CS 1713 Introduction to Programming II

Spring 2014 – Midterm 2 -- April 24, 2014 You have 75 min. Good luck. You can use the 2-page C reference card posted in the class web page.

Name:....

Score:/100

Background Survey (5pt bonus credit)

A. Please complete the below table for the computer-programming-related courses that you have taken before Spring 2014.

Programming courses	TAKEN AT UTSA, (Yes / No)	If you have taken equivalent courses from a different school, please give the school name and the programming language used.
CS 1063 Intro to Comp Prog I (in Java)		
CS 1713 Intro to Comp Prog II (in C)		(Say NO if this is the first time you are taking it)

B. How would you evaluate your programming skills and background? (circle one)

5: excellent, developed some programs, received A in all Intro programming courses.

4: good, developed a few programs, received B in Intro programming courses.

3: fair, developed very simple programs, received C in Intro programming courses

2: beginner, have not take any programming course, never wrote any program.

1: What is "programming"?

C. Do you think you had sufficient background to take cs1713?

^C Definitely. ^C Somewhat. ^C I don't know. ^C Not really. ^C Definitel
--

D. If you feel that you had (or did not have) sufficient background, what should be done (both on your part and in the curriculum) so that you can be successful in cs1713? (Please comment)

	— outp	out.	
<pre>#include <stdio.h> main()</stdio.h></pre>			MEMORY
{	name	Add ress	Content/Value
int x, y, z[3]={8,5,4}, *p1, *p2;	x	12	
p1 = &x	У	16	
*p1 = 9;	z[0]	20	
	z[1]	24	
p2 = z + 2;	z[2]	28	
y = *p2;	p1	32	
(p2-1) = 7;	p2	36	
*++p2 = 3;			
printf("%d %d %d %d \n",		100	
x, y, z[0], z[1], z[2]);	a	104	
y = f1(&x, *p1, p2-1, &p2);	b	108	
	С	112	
princi ("%d %d %d %d %d \n", x, y, z[0], p1, p2);	d	116	
printf("%d %d \n", *p1, *p2):	X	120	
	У	124	
}		120	
<pre>int f1(int *a, int b, int *c, int **d)</pre>		Ͳ	
{ int x=5, y=7;		101.	
*a = y % x;			
*d = c-1;			
**d = *a + b;			
return **d + *a; }			

Name:
1. (20 pt) Trace the following program, show how values change in the memory, and give the

Name:.....

2. (20pt) You are asked to implement the following function using pointers and pointer arithmetic int substrindex (char *str, char *substr); which returns -1 if substris not in str; otherwise, it returns the index value for the first appearance of substr in str. You can assume that both str and substr are NULL terminated strings. For example,

```
substrindex("abcabcdef", "abc") returns 0,
substrindex("abcabcdef", "abcde") returns 3,
substrindex("abc", "xy") returns -1
int substrindex(char *str, char *substr)
{ /* FOR FULL CREDIT DON'T USE ARRAY NOTATION!
* IF YOU SOLVE IT USING ARRAY NOTAION, YOUR MAX WILL BE 10Pt
*/
```

Name:....

3. (20pt) Develop a function char *my_concat(char *s1, char *s2); that concatenates two strings by joining them end to end. This function should dynamically allocate the memory for the new string and copy the original strings end to end into the new string.

For example, after calling s = my_concat("ABC", "DE"); s should be pointing to a new string "ABCDE".

If needed, you can use the standard string functions from <string.h> library.

char *my_concat(char *s1, char *s2) {

Name:.....

4. (20pt) Suppose we have the following structure and variable declarations.

```
typedef struct {
    int x;
    int y;
} myDataT;
myDataT a, *b, **c;
```

```
a) (2pt) Set all the fields of variable a to 0.
```

b) (3pt) Let b = &a; Then through pointer b set all the fields to 1.

c) (5pt) Allocate an array of 50 pointers to myDataT and save the address of that array into c.

d) (5pt) Suppose the memory allocation in part (c) is successful and c is now the array of 50 pointers to myDataT. Now, dynamically allocate an actual myDataT structure for each element of the array c and save its address in c[i].

e) (5pt) Free all the memory that you allocated in part (c) and part (d).

