

# ICS 103 Final Exam

For semester 102

June 12, 2011

<b>Name</b>	
<b>ID</b>	
<b>Section Instructor</b>	
<b>Section Number or class time</b>	

<b>Question 1</b>	<b>24</b>	
<b>Question 2</b>	<b>36</b>	
<b>Question 3</b>	<b>40</b>	
<b>Total</b>	<b>100</b>	

**Question 1: Find the output of each code block of the following**

**24pts = 4 + 4 + 3 + 4 + 5 + 4**

<pre>int A[ 8] , n=8; for(i=0; i &lt; n; i++)     if (i%2==0) A[ i] = 5*i;     else A[ n-i] = -5*i; for(i=0; i &lt; n; i++)     printf("%d\n",A[ i] );</pre>	<p>0 -35 10 -25 20 -15 30 -5 0.5 for each output -2 if exact numbers were found but not in correct locations. -3 if it is partially correct.</p>
<pre>int x[] = { 4,6} ; int y[] = { 10,4} ; int i,j; for(i=0 ; i&lt;2 ; i++) {     for(j=0 ; j&lt;2 ; j++)         printf("%d ", x[ i] * y[ j] );     printf("\n"); }</pre>	<p>40 16 60 24 +1 mark for each output -1 for incorrect formatting</p>
<pre>char names[][ 80] = { "Good Morning", "Hello", "Hi"} ; int i; for(i=0; i&lt;2; i++) printf("%d %c\n",strlen(names[ i] ), names[ i][ 2] );</pre>	<p>12 o 5 1 +1.5 for each line -1 if student got the the character at names[i][2] wrong</p>
<pre>int A[ 8] = { 2,1,3,6,5} ; int j = 7,i; for(i=0 ; i&lt;=7 ; i++)     printf("%d ", A[ j] - A[ i] );</pre>	<p>-2 -1 -3 -6 -5 0 0 0 0.5 for each output -1 for missing the zeros</p>
<pre>char str[] = "This is interesting"; char delims[] = "i"; char *token;  token = strtok( str, delims ); while ( token != NULL ) {     puts(token);     token = strtok( NULL, delims ); }</pre>	<p>Th s s n ng +1 for each line. -1.5 for including the delimiter in the output -1.5 for NOT writing the results on separate lines.</p>
<pre>void fun(int *a, int *b, int c, int d) {     int x;     x=*a; *a = c; d = *b; c = x;</pre>	<p>20 15 20 25 +1 points for each input</p>

```

}
int main(void)
{
    int p=10,q=15,r=20, s=25;
    fun(&p, &q, r, s);
    printf("%d %d %d %d", p, q, r, s);
    printf("\n");}

```

**Question 2 (36 pts = 3 pts each): Choose the correct answer of the following multiple choice questions**

**Answer Key -- A**

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
E	C	D	C	B	E	C	A	D	D	A	B

**Answer Key -- B**

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
D	B	A	D	C	A	D	B	C	E	A	B

**Question 3 (40pts):**

**1. (10pts) Complete the following recursive binary search function so that it returns the index of the array element when found or -1 if not found.**

```

int BinarySearch(double data[],double key,int start, int end, int
size){
    if( start > end ) //2 pts
        return -1; //the key was not found in the array

    int middle = (start + end) / 2 ; //2pts

    if( array[middle] == key )
        return middle; //2pts
    else
        if(array[middle] < key)
            return BinarySearch(data, key, middle + 1, end, size); //2pts
        else if(array[middle] > key)
            return BinarySearch(data, key, start, middle - 1, size
//2pts
);
}
//if student got mixed up between the two recursive calls : for
example looked in the other half of the array then -2pts. If the

```

student got ONE parameter in the call wrong, then -1pts. For example  
BS(data, key, middle, end, size) □ -1pts

**2. (6pts) Complete the program function *FixMe()* given below. Its purpose is to change any digit found in its string parameter to '\$'. For example, if the string parameter has the value "ICS 103 IS FUN", then after executing the function it will have the value "ICS \$\$\$ IS FUN"**

**In addition, answer why we should remove the return statement below**

```
void FixMe(char[] A){
    int i;
    for(i=0; i<strlen(A) ;i++) //also A[i] != '\0' 2pts
        if(isdigit(A[i])) //also A[i] >= '0' && A[i] <= '9' works 2pts
            A[i] = '$';

    return A;// This statement should be removed because A
```

**Acceptable Answers :** 2pts

Because arrays are passed by reference.

Because arrays are pointer variables.

**3. (6pts ) Complete the following function so that it returns the first index within the array of strings "sentences", that contain the string "key". If the "key" was not found in any of the "sentences", -1 should be returned. For example, the word "Today is Sunday" contains "Sun".**

```
int find_key(char sentences[][ LENGTH] , char key[ LENGTH] , int size)
{
    int i,j;
    for(i=0 ; i<size ; i++)
    {
        if( strstr(sentences[ i] ,key) != NULL )//2pts
        {
            return -1;//2pts
        }
    }
    return -1 ;//2pts
}
```

**4. (6pts) Complete the following function so that it swaps its two arguments. The two arguments for this function are strings.**

```
void swap_strings(char A[] , char B[] ) //1.5pts
{
    char temp[ LENGTH] ; //also including length as a passed parameter
    is okay. //1.5pts

    strcpy(temp, A); //1.5pts

    strcpy(A, B); //1.5pts
```

```
    strcpy(B, temp); //1.5pts  
}
```

**5. (6 pts) Complete the following function so that it finds the sum of each column of its argument (2D array) and returns the result as a 1-D array. Assume the type is double.**

```
void sum_columns(int A[][COLS] , int sum_cols[], int rows, int cols) {  
    int i,j;  
  
    for(i=0; i < cols; i++) //2pts  
    {  
  
        for(j=0 ; j<rows ; j++) //2pts  
        {  
            sum_cols[ i ] += A[ j][ i ]; //2pts  
        }  
    }  
}
```

**6. (6pts) Complete the following code fragment so that it will find how many capital letters are in the list of strings shown in the initialization.**

```
int i,j,count;

char text[ 4][ 80] = { "THis is ics 103","Final EXam","TErm 102","Good
Luck"};
count = 0;//1pt

for(i=0; i<4 ;i++) { //1pt
    for(j=0; j<strlen(text[ i] ) ;j++) //1pt
        if (isupper(text[ i][ j] ))//1pt
            count++; //1pt
}
printf("Count = %d",count); //print the result (1pt)
```