CS101 Fall 2013
Final Exam (120 minutes)
Monday, December 16th, 2013

Name: ___________________________ Section: ____________________

READ and complete the following:

- Bubble your Scantron only with a No. 2 pencil.
- On your Scantron (shown in the figure below), bubble:

  1. Your Name
  2. Your NetID
  3. Form letter "A"
  4. Bubble the corresponding 3-digit code (shown below) for your lab section on your Scantron.

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00-10:50</td>
<td>101</td>
<td>104</td>
<td>108</td>
<td>111</td>
</tr>
<tr>
<td>11:00-12:50</td>
<td>102</td>
<td>105</td>
<td>109</td>
<td>112</td>
</tr>
<tr>
<td>1:00-2:50</td>
<td></td>
<td>106</td>
<td></td>
<td>113</td>
</tr>
<tr>
<td>3:00-4:50</td>
<td>103</td>
<td>107</td>
<td>110</td>
<td>114</td>
</tr>
</tbody>
</table>

- No electronic devices, books, notes, or cheat sheets are allowed while taking this exam.
- Please fill in the most correct answer on the provided Scantron sheet.
- We will not answer any questions during the exam.
- Each question has only ONE correct answer.
- You must stop writing when time is called by the proctors.
  **No extra time will be given after the exam ends to fill in bubble sheets with answers.**
- Hand in both these exam pages and the Scantron.
- DO NOT turn this page UNTIL the proctor instructs you to.
STOP!!! Please ensure that your NETID is bubbled in correctly on your scantron sheet. If it is not, you will AUTOMATICALLY be deducted 5 points from your final score.
1. The following C program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

void increase1(int x)
{
    x = x + 1;
}

void increase2(int *x)
{
    *x = *x + 1;
}

void main(void)
{
    int x = 3;

    increase1(x);
    printf("%i, ",x);

    increase2(&x);
    printf("%i",x);
}
```

(a) 3, 3
(b) 3, 4
(c) 4, 3
(d) 4, 4
2. The following C program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

int global_var = 0;

int fun(int x)
{
    global_var = global_var + 1;
    return x + global_var;
}

void main(void)
{
    int x = 3;
    int y;
    global_var = global_var + 1;
    y = fun(x+global_var);
    printf("%i",y);
}
```

(a) 4
(b) 5
(c) 6
(d) 7
3. Which of the following C programs gives you a compile **ERROR**?

(a) 
```c
#include <stdio.h>

int fun1(int x, int y, int z)
{
    return x+y;
}

void main(void)
{
    int y=0;
    printf("%i",fun1(y, y, y));
}
```

(b) 
```c
#include <stdio.h>

int fun2(int x)
{
    int y=0;
    return x+y;
}

void main(void)
{
    int y=0;
    printf("%i",fun2(y));
}
```

(c) 
```c
#include <stdio.h>

int fun3(int x)
{
    return x+y;
}

void main(void)
{
    int y=0;
    printf("%i",fun3(y));
}
```

(d) 
```c
#include <stdio.h>

int fun4(int x)
{
    int z=0;
    return x+1;
}

void main(void)
{
    int y=0;
    printf("%i",fun4(y));
}
```
4. The following C program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

int x;

void func(int y)
{
    x = y;
    printf("%i ",x);
}

void main(void)
{
    int x = 1;
    func(2);
    printf("%i",x);
}
```

(a) 2 1  
(b) 2 2  
(c) 1 2  
(d) 1 1

5. The following program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

void main(void)
{
    int a[] = {20, 30, 50, 10, 40};
    int *c, *d, j;
    int *b = a;
    c = b;
    printf("%i %i ",(b)++,*(++c));
    d = c;
    for(j=0; j<3; j++)
    {
        printf("%i ", (*d)++);
    }
}
```

(a) 20 30 30 50 10  
(b) 20 20 30 50 10  
(c) 20 30 40 50 10  
(d) 20 30 30 31 32
6. The following program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

void main(void)
{
    char str[20] = " World!";    /* There is a space before the W in World! */
    char *p=str;
    *p='W';
    *(p+4) = '\0';
    printf("%s\n", str);
}
```

(a) W Worl
(b) W Wor
(c) World
(d) Worl

7. Replace the blank line with the correct code in order display the value 6 when the program is compiled and run without errors.

