Final Exam - Due 07/22/2020 @ 11:59 PM

NAME: Pavan Kumar Singara

Basics

Question 1
Tell us the sizes of the following data types in C:

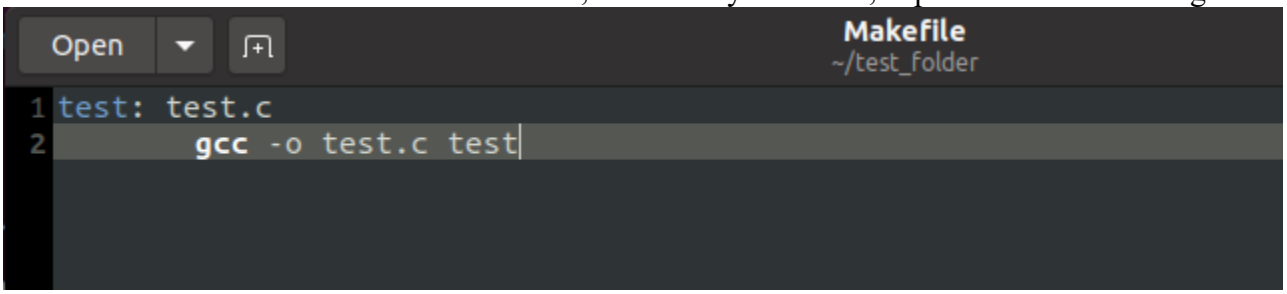
Data Type	Size (in bytes)
char	1 bytes
short	2 bytes
long	8 bytes

Question 2
Say we have a c file called **test.c**, what is the Linux terminal command we would use to compile it and create an executable called **a.out**.

Your Answer Here:

➔ gcc test.c or ./a.out

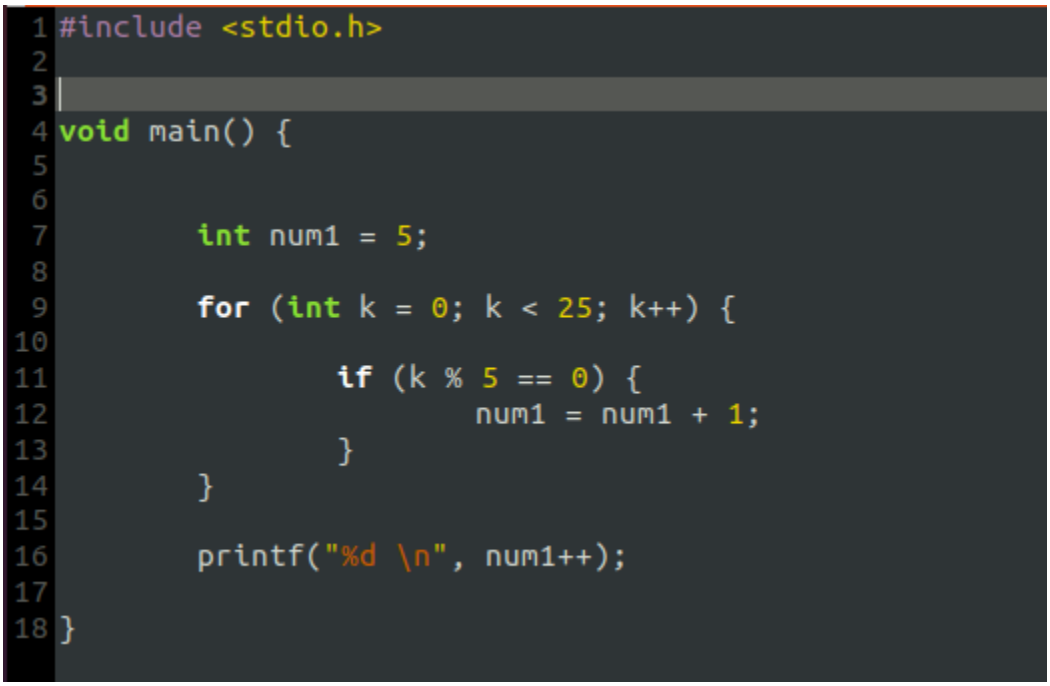
Question 3
Below is a screenshot of a Makefile I created, I made a syntax error, explain what I did wrong and how I can fix it.



