

/* function discounted_copy takes a Product parameter p and
    returns a pointer to a new copy of p after applying a 10%
    discount on the price */
Product* discounted_copy(Product p);

/* function prod_gt_999 takes an array of products parr of size
    s and returns a new array contains discounted copies of
    products with prices greater than 999.99, of size res_s */
Product* prod_gt_999(Product* parr, int s, int* res_s);

/* function save_to_file stores the size of array parr in the file
    fn, then stores all products of parr in the file and returns
    1 if succeeds, and 0 otherwise */
int save_to_file(char* fn, Product* parr, int size);

#endif
```

1. Write a C program **"prod.c"** that:
   a. Defines the type `struct Product`   1 point
   b. Implements the function `read_product`   1.5 point
   c. Implements the function `print_product`   1 point
   d. Implements the function `discounted_copy`   1.5 point
   e. Implements the function `prod_gt_999`   1.5 point
   f. Implements the function `save_to_file`   1.5 point
2. Bonus: write a main function that:
   a. Asks the user to enter number of products `num`
   b. Allocated enough memory for array `prod_arr` to store `num` products
   c. Reads the information of the `num` products from the keyboard
   d. Stores products with `price` > 999.99 in array `prod_999` of size `num_999`
   e. Prints the information of all products in `prod_999` to the screen
   f. Stores the number of products of `prod_999` and their information in the file
      `"prod-999.dat"`

Sample run:

```
Enter number of products: 4
Enter the serial: 22
Enter the name: Toshiba TV
Enter the price: 5700
Enter the serial: 25
Enter the name: iPod G6
Enter the price: 890
Enter the serial: 17
Enter the name: LG Freezer
Enter the price: 4200
Enter the serial: 29
Enter the name: Samsung Vacuum Cleaner
Enter the price: 397

products with prices more than 999.99:
======================
Serial: 22
Name: Toshiba TV
Price: 5130.00
--------------
Serial: 25
Name: LG Freezer
Price: 3780.00
--------------
```