```c
#include <stdio.h>

void main(void)
{
    int a=2, b=6, c=4;
    int *arr[]={&a, &b, &c};

    __________________________(what code goes here?)_____________________________
}
```

(a) printf("%i \n", arr[0]);
(b) printf("%i \n", arr[1]);
(c) printf("%i \n", *arr[0]);
(d) printf("%i \n", *arr[1]);
8. The following C program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

void main(void)
{
    int x[2] = {1, 2};
    int *ptrX;

    ptrX = x;
    ++(*ptrX);
    ptrX++;
    x[1]--;
    --(*ptrX);

    printf("(%i, %i)\n", x[0], x[1]);
}

(a) (0, 0)
(b) (0, 2)
(c) (2, 0)
(d) (2, 1)
```
The following program compiles and runs without errors. What is the output produced by this program?

```c
#include <stdio.h>

void swap1(int a[], int b[])
{
    int temp = a[0];
    a[0] = b[0];
    b[0] = temp;
}

void swap2(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}

void swap3(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

void main(void)
{
    int a[2] = {2, 6};
    int b[2] = {0, 4};
    swap2(a[0], a[1]);
    swap1(a, b);
    swap3(&b[0], &b[1]);
    printf("%i, %i, %i, %i \n", a[0], a[1], b[0], b[1]);
}
```

(a) 2, 6, 0, 4

(b) 0, 6, 4, 2

(c) 2, 6, 4, 0

(d) 0, 4, 2, 6
10. Which of the following statements about `qsort` is FALSE?

(a) `qsort` takes a pointer to the first element of an array as the first argument and the name of a function as the fourth argument.

(b) `qsort` returns an integer value.

(c) The comparison function called by `qsort` can only take two pointer arguments.

(d) If the comparison function returns 0 `qsort` determines the two values are equivalent.

11. Which one of the answer choices is the correct way to call `qsort`, so that replacing the blank in the program below will allow the program to compile without errors?

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef char string[50];

int compare(string * l, string * r); /* prototype */
/* file named comp.c contains the code for the sort function named compare */
/* code in this file not shown here; just the prototype is shown */

void main(void)
{
    int i;
    int length = 5;

    ____________________________(call to qsort goes here)________________________

    for(i=0; i<length; i++)
        printf("%s ", instruments[i]);
}

(a) qsort(instruments, length, sizeof(instruments[0]), compare);
(b) qsort(string instruments[5], sizeof(string), compare, length);
(c) qsort(string instruments[5], length, sizeof(string), compare);
(d) qsort(instruments, length, 5, sizeof(instruments), compare);
```
The following program successfully compiles and runs. What is the output produced?

```c
#include <stdio.h>
#include <stdlib.h>

int compare(int *a, int *b)
{
    return *a - *b;
}

void main(void)
{
    int i;
    int redundant[10] = {3,7,1,9,2,7,1,5,9,3};

    qsort(redundant, 10, sizeof(int), compare);

    for(i=0; i<10; i++)
        printf("%i ",redundant[i]);
}

(a) 9 9 7 5 3 3 2 1 1
(b) 1 1 2 3 3 5 7 7 9 9
(c) 9 7 5 3 2 1
(d) 1 2 3 5 7 9
13. Which of the following comparison functions named order when replacing the blank below will sort the array in **ascending** alphabetical order, e.g. "dog", "fish", "lion", "owl", "zebra"?

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef char string[50];

_________(YOUR COMPARISON FUNCTION NAMED order GOES HERE)___________

void main(void)
{
    int i;
    qsort(pets, 5, sizeof(pets[0]), order);
    for(i=0; i<5; i++)
        printf("%s ", pets[i]);
}

(a) string order(string *a, string *b)
{
    if(*a > *b)
        return 1;
    else if(*a < *b)
        return -1;
    else return 0;
}

(b) int order(string *a, string *b)
{
    if(strcmp(*a,*b) > 0)
        return -1;
    else if(strcmp(*a,*b) < 0)
        return 1;
    else return 0;
}

(c) int order(string *a, string *b)
{
    return strcmp(*a,*b);
}

(d) string order(string *a, string *b)
{
    return strcmp(*b,*a);
}
```
The next four questions refer to the following structure data type.

