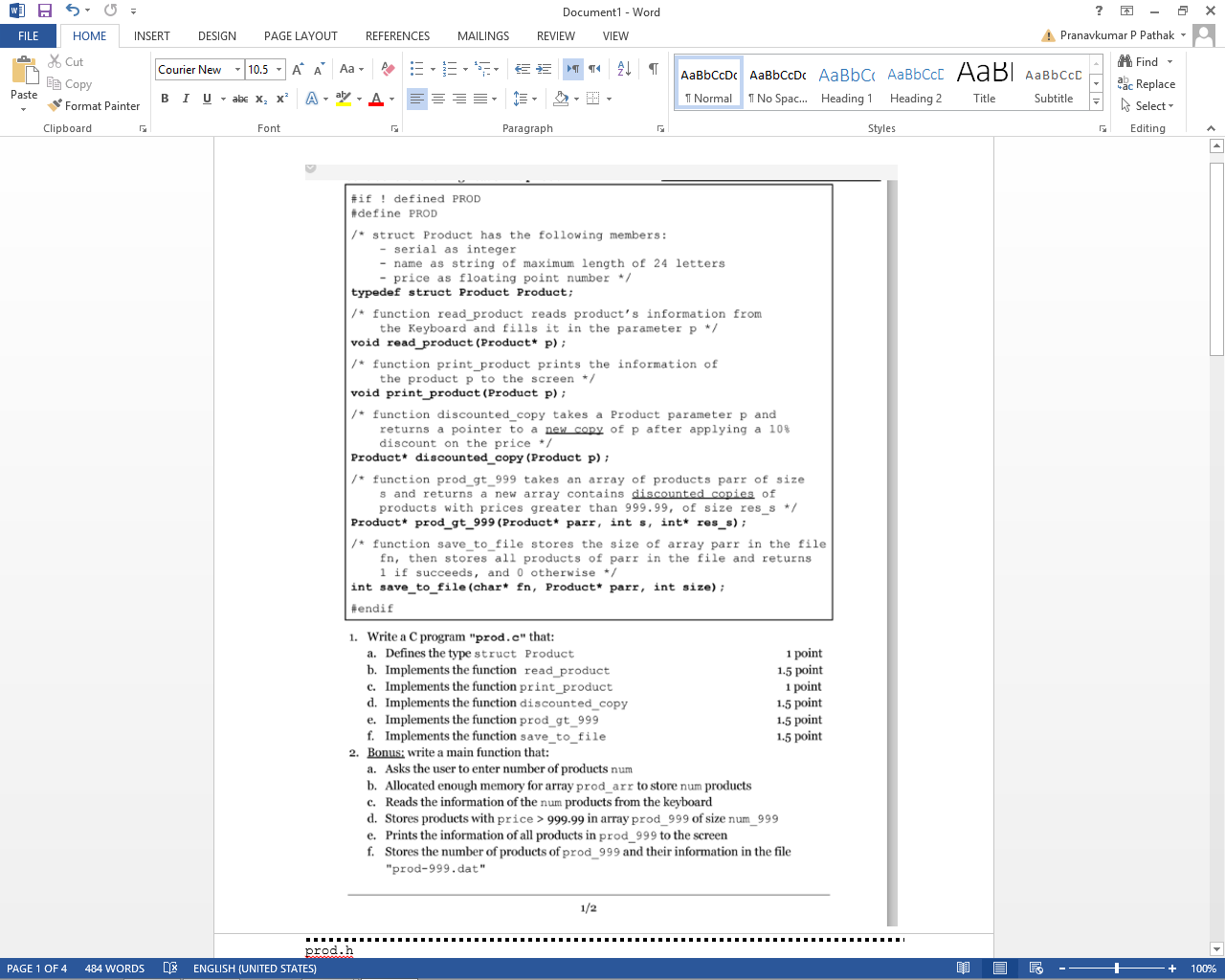
