Q1 References & Assumptions

0 Points

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

If you have to make any unstated assumptions while answering any of the questions on the quiz, let us know the question numbers and assumptions you made here. You are not required to answer this question.

Q2 C Variable Declaration

8 Points

For each of the prompts below, either convert the C variable declaration into words or vice-versa

Q2.1

2 Points

A variable 'apple' that is an array of 4 pointers to ints

int *(apple[4])

Q2.2
2 Points

float *(*banana)(char*)

A variable 'banana' that is a pointer to the function taking in a pointer to a char that returns a pointer to a float

Q2.3
2 Points

A variable 'cat' that is pointer to an array of 5 ints

int (*cat)[5]

Q2.4
2 Points

int *(*(*dog)) ()

A variable 'dog' that is a pointer to a pointer to a function that returns a pointer to an int

Q3 Based on the code provided, answer the following questions

9 Points

```
typedef struct Student
{
   int age;
```

```
    char first_name[20];
    char *classes[5];
    float gpa;
} YellowJacket;

struct Student *sophie;
```

*Assume: sizeof(char) = 8 bits, sizeof(int) = 16 bits, sizeof(float) = 16 bits, sizeof(char *) = 32 bits for this question*

Q3.1 What is the size of the struct in bytes?
3 Points

*Remember to provide your answer in **bytes***

44

Q3.2 Which of the following are syntactically correct ways to set sophie's 'gpa' to 4.0 given the code snippet above?
3 Points

Select all that apply

- [x] (*sophie).gpa = 4.0
- [ ] sophie.gpa = 4.0
- [x] sophie->gpa = 4.0
- [ ] sophie.(*gpa) = 4.0

Q3.3 Create a new pointer to a struct with the name 'dayne' using the typedef 'YellowJacket'
3 Points

YellowJacket dayne

Q4 Consider the following code snippet
6 Points

```c
#include <stdio.h>

int main(void) {
   int x[20];
   printf("%d", sizeof(x));
   return 0;
}
```

What does the running this program print to the console?
*Assume: sizeof(char) == 1, sizeof(int) == 5, sizeof(int *) == 10 for this question*

10

Q5 Consider the following code and determine what the program will print.
9 Points

```c
#include <stdio.h>

void modify(int x, int *y, int *z) {
   x += 2;
   (*y) *= 5;
   *z = x + *y;
}

int main(void) {
   int num1 = 12;
   int num2 = 15;
   int num3 = 9;
   modify(num1, &num2, &num3);
   printf("num1: %d\n", num1);
   printf("num2: %d\n", num2);
   printf("num3: %d\n", num3);
   return 0;
}
```

Q5.1 Value of num1
3 Points

12

Q5.2 Value of num2
3 Points

75

Q5.3 Value of num3
3 Points

87

Q6 Consider the code provided below. Complete the swapStructs method, which takes in two puppy structs and swaps their stats. Assume the structs have already been initialized with some initial values, and any parameter name may be used in blank [1], as long as it is consistent with blank [2].
10 Points

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

   //stats struct
   struct stats {
      int age;
      double barkPower;
      double jumpHeight;
   }

   //puppy struct
   struct puppy {
      char *name;
      struct stats puppyStats;
   }

   int main() {
      struct puppy, p1, p2;
      swapStats(&p1, &p2);
      return 0;
```

```
}

int swapStats(____[1]____) {
    _____[2]_____;
}
```

Q6.1 For the labeled [1], fill in the parameter list according to the implementation of swapStats provided in the code (there can be multiple parameters).
4 Points

Hint: Consider what arguments the main method provides to swapStats

```
struct puppy *p1, struct puppy *p2
```

Q6.2 For blank [2], write the statements that will swap the stats of two puppies.
6 Points

```
struct puppy p3;
p3.name = (*p1).name;
p3.puppyStats.age = (*p1).puppyStats.age;
p3.puppyStats.barkPower = (*p1).puppyStats.barkPower;
p3.puppyStats.jumpHeight = (*p1).puppyStats.jumpHeight;

