

COMP 2001/2401

Test #1

[out of 80 marks]

Duration: 90 minutes

Authorized Memoranda: NONE

Note: for all questions, you must show your work!

Name: _____

Student#: _____

1. What exact shell command would you use to: [6 marks]

a. change a directory's permissions to: drwxr-x--x

```
chmod 751 dir-name
```

```
also valid: chmod u=rwx,g=rx,o=x dir-name
```

b. change a file's permissions to: -rw-r-----

```
chmod 640 filename
```

```
also valid: chmod u=rw,g=r,o= filename
```

2. Given the following C code: [8 marks]

```
char c = 0xf4;  
printf("%d", c);
```

what is output: -12

- convert 0xf4 to binary
- apply two's complement
- have correct answer, including sign

3. Given the following C code:

[10 marks]

```
char x = 189;  
unsigned char y = x + 0x4f;  
printf("%d %d", x, y);
```

what is output: -67 12

Option 1:

- convert 189 to binary
- use two's complement to get proper decimal value: -67
- convert 0x4f to decimal: 79
- add both to get 12

Option 2:

- convert 189 to binary
- use two's complement to get proper decimal value: -67
- convert 0x4f to binary
- do binary addition to get 12

4. Show the binary representation of the following floating point number: -52.375

[7 marks]

Answer: 1100 0010 0101 0001 1000 0000 0000 0000

- sign bit: 1
- fixed point: $-52.375 = 2^5 + 2^4 + 2^2 + 2^{-2} + 2^{-3} = 110100.011 = 1.10100011 \times 2^5$
- exponent: $5 + 127 = 132 = 1000\ 0100$
- fraction: 1010 0011

5. Given the following C code:

[4 marks]

```
int j=10, k;  
k = --j;  
printf("%d ", k);  
k = j--;  
printf("%d ", k);
```

what is output: 9 9

6. Which bit model is not used by data types in C? why?

[6 marks]

signed-magnitude bit model

- 1). there are two possible bit values for zeros: 0000 0000 and 1000 0000
- 2). binary addition is somewhat complicated

7. Given the following code,

[20 marks] [Bonus marks]

```
struct Collection {  
    char        name[5];  
    unsigned char id;  
    int         year;  
    double      price;  
}  
  
struct Collection books [2];  
books[0].id = 2;  
books[0].year = 2012;  
books[0].price = 100.32;  
strcpy(books[0].name, "XI");  
books[1].price = 20.9;  
strcpy(books[1].name, "XII");
```

the memory map for this code could be written as follows. It is assumed that the addresses indicate the sizes of the variables, in bytes.

Address label(s)	Label(s)	Address	Value
books books[0].name	books[0] books[0].name[0]	400	'X'
	books[0].name[1]	401	'I'
	books[0].name[2]	402	'\0'
	books[0].name[3]-[4]	403-404	
	books[0].id	405	2
	books[0].year	406-409	2012
	books[0].price	410-417	100.32
books[1].name	books[1] books[1].name[0]	418	'X'
	books[1].name[1]	419	'I'
	books[1].name[2]	420	'I'
	books[1].name[3]	421	'\0'
	books[1].name[4]	422	
	books[1].id	423	
	books[1].year	424-427	
	books[1].price	428-435	20.9

8. Given the following code:

[9 marks]

```
char str[20];
strcpy(str, "Hello");
strcpy(&str[6], " world");
```

a. fill in each cell below with the contents of the str array after the statements execute:

H	e	l	l	o	0		w	o	r	l	d	0							
---	---	---	---	---	---	--	---	---	---	---	---	---	--	--	--	--	--	--	--

b. assuming that this additional statement executes: `strcat(&str[7], " Timmy");`

fill in each cell below with the new contents of the str array:

H	e	l	l	o	0		w	o	r	l	d		T	i	m	m	y	0	
---	---	---	---	---	---	--	---	---	---	---	---	--	---	---	---	---	---	---	--

- c. assuming that the statement in part (b) has executed, what is the output of `printf("%s",str):`

"Hello"

9. Given a complete C source code called **test.c** for a program, [13 marks]
shown in the following:

```
#include <stdio.h>

int main (int argc, char *argv[]) {
    int count;
    printf ("This program was called with \"%s\".\n", argv[0]);

    if (argc > 1) {
        for (count = 1; count < argc; count++) {
            printf("argv[%d] = %s\n", count, argv[count]);
        }
    }
    else{
        printf("The command had no other arguments.\n");
    }
    return 0;
}
```

Instructions:

- (1) what exact shell command you would use to compile and run the code? [3 marks]
The executable file name is called test.out.