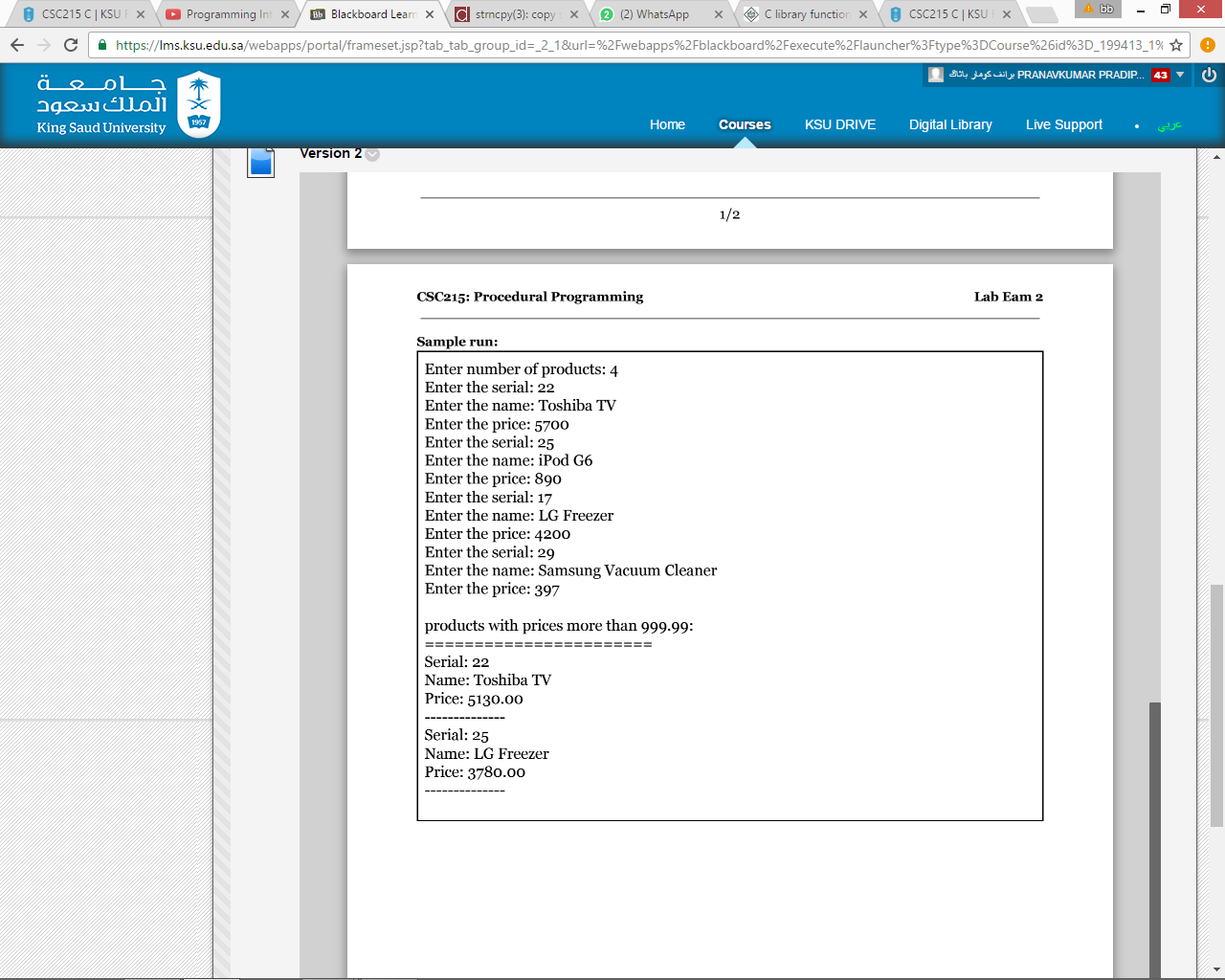
LAB Exam 2.