(*p1).name = (*p2).name;
(*p1).puppyStats.age = (*p2).puppyStats.age;
(*p1).puppyStats.barkPower = (*p2).puppyStats.barkPower;
(*p1).puppyStats.jumpHeight = (*p2).puppyStats.jumpHeight;

(*p2).name = p3.name;
(*p2).puppyStats.age = p3.puppyStats.age;
(*p2).puppyStats.barkPower = p3.puppyStats.barkPower;
(*p2).puppyStats.jumpHeight = p3.puppyStats.jumpHeight;
```

Q7 Memory Layout in C

4 Points

For the following code snippet, select where in memory each of the following are located.

```
#include <stdio.h>

static int minimum = 1000000;

int getMaximumOfList(int arr[], int currentCount, int size) {
   static int currentMax = 0;
   int currentVal = *arr;
   if (currentCount == size) {
      return currentMax;
   } else {
      if (currentVal > currentMax) {
         currentMax = currentVal;
      }
      getMaximumOfList(arr + 1, currentCount + 1, size);
   }
}

int maximum = 0;

int main(void) {
   int arr[] = {0, 6, 2, 3, 5};
   int size = sizeof(arr)/sizeof(arr[0]);
   maximum = getMaximumOfList(arr, 0, size);
   printf("%d\n", maximum);
   return 0;
}
```

Q7.1 currentMax is located at:
1 Point

- Stack
- Data
- Heap
- Program Text
- System Space

Q7.2 currentVal is located at:
1 Point

- Stack
- System Space
- Heap
- Data
- Program Text

Q7.3 currentCount is located at:
1 Point

- Program Text
- Heap
- System Space
- Data
- Stack

Q7.4 maximum is located at:
1 Point

- Data
- System Space
- Stack
- Heap
- Program Text

Q8 C Keywords
2 Points

Given the code, answer the following questions

```
extern int a = 10;
int foo()
```

```
{
   static int b = 30;
   return b;
}
```

Q8.1 In which memory region is 'b' stored?
1 Point

- not enough information
- code
- heap
- stack
- data

Q8.2 Is 'foo()'? visible outside of this C file?
1 Point

- Yes
- Not enough information
- No

Q9 Strings in C
4 Points

What does the following code print? If you think that this code breaks a C language rule and might cause an error, please fill in the blank with "Error" without the quotation marks.

```
#include <stdio.h>
int main(void) {
   char* str = "CS2110";
   str+=2;
   printf("%s", str);
   return 0;
}
```

Error

Q10 Strings in C
4 Points

What does the following code print? If you think that this code breaks a C language rule and might cause an error, please fill in the blank with "Error" without the quotation marks.

```c
#include <stdio.h>

int main(void) {
   char* str = "hello";
   str[2] = 'c';
   printf("%s", str);
   return 0;
}
```

heclo

Q11 Read the given block of code and describe in a few sentences what it is trying to accomplish. You can assume all the code compiles and there are no syntax or logical errors.
6 Points

```c
#include <string.h>

   void mysteryFunction(char str[]) {
      int myst_temp;
      int startC = 0;
      int endC = strlen(str) - 1;

      while (startC < endC) {
         myst_temp = str[startC];
         str[startC] = str[endC];
         str[endC] = myst_temp;
         startC++;
         endC--;
      }

   }
```

This code simply flips the inputted char array 'str'. It will swap the first and last element, the second and second last element, etc.

Q12 Write a macro called FUN_DIVIDE that divides numbers A and B then adds 9 to the result.
4 Points

```
#define FUN_DIVIDE(A,B): ((A/B) + 9)
```

Q13 In the table below you'll find a representation of memory in a computer. Each memory address is capable of storing 1 byte of information. For example, if one were to retrieve 2 bytes starting at address 0x123002, one would get 0xE058.
12 Points

Memory Layout

Mem Addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x12300_	AE	2F	E0	58	40	61	12	08	90	21	5C	2E	A0	BC	E8	E4
0x12301_	DB	8E	3C	2B	5E	8A	9F	F0	1E	54	1B	C2	D4	E3	A1	0F

Assume: sizeof(char) == 1, sizeof(short) = 2 , sizeof(int) = 4, sizeof(long) = 8 for this question

