COP 2220 End of Semester Review

Instructor: Sareh Taebi

Fall 2021

Florida Atlantic University

About Final Exam

- Held online on Sunday Dec 12 .
- Exam is open from 1 pm – 4:30 pm (Exam duration is about 90 mins).
- You would need to install **Lockdown Browser** for this exam. It's the software that would do the proctoring.
- You can use a **cheat sheet**. Double sided letter-sized paper.
- The exam has the format of the quizzes (about 40 – 50 questions):
    - multiple choice
    - true/false
    - fill in the blanks

Lockdown Browser Instructions

- Lockdown Browser is a background program that would ensure the student stays in the exam page for the entire duration of the exam. Watch this video to get a basic understanding of LockDown Browser:
- https://www.respondus.com/products/lockdown-browser/student-movie.shtml
- **Download Instructions**
  Download and install LockDown Browser from this link:
- https://download.respondus.com/lockdown/download.php?id=721312624
- **Once Installed**
- Start LockDown Browser
- Log into to Canvas
- Navigate to the quiz
- Note: You won't be able to access a quiz that requires LockDown Browser with a standard web browser. If this is tried, an error message will indicate that the test requires the use of LockDown Browser. Simply start LockDown Browser and navigate back to the exam to continue.

Sample Problems

Attempt them all. Also review all the quizzes and the lectures!

Components of a Computer

- Which part of the computer coordinates all computer operations and performs arithmetic and logical operations on data? Which part stores data and results?

- What are the two types of main memory most computers have?

- What are registers and where are they located in a computer?

- List examples of input devices and output devices

- What is auxiliary memory?

C Language Elements

- Who developed the C language and what are some of its advantages?
- What is a preprocessor directive?
- What are the two parts of a function body? In which function does program execution begin?
- What is the difference between a standard identifier and a reserved word?
- List the syntax rules for valid identifiers and examples of valid vs. invalid identifiers

Overview of C

```
 1 #include <stdio.h>
 2 #define IN_TO_CM 2.54
 3 /* this program converts inches to centimeters */
 4 int main()
 5 {
 6     double in, cm;
 7
 8     printf("Enter the length in inches\n>");
 9     scanf("__",___);
10     cm = ______;
11     printf("___ inches = ___ centimeters\n",__,__);
12     return 0;
13 }
```

What is executed on line 3?

Complete the statements on lines 9, 10, and 11.

Output:

```
     17.45 inches = 44.32 centimeters
```

Overview of C

```
int goo = 5, bam = 12, hah = 6, eek;
double meh;

eek = goo + bam % hah;
meh = eek * goo + hah / 4 * 3;
```

What are the values of `eek` and `meh`?

Overview of C

```
#include <stdio.h>
int main()
{
    int moo = 6;
    double koi = 12.82673;
    printf("%9.3f%5d",koi,moo);
    return 0;
}
```

What will this program print?

Library functions

Given variables

```
    int c = 15;
    double n = 7.0, fn;
```

Write the assignment statement for variable `fn` using this equation:

$$f(n) = \sqrt{3n\left(\left\lfloor {}^{n}/_{4} \right\rfloor\right) + cn^2}$$

Note: $\lfloor {}^{n}/_{4} \rfloor$ is the floor of ${}^{n}/_{4}$

User-Defined Functions

Write the appropriate function prototypes:

- `donut()` takes an `int` argument and prints the number of `0`s in its `int` argument.

- `double_up()` prompts the user for an integer and returns two times that integer.

- `min()` returns the smaller of two `double` arguments.
  double min (double a, double b)

Conditions & logical Expressions

What is the numeric value of the following expression if `rah` is `17`, `nom` is `2` and `tot` is `9`.

```
    nom != 3 || tot < 7 && rah >= 10
```

Write the complement of the expression above.

Nested **`if`** Statement

What would be printed by this code fragment if `number` is `8`?

```
if (number > 6)
    if (number > 12)
        printf("You're close!\n");
    else
        printf("Sorry, you lose a  turn!\n");
```

switch statement

Rewrite the `if else-if` statement using a `switch` statement:

```
if (fruit == 'p' || fruit == 'P')
        papaya++;
else if (fruit == 'b' || fruit == 'B')
        banana++;
else if (fruit == 'm' || fruit == 'M')
        mango++;
else
        printf("Invalid entry\n");
```

True or False: You can use a switch statement on type `int`, `char`, and `double`.

nested loop

How many lines of data will the following program print?

```
#include <stdio.h>
int main()
{
     int i, j, k = 5;
     for(i = k; i > 0; i--)
     {
          for(j = i; j > 0; j--)
               printf("%d" ,i);
          printf("\n");
     }
     return 0;
}
```

Arrays

```
#include <stdio.h>
#define SIZE 10
int main()
{
     int i, fib[SIZE] = {0, 1};
     for(i = 2; i < SIZE; i++)
          fib[i] = fib[i - 2] + fib[i - 1];
     return 0;
}
```

What are the contents of array `fib`? Write a `for` loop that would print them.