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

prod.c

#include <stdio.h>

#include <stdlib.h>

#include "prod.h"

struct Product{

int serial;

char name[25];

float price;

};

void read\_product(Product\* p){

char c;

printf("Enter the serial number: ");

scanf("%d", &(p->serial));

printf("Enter the name: ");

while ((c=getchar())=='\n');

ungetc(c, stdin);

gets(p->name);

printf("Enter the price: ");

scanf("%f", &(p->price));

}

void print\_product(Product p){

printf("Serial number: %d\n", p.serial);

printf("Name: %s\n", p.name);

printf("Price: %.2f\n", p.price);

printf("--------------\n");

}

Product\* discounted\_copy(Product p){

Product\* result;

result = (Product\*)malloc(sizeof(Product));

result->serial = p.serial;

sprintf(result->name, "%s", p.name);

result->price = p.price \* 0.9;

return result;

}

Product\* prod\_gt\_999(Product\* parr, int s, int\* res\_s){

int i, j;

Product\* result;

\*res\_s = 0;

for (i=0; i< s; i++)

if (parr[i].price > 999.99) (\*res\_s)++;

result = (Product\*)malloc((\*res\_s)\*sizeof(Product));

for(i=0,j=0; i<s; i++)

if (parr[i].price > 999.99)

result[j++] = \*(discounted\_copy(parr[i]));

return result;

}

int save\_to\_file(char\* fn, Product\* parr, int size){

int i;

FILE\* fp = fopen(fn, "w");

fwrite(&size, sizeof(int), 1, fp);

for (i=0; i<size; i++)

fwrite(&(parr[i]), sizeof(Product), 1, fp);

fclose(fp);

return size;

}

int main(void) {

int num, num\_999, i;

Product \*prod\_arr, \*prod\_999;

/\* ask the user to enter number of products num \*/

printf("Enter number of products:");

scanf("%d", &num);

/\* allocate enough memory for array prod\_arr to store num products \*/

prod\_arr = (Product\*)malloc(num\*sizeof(Product));

/\* read the information of the num products from the keyboard \*/

for (i=0; i < num; i++)

read\_product(&(prod\_arr[i]));

/\* store products with price > 999.99 in the array prod\_999

of size num\_999 \*/

prod\_999 = prod\_gt\_999(prod\_arr, num, &num\_999);

/\* print the information of all products in prod\_999 to the screen \*/

printf("\nProducts with price more than 999.99\n");

printf("============================================\n");

for (i=0; i < num\_999; i++)

print\_product(prod\_999[i]);

/\* store the number of products in prod\_999 and their information in

the file "prod-999.dat"\*/

save\_to\_file("prod-999.dat", prod\_999, num\_999);

return 0;