`gcc test.c -o test.out`

(2) after compiling successfully, you have an executable file called test.out. You are going to give two tests for this executable file by running this executable file under Linux shell window to cover two kinds of conditions used by 'if' control structure.

- First, write down the command for running this executable file; [2 marks]

- second, write down its output. [3 marks]

Test condition 1:

`./test.out`

This program was called with ./test.out.
The command had no other arguments.

Test condition 2:

```
./test.out -option1 -option2
```

This program was called with ./test.out.

```
argv[1] = -option1
```

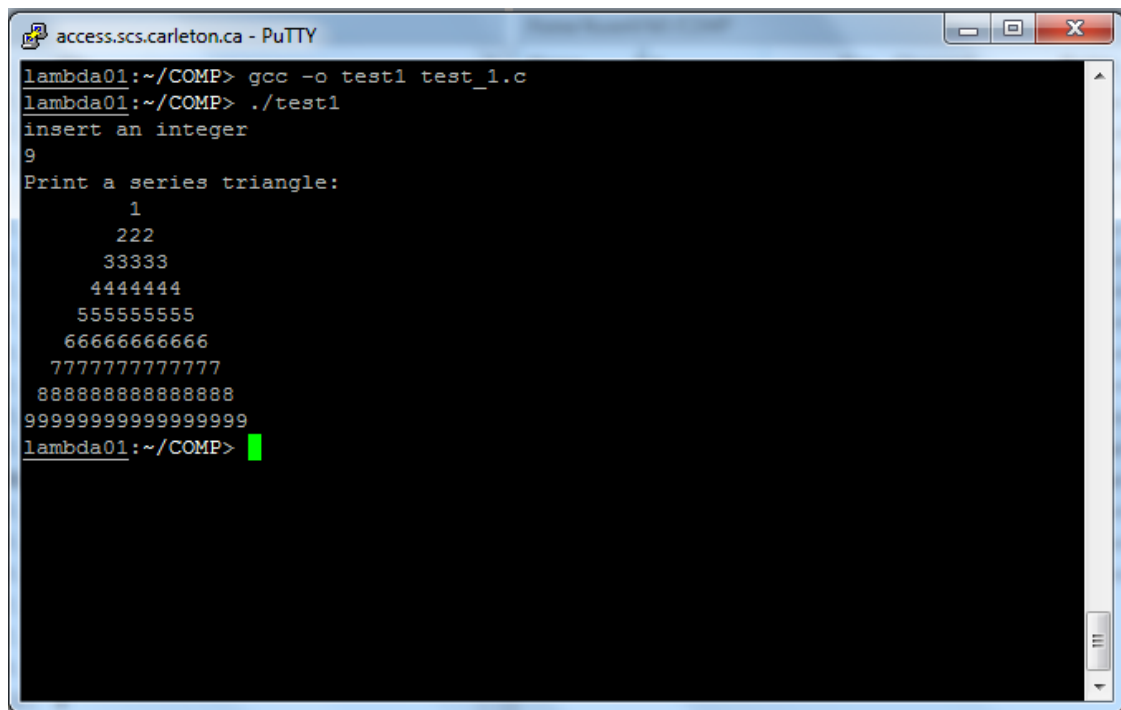
```
argv[2] = -option2
```

10. You are writing a small program to print a series triangle as follows. For this test, you will be filling in the missing portions in the program, whose goal is to manage the loop and print out the triangle displaying all those numbers, as found in the sample output below.

[17 marks]

Instructions:

- Read and understand the program provided as follows, which was used to generate the sample output
- Fill in the missing portions of the program
- **Note:** with the code that you fill in, the program must be complete, i.e. make **no** assumptions about "magic" functions that don't exist!
- **Note:** assume the input is valid and the input value is in the range of [1, 9].



```
access.scs.carleton.ca - PuTTY
lambda01:~/COMP> gcc -o test1 test_1.c
lambda01:~/COMP> ./test1
insert an integer
9
Print a series triangle:
 1
 22
333
4444
55555
666666
7777777
88888888
999999999
lambda01:~/COMP>
```

Source code for the program printing a series triangle

```
#include <stdio.h>

void main()
{
    int i, j, k, m = 1, num;

    // supply an integer number here
    printf("insert an integer \n");
    scanf("%d", &num);_ [2 marks]

    printf("Print a series triangle:\n");
    // print a series triangle
    for(i = 1, j = num - i; i <= num; i++, j--) [5 marks]
    {
        for(k = 1; k <= j; k++) [3 marks]
        {
            printf(" ");
        }
        for(k = 1; k <= m; k++) [3 marks]
        {
            printf("%d", i);_ [1 marks]
        }
        printf(" \n"); [1 marks]
        m = m + 2; [2 marks]
    }
}
```