What is the value of `fib[3] + fib[8]`?

Arrays and Pointers

```
#include <stdio.h>
#define SIZE 5
int main()
{
     int a[SIZE] = {12,4,8,15};
     int *aPtr;
     aPtr = a;
     aPtr += 3;
     *aPtr %= 2;
     aPtr--;
     *aPtr /= 2;
     return 0;
}
```

What are the final contents of array `a`?

Strings

- True of False: You should still apply the address-of operator (`&`) to a string argument passed to `scanf()` or to any other function.
- All strings are character arrays but not all character arrays are strings. Why?
- Write the `printf` statement that would print a string `studytime` left-justified.
- What is the difference between using functions `scanf()` and `gets()` to scan a string?
- What is buffer overflow?
- Which preprocessor directive should you use to modify strings in your program?

Strings

- Which `string.h` function is the equivalent of the assignment operator for strings?

- How is the operation of `strcpy()` different from the operation of `strncpy()`?

- Which two `string.h` functions are used to modify their first string argument by adding all or part of their second string argument at the end of the first argument? What is the difference between the two functions?

- Write an `if` statement utilizing the `strlen()` function that will concatenate two strings only if the sum of their lengths are less than the size of the receiving string.

- Illustrate the operation of `strcmp(z1,z2)` if `z1` is `"ziti"` and `z2` is `"zits"`.

ctype.h

- *True or False*: The `ctype.h` library can be used on characters and strings
- *True or False*: The ctype.h function `islower()` converts its character letter argument to the argument’s lowercase equivalent.
- Write an `if` statement using `isdigit()` that would print `Number.` if a character variable `ch` is one of the ten digits.

The `if` Statement

What would be printed by this code fragment if `happy` is `2`?

```
if (happy)
       printf("Nice to see you!\n");
   else
       printf("Put on a happy face\n");
```

User-Defined Functions

```
#include <stdio.h>
double math_stuff(double h, int k);
int main()
{
    int num1 = 7;
    double num2, num3;
    num3 = math_stuff(3.2,2);
}

double math_stuff(double h, int k)
{
    return h * k / 10.0;
}
```

What value is assigned to `num3` for the function call?

Remember the NOT rule!

Two-Dimensional Arrays

Write the appropriate declaration for array `pug`.
How many elements can be stored in this array?

`pug`

24	46	68	0
13	0	0	0
74	96	0	0

True or False: The value of `pug[2][1]` is `13`.

True or False: `pug *= 2` multiplies each element of `pug` by 2 .

Filling arrays using input prompt

Q: What are the values of elements or array orange?

```
int orange[4] = {0};

 int i, n;

 for  (i = 0;  i < 4;  ++i) {

           scanf("%d", &n);    /* 5, 1, 3, 6 are
consecutive input values */

           orange[n] = i;
```

Pointers

Trace the execution of the following program. What is being printed?

```
#include <stdio.h>
int main()
{
     int kip = 6, pal = 3;
     int *p1, *p2;
     p1 = &kip;
     p2 = &pal;
     p1 = p2;
     *p1 += *p2;
     *p2 = *p1 - 1;
     printf("%4d%4d%4d%4d\n", kip, *p1, pal, *p2);
     return 0;
}
```

Call-By-Reference Functions

Given the function definition

```
void zero(int * a)
{
    *a = 0;
}
```

Write a call to this function using a variable name of your choice.

What is the result of calling this function?

More Call-By-Reference Functions

Modify function `zero()` so that it initializes an array to all `0`s. Include an additional integer argument for the array size.

Strings

What will the following program print?

```
char str1[] = "gawsie";
char str2[] = "bletonism";
char whimsical[strlen(str1)+strlen(str2)];
strcpy(whimsical, str2);
strcpy(&whimsical[2], &str1[1]);
printf("%s!", whimsical);
```

You were an amazing group of enthusiastic programmers!

Thanks you for the great semester ☺