}

data.txt

4

22

Toshiba TV

5700

17

iPod G6

890

25

LG Freezer

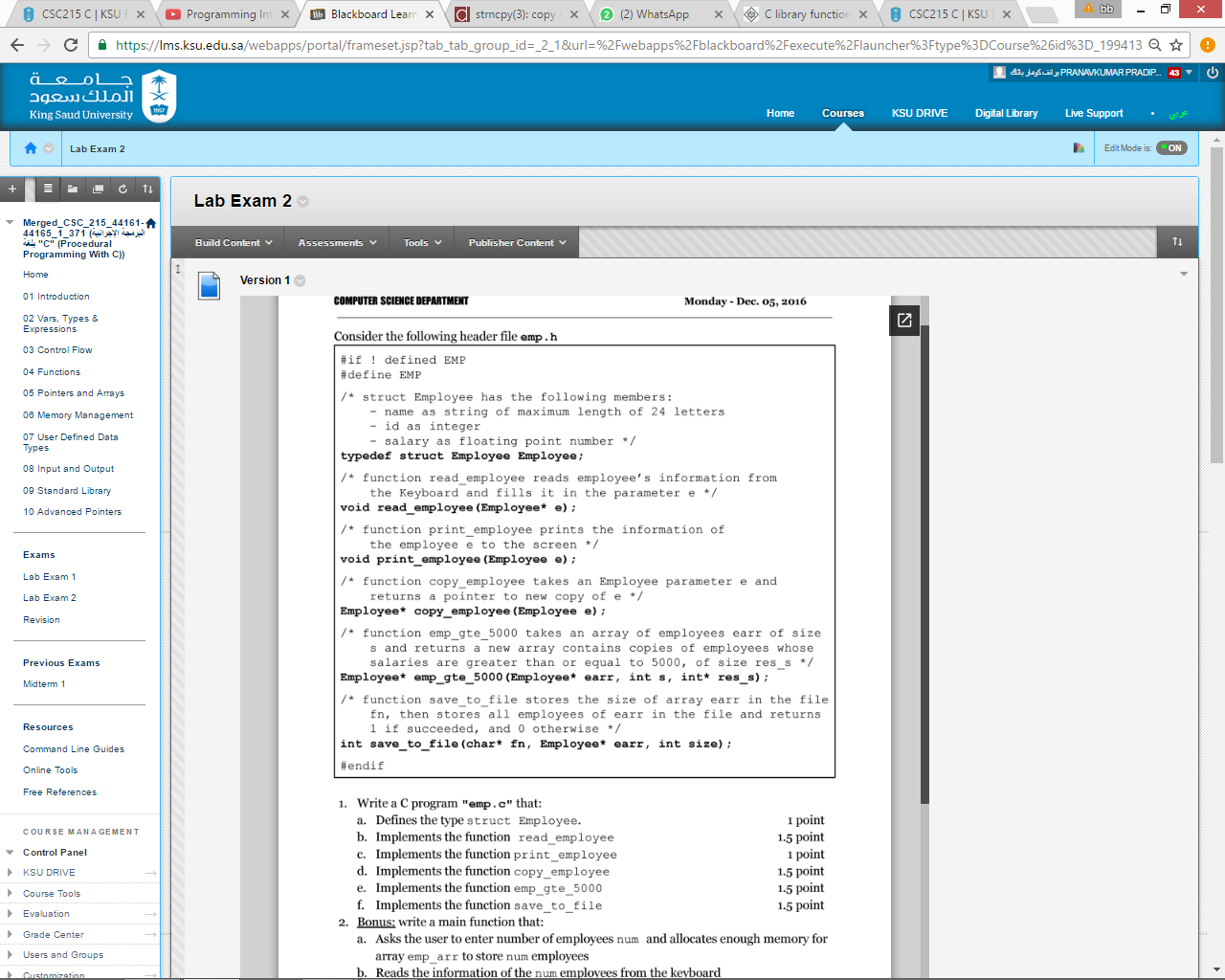
4200

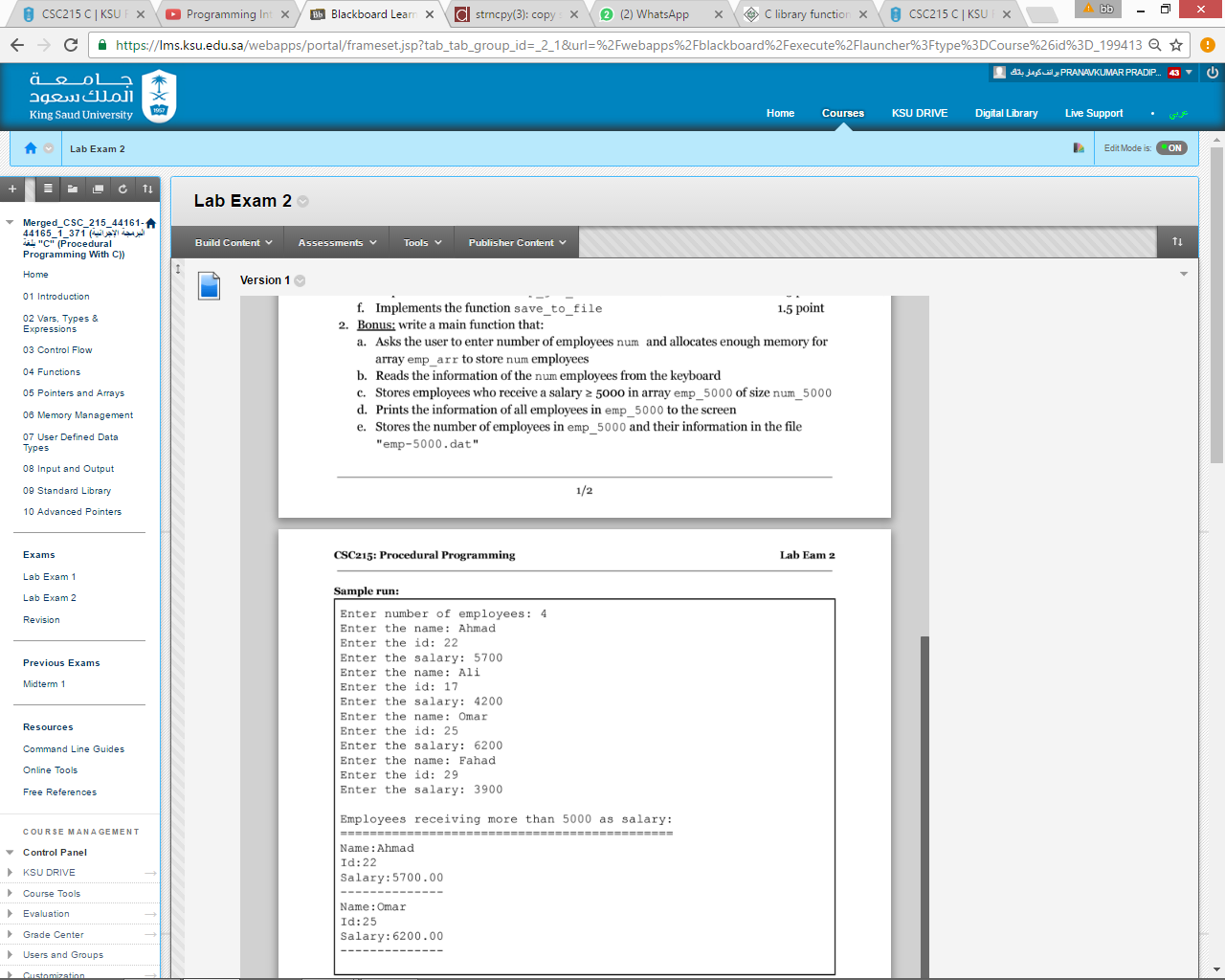
29

Samsung Vacuum Cleaner

397

\*=+\*+\*+\*+\*+\*+\*+\*+\*+\*+\*\*+\*+\*+\*+\*+\*+\*+\*\*+\*+\*++++\*\*+\*+\*\*\*\*\*\*\*\*\*\*\*+\*+\*+





// emp.h

#if ! defined EMP

#define EMP

typedef struct Employee Employee;

void read\_employee(Employee\* e);

void print\_employee(Employee e);

Employee\* copy\_employee(Employee e);

Employee\* emp\_gte\_5000(Employee\* earr, int size, int\* res\_size);

int save\_to\_file(char\* fn, Employee\* earr, int size);

#endif

#include <stdio.h>

#include <stdlib.h>

#include "emp.h"

struct Employee{

char name[25];

int id;

float salary;

};

void read\_employee(Employee\* e){

char c;

printf("Enter the name: ");

while ((c=getchar())=='\n');

ungetc(c, stdin);

gets(e->name);

printf("Enter the id: ");

scanf("%d", &(e->id));

printf("Enter the salary: ");

scanf("%f", &(e->salary));

}

void print\_employee(Employee e){

printf("Name:%s\n", e.name);

printf("Id:%d\n", e.id);

printf("Salary:%.2f\n", e.salary);

printf("--------------\n");

}

Employee\* copy\_employee(Employee e){

Employee\* result;

result = (Employee\*)malloc(sizeof(Employee));

\*result = e;

/\* you can also do this:

sprintf(result->name, "%s", e.name);

result->id = e.id;

result->salary = e.salary; \*/

return result;

}

Employee\* emp\_sal\_gte\_5000(Employee\* earr, int size, int\* res\_size){

int i, j;

Employee\* result;

\*res\_size = 0;

for (i=0; i< size; i++)