```c
typedef struct
{
    int id;
    char name[20];
} Person;
```

14. Which is a correct way to declare a variable `p` of data type `Person`, and initialize its `id` field to 1001 and `name` field to "Jack"?

(a) `Person p = {1001, "Jack"};`
(b) `Person p = [1001, "Jack"];`
(c) `Person p = {"Jack", 1001};`
(d) `Person * p = {1001, "Jack"};`

15. Which is a correct way to print out the `name` field of variable `p` in the question above?

(a) `printf("%c", p->name);`
(b) `printf("%c", p.name);`
(c) `printf("%s", p->name);`
(d) `printf("%s", p.name);`

16. You are given an array named `people` of data type `Person`, that contains 20 elements that has been correctly initialized. Which is a correct way to sort the values in the `people` array with the `qsort` function and the comparison function named `compare_people`?

(a) `qsort(people, 20, sizeof(people), compare_people);`
(b) `qsort(people[0], 20, sizeof(people), compare_people);`
(c) `qsort(people, 20, sizeof(people), function compare_people);`
(d) `qsort(people, 20, sizeof(Person), compare_people);`

17. Fill in the blank to complete the code for the function named `compare_people` to sort the array named `people` (described in the previous question) by values in the `id` field in **ascending** order.

```c
int compare_people(Person* ptr1, Person* ptr2)
{
    return ______________;  
}

(a) (ptr1.id) - (ptr2.id)
(b) (ptr2.id) - (ptr1.id)
(c) ((*ptr2).id) - ((*ptr1).id)
(d) ((*ptr1).id) - ((*ptr2).id)
```
18. The program below compiles and executes without errors. What is the output of this program?

```c
#include <stdio.h>

typedef struct
{
  int a[5];
  int b[5];
} Test;

void mystery(int a[], int b[], int num)
{
  int i;
  for(i = 0; i < num; ++i)
  {
    int t = a[i];
    a[i] = b[i];
    b[i] = t;
  }
}

void display(int x[], int num)
{
  int i;
  for(i = 0; i < num; ++i)
  {
    printf("%i ", x[i]);
  }
}

void main(void)
{
  int i;
  Test x;
  for(i = 0; i < 5; ++i)
  {
    x.a[i] = i;
    x.b[i] = 5 - i;
  }
  display(x.a, 5);
  printf("\n");
  mystery(x.a, x.b,5);
  display(x.b, 5);
}
```

(a) 0 1 2 3 4
    5 4 3 2 1
(b) 0 1 2 3 4
    4 3 2 1 0
(c) 0 1 2 3 4
    0 1 2 3 4
(d) 5 4 3 2 1
    0 1 2 3 4
19. The program below compiles and executes without errors. What is the output of this program?

```c
#include<stdio.h>

typedef struct
{
    int x;
    char str[40];
} Test;

void modfunc1(Test t)
{
    t.x = 2 * t.x;
    t.str[0] = 'a';
}

void modfunc2(Test* t)
{
    t->x = 2 * t->x;
    t->str[0] = 'a';
}

void main(void)
{
    int i;
    Test tests[2] = { {1, "xxx"}, {2, "yyy"} };
    modfunc1(tests[0]);
    modfunc2(&tests[1]);
    for(i = 0; i < 2; ++i)
    {
        printf("%i %s\n", tests[i].x, tests[i].str);
    }
}
```

(a) 1 xxx
    2 yyy
(b) 1 xxx
    4 ayy
(c) 2 axx
    2 yyy
(d) 2 axx
    4 ayy
20. The program below compiles and executes without errors. What is the output of this program?