Name:.....

5. (20pt) We maintain two files to store the IDs and homework grades of the students in two sections. At the end of the semester, I need to merge these two files into a single file. Each student has **nine** homework grades after his/her ID, but the number of students in a file could be anything! Also the IDs in the input files are already *sorted*.

Now you are asked to write a program that takes three file names as *command line arguments*, and **merges** the data in the first two files into the third file while keeping the IDs **sorted**.

2	3	5	2	5	3	5	2	5	3			14	3	5	5 2	5	3	5	2	5	3
8 5	5 3	2 5	5 2	3 5	3	5 2	2 5	5 3	3 5			17 22	5 3	5	2 5	3 5	3	5 2	2 5	5 3	3 5
												30	5	3	3 3	5	2	5	3	5	2
				5	5.41	Ът		as	101		5										
pr	og	se	ect	cio	n1	.t>	кt	se	ect	ior	n1.	txt	se	ct:	ion-	-al	1.	tx	t		
pr	og Id n	se rin	ect	io 1f fl	n1 re f	.tx	kt Owi	s∈ ina	ect lin	ior	n1.	txt	se	ct:	ion-	-al	.1.	tx	t		
pr Iou	og ld p	se orin	ect tou	io ut tl	n1 ne f	.t> foll	kt owi	se ing	ect lin	ior es i	n1. nto	txt sec	se tic	ct: on-	ion- all	-al .t:	.l. xt	tx	t		
pr ou	og Id p	se orin	ect t oi	io ut tl	n1 ne f	.t≥ folle 12	kt owi	se ing 3	lin	ior es i 2	n1. nto	txt sec 3 5	se tic 2	ct: on- 5	ion- all 3	-al .t:	.l. xt	txi	t		
pr ou	og Id p	s. orin	ect t oi	io ut tl	n1 ne f	.tz folle 12 14	kt O W i	se ing 3 3	lin 5	ior esi 2 2	nto 5	txt sec 3 5 3 5	se tic 2 2	ct: on- 5 5	ion- all 3 3	-al .t:	.l. xt	txi	t		
pr ou	rog Id p	se orin	ect t oı	io ut tl	n1 ne f	.tz follo 12 14 17	kt owi	se ing 3 3 5	lin 5 2	ior es i 2 2 5	nto 5 (3 (txt sec 3 5 3 5 3 5 3 5	se tic 2 2 2	ct: on- 5 5 5	ion- all 3 3 3	-al .t:	.l. xt	txi	t		
pr ou	og Id p	se orin	ect	io ut tl	n1 ne f	.t≥ folle 12 14 17 18	kt owi	se ing 3 3 5 5	2 2	ior es i 2 2 5 5	nto 5 (3 (3 (txt sec 3 5 3 5 3 5 3 5 3 5	se tic 2 2 2 2	ct: on- 5 5 5 5	ion- all 3 3 3 3	-al .t:	.l. xt	tx	t		
pr ou	og Id p	s. orin	ect	io ut tl	n1 ne f	.t≥ follo 12 14 17 18 22	kt owi	se ing 3 5 5 3	5 2 5 5	ior es i 2 2 5 5 2	nto 5 (5 (3 (5 (txt sec 3 5 3 5 3 5 3 5 3 5 3 2	se tic 2 2 2 2 5	ct: on- 5 5 5 3	ion- all 3 3 3 3 5	-al .t:	.l. xt	txi	t		
pr IOU	og Id p	s. orin	ect	io ut tl	n1 ne f	.t≥ folle 12 14 17 18 22 25	st owi	se ing 3 3 5 5 3 3	2 5 5 5 5 5	ior es i 2 2 5 5 2 2	nto 5 2 3 2 5 2 5 2	txt sec 3 5 3 5 3 5 3 5 3 5 3 2 3 2	se tic 2 2 2 5 5	ct: on- 5 5 5 3 3	ion- all 3 3 3 5 5	-al .t:	.l. xt	tx	t		

Implement your program by completing the following program also use the next page.

/* Name:.....