if (earr[i].salary > 5000) (\*res\_size)++;

result = (Employee\*)malloc((\*res\_size)\*sizeof(Employee));

for(i=0,j=0; i<size; i++)

if (earr[i].salary >= 5000)

result[j++] = \*(copy\_employee(earr[i]));

return result;

}

int save\_to\_file(char\* fn, Employee\* earr, int size){

int i;

FILE\* fp = fopen(fn, "w");

fwrite(&size, sizeof(int), 1, fp);

for (i=0; i<size; i++)

fwrite(&(earr[i]), sizeof(Employee), 1, fp);

fclose(fp);

return size;

}

int main(void) {

int num, num\_5000, i;

Employee \*emp\_arr, \*emp\_5000;

/\* ask the user to enter number of employees num \*/

printf("Enter number of employees:");

scanf("%d", &num);

/\* allocate enough memory for array emp\_arr to store num employees \*/

emp\_arr = (Employee\*)malloc(num\*sizeof(Employee));

/\* read the information of the num employees from the keyboard \*/

for (i=0; i < num; i++)

read\_employee(&(emp\_arr[i]));

/\* store employees who receive a salary = 5000 in the array emp\_5000

of size num\_5000 \*/

emp\_5000 = emp\_sal\_gte\_5000(emp\_arr, num, &num\_5000);

/\* print the information of all employees in emp\_5000 to the screen \*/

printf("\nEmployees receiving more than 5000 as salary\n");

printf("============================================\n");

for (i=0; i < num\_5000; i++)

print\_employee(emp\_5000[i]);

/\* store the number of employees in emp\_5000 and their information in

the file "emp-5000.dat"\*/

save\_to\_file("emp-5000.dat", emp\_5000, num\_5000);

return 0;

}

// data.txt

4

Ahmad

22

5700

Ali

17

4500

Omar

25

6200

Fahad

29

3900