```c
#include <stdio.h>

void func(int **ptr)
{
    int temp = *(*ptr);
    (*ptr)[0] = (*ptr)[1];
    (*ptr)[1] = temp;
}

void main(void)
{
    int a[] = {10, 5, 0};
    int * c[2];

    c[0] = &a[0];
    c[1] = &a[1];

    func(c);

    printf("%i %i\n", *c[0], *c[1]);
}
```

(a) 10 5
(b) 10 0
(c) 5 0
(d) 5 10
21. The file data.txt is in your current working directory and contains the following text.

Original file contents.

After the following program is compiled and executed, what does the file data.txt contain?

```c
#include <stdio.h>

void main(void)
{
    int i, count = 0;
    char data[20][20];

    FILE * fileptr = fopen("data.txt", "w");

    fprintf(fileptr, "New file contents. \n");
    fclose(fileptr);
}
```

(a) New file contents.
(b) Original file contents.
(c) Original file contents. New file contents.
(d) Original file contents. New file contents.

22. What would happen if you tried to compile and run the following C program? You may assume that the file named input.dat already exists in your current working directory.

```c
#include <stdio.h>

void main(void)
{
    FILE * ptr;
    char file[] = "input.dat";
    char a[] = "a";

    ptr = fopen(file, a);

    fprintf(ptr, "%s\n", "Hello!");
    fclose(ptr);
}
```

(a) A compile error would occur.
(b) The file input.dat would be opened in append mode and "Hello!" written at the end of the file.
(c) The program would compile and run without any errors, but the file input.dat would not be opened.
(d) The file input.dat would be opened and all of its contents would be deleted then "Hello!" is written to the file.
The file data.csv is in your current working directory and contains the following data:

400,699,309,138,893

Fill in the blank in the following program such that the data, located inside the file data.csv, is successfully read into the array named data and then printed as follows:

400 699 309 138 893

```c
#include <stdio.h>

void main(void)
{
    int i, count = 0;
    int data[20];

    FILE * fileptr = fopen("data.csv","r");

    if(fileptr == NULL)
    {
        printf("File not opened");
        return;
    }

    while(EOF != _________________________)
        ++count;

    for(i = 0; i < count; ++i)
        printf("%i ",data[i]);

    printf("\n");

    fclose(fileptr);
}
```

(a) fscanf(*fileptr,"%i",&data[count])
(b) fscanf(fileptr,"%i",&data[count])
(c) fscanf(*fileptr,"%i",&data[count])
(d) fscanf(fileptr,"%[^;]; ",&data[count])
#include <stdio.h>

int func(int, int); /* prototype */

void main(void)
{
    int num1 = 10;
    int num2 = 42;
    int answer;

    answer = func(num1, num2);

    printf("%i\n", answer);
}

int func(int num1, int num2)
{
    if(num1%num2 == 0) /* VOOM!! */
        return num2;
    else
        return func(num2, num1%num2);
}

(a) 2

(b) 4

(c) 5

(d) 10
#include <stdio.h>

void star(int n, int k)
{
    if (n == k)
        printf("*\n");

    if (n < k)
    {
        printf("*");
        star(n+1,k);
    }
    else if (k > 1)
        star(1,k-1);
}

void main(void)
{
    star(1,6);
}

(a) *
   **
   ***
   ****
   *****
   ******
   *******

(b) ******
   *****
   ****
   ***
   **
   *

(c) ******
   *****
   ****
   ***
   **

(d) ******
Extra Credit

Answering the questions below correctly will add points to your exam total. Answering incorrectly or not answering will not add points to your exam total.

The next two questions refer to the following type definitions and variable declarations.

```c
typedef struct {
    int a;
    int b;
} S;

typedef struct {
    int c;
    S d;
    S *e;
} T;
```

```c
S s = {0,1};
T t = {2,{3,4}, &s};
T *pt = &t;
```

26. Which of following statements will **NOT** assign the value 6 to t.d.b?
   - (a) (t.e)->b = 6;
   - (b) ((&t)->d).b = 6;
   - (c) (pt->d).b = 6;
   - (d) (*pt).d.b = 6;

27. Which of following statements will dynamically allocate memory for an array of 100 elements of data type T and use pt as the array name?
   - (a) pt = malloc(100, sizeof(pt));
   - (b) pt = calloc(t, sizeof(100));
   - (c) pt = malloc(t, sizeof(100*t));
   - (d) pt = calloc(100, sizeof(T));