

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?

☐ Definitely. ☐ Somewhat. ☐ I don't know. ☐ Not really. ☐ Definitely not.

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)

Name: _____

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

```
#include <stdio.h>
main()
{
    int x, y, z[3]={8,5,4}, *p1, *p2;

    p1 = &x;

    *p1 = 9;

    p2 = z + 2;

    y = *p2--;

    *(p2-1) = 7;

    *++p2 = 3;

    printf("%d  %d  %d  %d  %d \n",
           x, y, z[0], z[1], z[2]);

    y = f1(&x, *p1, p2-1, &p2);

    printf("%d  %d  %d  %d  %d \n",
           x, y, z[0], p1, p2 );

    printf("%d  %d \n", *p1, *p2);
}

int f1(int *a, int b, int *c, int **d)
{
    int x=5, y=7;

    *a = y % x;

    *d = c-1;

    **d = *a + b;

    return **d + *a;
}
```

MEMORY		
name	Add ress	Content/Value
x	12	? 9 2
y	16	? 4 13
z[0]	20	8 7 11
z[1]	24	5
z[2]	28	4 3
p1	32	? 12
p2	36	? 28 24 28 20
...		
	100	
a	104	12
b	108	9
c	112	24
d	116	36
x	120	5
y	124	7
	128	

OUTPUT:

```
9    4    7    5    3
2    13   11   @12  @20
2    11
```

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 `substr` is 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 NOTATION, YOUR MAX WILL BE 10Pt
  */

  char *s1, *s2;
  int i=0;

  while(*str){
    if (*str == *substr) {
      s1=str + 1;
      s2=substr + 1;
      while(*s1 && *s2 && *s1 == *s2){
        s1++;
        s2++;
      }
      if(*s2=='\0')
        return i;
    }
    str++;
    i++;
  }

  return -1;
}
```

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)
{
    char *s;
    int len1, len2;

    if (s1 == NULL || s2 == NULL) {
        printf("NULL string passed to my_concat\n");
        return NULL;
    }

    len1 = strlen(s1);
    len2 = strlen(s2);

    s = malloc(len1 + len2 + 1 );
    if (s==NULL) return NULL;

    strcpy(s, s1);
    strcpy(s + len1, s2);    /* or strcat(s,s2); */

    return (s);
}
```

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.

```
a.x = 0;  
a.y = 0;
```

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

```
b->x = 1;  
b->y = 1;
```

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

```
c = (myDataT **) malloc( 50 * sizeof(myDataT *) );  
if(c==NULL) { printf("Error"); exit(-1); }
```

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]`.

```
for(int i=0; i < 50; i++){  
    c[i] = (myDataT *) malloc( sizeof(myDataT) );  
    if(c[i]==NULL) {printf("Error"); exit(-1); }  
}
```

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

```
for(int i = 0; i<50; i++)  
    free(c[i]);  
  
free(c);
```

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**.

Here is an **example**.

Suppose we have the following two files: section1.txt and section2.txt

12	3	5	2	5	3	5	2	5	3
18	5	2	5	3	3	5	2	5	3
25	3	5	2	5	3	2	5	3	5

14	3	5	2	5	3	5	2	5	3
17	5	2	5	3	3	5	2	5	3
22	3	5	2	5	3	2	5	3	5
30	5	3	3	5	2	5	3	5	2

When we execute your prog as follows

```
> prog section1.txt section1.txt section-all.txt
```

it should print out the following lines into section-all.txt

12	3	5	2	5	3	5	2	5	3
14	3	5	2	5	3	5	2	5	3
17	5	2	5	3	3	5	2	5	3
18	5	2	5	3	3	5	2	5	3
22	3	5	2	5	3	2	5	3	5
25	3	5	2	5	3	2	5	3	5
30	5	3	3	5	2	5	3	5	2

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

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    FILE *infp1, *infp2, *outfp;
    int id1, id2, end1=0, end2=0;

    ..... /* if you need declare more var */
    if (argc!=4) {
        printf("you need to give three file names\n"); return -1;
    }
    if ( (infp1=fopen(argv[1], ..... "r".....)) == NULL ||
        (infp2=fopen(argv[2], ..... "r".....)) == NULL ||
        (outfp=fopen(argv[3], ..... "w".....)) == NULL ) {
        printf("some files cannot be opened\n"); return -1;
    }
}
```

```

/* Name:..... */

if (fscanf(infp1, "%d", &id1)!=1) end1 = 1;
if (fscanf(infp2, "%d", &id2)!=1) end2 = 1;

while( !end1 || !end2 ){
    if (end2 || id1 < id2) {
        fprintf(outfp, "%d ", id1);
        copy_grades(infp1, outfp);
        if (fscanf(infp1, "%d", &id1)!=1) end1 = 1;
    } else {
        fprintf(outfp, "%d ", id2);
        copy_grades(infp2, outfp);
        if (fscanf(infp2, "%d", &id2)!=1) end2 = 1;
    }
}
fclose(infp1);
fclose(infp2);
fclose(outfp);
}

void copy_grades(FILE *infp, FILE *outfp)
{
    int i, hw;
    for(i=1; i<=9; i++) {
        fscanf(infp, "%d", &hw);
        fprintf(outfp, "%d ", hw);
    }
    fprintf(outfp, "\n");
}

```