```
int *ptr1 = (int*) (0x123013);
short *ptr2 = (short*) (0x123008);
long *ptr3 = (long*) (0x123000);
```

Provide your answer in hexadecimal and include the 0x prefix in your answer

Q13.1 What does ptr3 hold?
3 Points

0x123000

Q13.2 What does ptr3 + 1 hold?
3 Points

0x123008

Q13.3 What does *(ptr1 + 2) equal?
3 Points

0xC2D4E3A1

Q13.4 What does *(ptr2 + 4) equal?
3 Points

0xDB8E

Q14 Debugging in C
7 Points

Your TAs claim that the following code is supposed to find a target character in a given string.

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

char findChar(char *str, char target) {
   int i = 0;
   while (i < sizeof(str)) {
      if ((str + i) == target) {
         return target;
      }
      i++;
   }
   return '0';
}
```

```
int main() {
   char* str = "hello";
   char target = 'o';
   char characterFound = findChar(str, target);
   printf("%c", characterFound);
   return 0;
}
```

If the code is incorrect, please specify at least one error and explain it. If the code is correct, please explain what the output looks like.

The line containing "if ((str + i) == target)" would be incorrect as str itself would be the address, which could not be compared directly to the char target. This would cause an error and would be incorrect.

Q15 Given this information, write this method:

15 Points

```
/*
\* @brief: This function removes any occurrences of a specified character from a
\*
\* @param str: pointer to an existing null-terminated string
\* @param char_to_remove: the character to be removed from the string
\* @return: the number of characters removed from the string
\*/

int remove_character(char \*str, char char_to_remove);
```

Successful Example:

```
char myStr[] = "hello, world!";

int removed_chars = remove_character(myStr, ',');

// myStr now contains "hello world!"
// removed_chars is 1
```

Requirements (Please read carefully, so you do not lose points):

- You may only use strlen() from string.h; any other string functions you might need should be written by you (though you may not need strlen().
- You should not create any new string or use additional memory to store the modified string. Modify the input string in place using pointers.
- The input argument can have any length but will always be null-terminated.

Hint from your TAs

To implement this method, consider using two different pointers. Using said pointers, iterate through the string, copying characters from read to write positions when they don't match char_to_remove. Advance both pointers accordingly.

Q15.1 Write your code here for int remove_character(char *str, char char_to_remove)
15 Points

```
int remove_character(char *str, char char_to_remove) {
int a = 0;
int b = 0;

while (str[a] != '\0') {
if (str[a] != char_to_remove) {
str[b] = str[a];
a++;
b++;
} else {
a++;
}
str[b] = '\0';
int numRemoved = a - b;
return numRemoved;
}
```

Q16 Extra Credit
2 Points

Here is a sample Makefile. For the first category, choose which option provides the correct command to compile an executable. For the second category, choose the option which will run the executable. There is only ONE correct answer per category.

```
# Quiz 4 - CS2110
# GCC flags from the syllabus (each flag described for the curious minds!)

CFLAGS = -std=c99                                    # Using the C99 standarc
CFLAGS += -Wall                                      # This enables all the v
CFLAGS += -Wextra                                    # This enables some extı
CFLAGS += -Werror                                    # Make all warnings intc
CFLAGS += -g                                         # Generate debugging inf

# Source files to be compiled together (for local command line testing)
CFILES = main.c hw7.c my_string.c

OBJNAME = hw7

# Note: '@' added to disable echo on the command
hw7: $(CFILES)
        @ # Compile all source files with the given flags into the specified exe
        @ gcc -fno-asm $(CFLAGS) $(CFILES) -o $(OBJNAME)

.PHONY: clean
clean:
        @ # Removing all sort of object files and executables
        @ rm -f $(OBJNAME) tests *.o *.out
```