Your Answer Here:

The syntax error is in “gcc -o test.c test” .
The proper syntax is “gcc -o test test.c “
Explanation: gcc converts the source code to executable instruction file, so it should convert test.c file to test!

Question 4
Below is a screenshot of a c file called **do_op.c**, what will the output be when I run this program? Use the space below to work through the code.



```c
#include <stdio.h>

void main() {


        int num1 = 5;

        for (int k = 0; k < 25; k++) {

                if (k % 5 == 0) {
                        num1 = num1 + 1;
                }
        }

        printf("%d \n", num1++);

}
```

Your Answer Here:

You can see here that ,

Num1 value is been initialized to 5 !
So, the next is for loop which checks the value less than 25 and adds up value num1 value by 1!
So, when k value is 0 and when the for loop checks num1 changes to be 6 , with the help of if condition !
So, when k value is 5 and if condition checks with 5 and num1 adds up with 1 so 5+1 = 7!
Similarily till when k = 20 and it can't check 25 because in for loop condition k is not equal to 7 so, lastly it prints out the vale of 10 with k = 20 !

Binary, Hex, and Bit Manipulation

Question 1
Convert the following decimal numbers into binary and hexadecimal: (I understand that you can just Google these conversions, but please do this on paper using the conversion techniques we learned! You **MUST SHOW ALL** work to get credit)

Decimal Number	Binary Representation (use 16-bits for each number)	Hexadecimal Representation
18		
45		
800		

Answer is attached below with written secreensot !

Question 2
In what kind of situation(s) would we want to use the hexadecimal number system?

Your Answer Here:

It is used to shorten the binary to make it more understandable! And it converts to 4 binary numbers so that its easy to understand ! I used hexadecimal in previous question as well which made me easy to convert it to 4 digit!

Question 3
Is it possible to multiply a number by 8 by only using a shifting operator? If so, explain how? (hint: think about this with binary numbers, write a few examples down below).

Your Answer Here:

Using left shift operator a number can be multiplied by 2 ^no_of_shifts
To multiply by eight no of shift = 3

e.g.
0001 - (1 is decimal)

Left shift by 3:
1000 - (8 in decimal)

Question 4
Perform the following bitwise operations (steps 1-3):

1. XOR 21 (00010101) with 9 (00001001)
2. AND the result of **step 1** with 15 (00001111)

3. OR the result of **step 2** with 1 (00000001)

After these operations, what decimal number do we have?

Your Answer Here:

Answer is attached below with written secreensot !

Functions

Question 1
Explain the difference between call by value and call by reference.

Your Answer Here:

Call by value copies the value from a specified addresses to the called function. As a copy is being passed, any modification done to the variable within the called function and it will not replicate in the calling function, unless the called function returns the value and is stored in the same variable in the calling function.

Call by reference the address of the location is passed to the called function. Now, you'll use a pointer to access the location, and usually the modifications done in the called function will use the reference of the original location, and therefore, the modification will replicate in the calling function, after the called function has been exited.

Question 2
Is it possible to pass an array to a function using call by value? Explain why or why not.

Your Answer Here:

Array to a function cannot be passed to a function using call by value directly. Instead, address of the array to a function can be passed by value to that function. It is done by passing a pointer to the first element of the array by value.
e.g.
char a[10];
foo(a);

Here array is not passed to the function. A pointer to its first element is passed instead.

Question 3

Below you will see two screenshots (option 1 and option 2), which of these is valid C code that will compile and run?

```c
#include <stdio.h>

int do_op(int x) {

        for (int k = 0; k < 25; k++) {

                if (k % 5 == 0) {
                        x = x + 1;
                }
        }

        return x++;
}

void main() {


        int num1 = 5;

        printf("%d \n", do_op(num1));


}
```

*Option 1

```c
#include <stdio.h>

int do_op(int x);

void main() {


        int num1 = 5;

        printf("%d \n", do_op(num1));


}

int do_op(int x) {

        for (int k = 0; k < 25; k++) {

                if (k % 5 == 0) {
                        x = x + 1;
                }
        }

        return x++;
}
```

*Option 2

Choose the correct option here, delete the 2 options that are incorrect so that you are only left with one:

1. Option 1 - NO
2. Option 2 – NO
3. Both – **YES**

Answer : It is BOTH !

Structs

Question 1
What is a struct and why is it useful?

Your Answer Here:

Struct holds several data items of the same kind. Structure creates record where you can declare structure variables with the help of ‘struct’ tag . and access the structure members. So, basically structure is a user defined data type and it creates data type and holds different types into a single type.

Pointers

Question 1
How can we use pointers to write more efficient code in terms of speed and memory usage? (hint: think about your lab 4 results)

Your Answer Here:

- ➔ It is much faster to copy only the memory location of the data rather than the entire content of the target data, thus pointers increase speed of operation.
- ➔ Pointers avoid initialization of large portions of memory with some elements. Thus saving space compared to programs with initialization of memory before usage is done.

Without Pointers:

```
For (int i=0; i<n; i++)
nArray[i] = nSomeValue;
```

With Pointers

```
For (int* ptrInt = nArray; ptrInt < nArray+n; ptrInt++)
*ptrInt = nSomevalue;
```

Question 2
Below you will see a screenshot of code that uses the **strtok** function, briefly explain how this function works. Your explanation must include details about how the function parses the data and how it uses a pointer to do this.

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

void main() {

        char my_string[50] = "Data_1, Data_2, Data_3";

        char *token;
        token = strtok(my_string, ", ");

        while (token != NULL) {

                printf("The token is -> %s \n", token);
                token = strtok(NULL, ", ");
        }

}
```

Your Answer Here:

In the above program , *token is the pointer . So, baiscally strtok function splits the string! So, my _string is declared as = “Data_1, Data_2, Data_3” So in the function strtok it splits with “ ,” therefore instead of printing horizontally , my_string[50] prints it in vertical way without “ , “ . And how is the pointer used here is , the value gets stored and returns the pointer which was stored in it !

Malloc

Question 1
In this class and on labs 6 & 7 we saw how we can use malloc to dynamically allocate space, explain what is meant by dynamically allocating space and how it is useful.

Your Answer Here:

malloc() function basically means memory allocation. It allocates the memory dynamically. It saves some space for specified size and returns the null pointer pointing to the memory location.

Dynamically allocating space means that when the memory is created dynamically directly at runtime, and this helps us to save space.

Example : ptr = (int*) malloc (50)

Linked Lists

Question 1

Part A.- Name and explain 2 advantages that a linked list has over a traditional array.

1. There is no memory wastage in Linked List
2. Data structures can be performer easier in Linked List

Part B.- Name and explain 2 advantages that a traditional array has over a linked list

1. The array has a specific address for each element so we can access memory of the data easier
2. The main advantage is the Binary search can be easily performed in the Dynamic array because we can find the middle element

1. $(18)_{10} = (?)_2$

2 | 18		
9	0	
4	1	↑
2	0	
1	0	
	1	

$(18)_{10} = (10010)_2$

$(18)_{10} = (?)_{16}$

16 | 18		
16 | 1	2	↑
	1	

$(18)_{10} = (12)_{16}$

3.

$(800)_{10} = (?)_2$

2	800	
2	400	0
2	200	0
2	100	0
2	50	0
2	25	0
2	12	1
2	6	0
2	3	0
	1	1

$(800)_{10} = (1100100000)_2$

$(800)_{10} = (?)_{16}$

16	800	
16	50	0
16	3	2
		3

$(800)_{10} = (320)_{16}$

Question H.

X-OR

```
  0 0 0 1 0 1 0 1
  0 0 0 0 0 1 0 0 1
  ---------------
  0 0 0 1 1 1 1 0 0
  ---------------
```

AND

```
  0 0 0 1 1 1 0 0
  0 0 0 0 0 1 1 1 1
  ---------------
  0 0 0 0 0 1 1 0 0
  ---------------
```

OR

```
  0 0 0 0 0 1 1 0 0
  0 0 0 0 0 0 0 0 1
  ---------------
  0 0 0 0 0 1 1 0 1
  ---------------
```

$1\ 1\ 0\ 1$

$= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 8 + 4 + 0 + 1$

$= (13)_{10}$

2. $(45)_{10} = (?)_{2}$

2	45	1
2	22	0
2	11	1
2	5	1
2	2	0
2	1	1
	1	

$(45)_{10} = (101101)_{2}$

$(45)_{10} = (?)_{16}$

16	45
16	2
	2

13 (D) $(45)_{10} = (2D)_{16}$