Compile Executable

- run hw7
- compile hw7
- ./hw7
- hw7.c
- chmod +x make
- make main
- make gcc
- gcc hw7.c
- make hw7.c
- make hw7

Run Executable

chmod +x make

make main

hw7.c

gcc hw7.c

./hw7

make hw7.c

make hw7

make gcc

run hw7

compile hw7

Q17 Extra Credit
1 Point

In the early days of C language development, there was an influential book that served as both a tutorial and a reference for the language. This book was written by the creators of C. What is the full title of this book?

The Programming Language C

C: The Programming Language

The C Programming Language

Quiz 4C

● Graded

Student
Vidit Dharmendra Pokharna

Total Points

81 / 103 pts

Question 1

References & Assumptions **0** / 0 pts

Question 2

C Variable Declaration **8** / 8 pts

2.1 (no title) **2** / 2 pts

2.2 (no title) **2** / 2 pts

2.3 (no title) **2** / 2 pts

2.4 (no title) **2** / 2 pts

Question 3

Based on the code provided, answer the following questions **6** / 9 pts

3.1 What is the size of the struct in bytes? **3** / 3 pts

3.2 Which of the following are syntactically correct ways to set sophie's 'gpa' to 4.0 given the code snippet above? **3** / 3 pts

3.3 Create a new pointer to a struct with the name 'dayne' using the typedef 'YellowJacket' **0** / 3 pts

Question 4

Consider the following code snippet **0** / 6 pts

Question 5

Consider the following code and determine what the program will print. **6** / 9 pts

5.1 Value of num1 **3** / 3 pts

5.2 Value of num2 **3** / 3 pts

5.3 Value of num3 **0** / 3 pts

Question 6

Consider the code provided below. Complete the swapStructs method, which takes in two puppy structs and swaps their stats. Assume the structs have already been initialized with some initial values, and any parameter name may be used in blank [1], as long as it is consistent with blank [2]. **10** / 10 pts

6.1 For the labeled [1], fill in the parameter list according to the implementation of swapStats provided in the code (there can be multiple parameters). **4** / 4 pts

6.2 For blank [2], write the statements that will swap the stats of two puppies. **6** / 6 pts

Question 7

Memory Layout in C **4** / 4 pts

7.1	currentMax is located at:	**1** / 1 pt
7.2	currentVal is located at:	**1** / 1 pt
7.3	currentCount is located at:	**1** / 1 pt
7.4	maximum is located at:	**1** / 1 pt

Question 8

C Keywords **2** / 2 pts

8.1	In which memory region is 'b' stored?	**1** / 1 pt
8.2	Is 'foo()'? visible outside of this C file?	**1** / 1 pt

Question 9

Strings in C **0** / 4 pts

Question 10

Strings in C **0** / 4 pts

Question 11

Read the given block of code and describe in a few sentences what it is trying to accomplish. You can assume all the code compiles and there are no syntax or logical errors. **6** / 6 pts

Question 12

Write a macro called FUN_DIVIDE that divides numbers A and B then adds 9 to the result. **2** / 4 pts

Question 13

In the table below you'll find a representation of memory in a computer. Each memory address is capable of storing 1 byte of information. For example, if one were to retrieve 2 bytes starting at address 0x123002, one would get 0xE058. **12** / 12 pts

13.1	What does ptr3 hold?	**3** / 3 pts
13.2	What does ptr3 + 1 hold?	**3** / 3 pts
13.3	What does *(ptr1 + 2) equal?	**3** / 3 pts
13.4	What does *(ptr2 + 4) equal?	**3** / 3 pts

Question 14

Debugging in C **7** / 7 pts

Question 15

Given this information, write this method: **15** / 15 pts

15.1	Write your code here for int remove_character(char *str, char char_to_remove)	**15** / 15 pts

Question 16

Extra Credit **2** / 2 pts

Question 17

Extra Credit **1** / 1 pt