**CprE 288 Fall 2018 – Homework 2**

**Due Tuesday September 11 (on Canvas by midnight: 11:59pm)**

**Notes:**

· Homework must be typed and submitted as a PDF or Word Document (i.e. .doc or .docx) only.

· If collaborating with others, you must document who you collaborate with, and specify in what way you collaborated (see last page of homework assignment), review the homework policy section of the syllabus:<http://class.ece.iastate.edu/cpre288/syllabus.asp> for further details.

· Review University policy relating to the integrity of scholarship. See (“Academic Dishonesty”):<http://catalog.iastate.edu/academic_conduct/#academicdishonestytext>

· Late homework is accepted within two days from the due date. *Late penalty is 10% per day.* ***Except on Exam weeks****.*

· ***Note:*** *Code that will not compile is a typo. Answering a question as “will not compile”* ***will be marked incorrect****. Contact the Professor if you think you have found a typo.*

**Note: Unless otherwise specified, all problems assume the TM4C123 is being used**

## **Question 1 (10 pts): Memory Allocation (Reminder assume an TM4C123 is the target hardware unless otherwise specified)**

a. For each declaration indicate how many bytes will be allocated in memory (5 pts)

i)  **4**  int apple;

ii)  **4**  long my\_oranges[100];

iii)  **4**  signed float sensor\_readings[100];

iv)  **2 \_\_** unsigned short class\_grades[140];

v)  **4**  unsigned int scan\_data[10][100];

b. For each indicate the value of my\_length after executing each fragment of C code. Function strlen is a standard C library function. Specify N/A if the value of my\_length cannot be determined. (5 pts)

i) my\_length is 7 .

char msg[] = “CPRE288”;

int my\_length = 0xFFFFFFFF;

my\_length = strlen(msg);

ii) my\_length is N/A .

char msg[7] = “CPRE288”**;**

int my\_length = 0xFFFFFFFF;

my\_length = strlen(msg);

iii) my\_length is 4 .

char msg[100] = {’C’,’P’,’R’,‘E’,‘\0’,‘2’,‘8’, ‘8’,‘\0’};

int my\_length = 0xFFFFFFFF;

my\_length = strlen(msg);

iv) my\_length is N/A .

char msg[] = {5, 7, 10, 35, 77, 2, 3, 0, 5, 13, 23};

int my\_length = 0xFFFFFFFF;

my\_length = strlen(msg + 4);

v) my\_length is 0 .

char msg[] = “CPRE288”;

int my\_length = 0xFFFFFFFF;

my\_length = strlen(msg + strlen(msg));

## **Question 2 (10 pts): Memory Map**

a) Fill in the given memory map after the C fragment below has been executed. Note: The TM4C123 uses Little Endian ordering for memory (i.e. lower significant bytes of an element are stored at lower addresses) (7 pts)

char msg[4] = “Cat”;

short age[3] = {0x2061, 0x6955, 0x616B);

long speed[1] = {0x65756C22};

msg[3] = 0x63;

age[0] = age[0] + 7;

age[1] = age[1] – 5;

msg[13] = 0;

age[3] = 0x6863;

printf(“%s”, msg);

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Memory  Location | 0xFF00 | 0xFF01 | 0xFF02 | 0xFF03 | 0xFF04 | 0xFF05 | 0xFF06 | 0xFF07 | 0xFF08 | 0xFF09 | 0xFF0A | 0xFF0B | 0xFF0C | 0xFF0D |
| Value | 0x43 | 0x61 | 0x74 | 0x63 | 0x68 | 0x20 | 0x50 | 0x69 | 0x6B | 0x61 | 0x63 | 0x68 | 0x75 | 0 |
| Array | msg | | | | age | | | | | | speed | | | |
| Index | 0 | 1 | 2 | 3 | 0 | | 1 | | 2 | | 0 | | | |

b) What message will printf print for part a) (3 pts)

Catch Pikachu

## **Question 3 (5 pts): C-string formatting**

**Given:**

**char message[100];**

**char str1[] = “CprE”;**

**char str2[] = “iRobot”;**

**int num = 100;**

**char ch1 = 50;**

**char ch2 = 56;**

**Predict the C-string contained in message after each sprintf (2 pts each)**

**Note 1: Treat each part independently**

**Note 2: You may need to look up on your own more details on printf and sprintf**

**a) sprintf(message, “Our %s is moving”, str2);**

Our iRobot is moving

**b) sprintf(message, “Read %d datasheet pages every week”, num);**

Read 100 datasheet pages every week

**c) sprintf(message, “CprE%c%c%c is %s!”, ch1, ch2, ch2, “fun”);**

CprE288 is fun!

**d) sprintf(message, “Move the %s forward for %d cm”, str2+3, num/2);**

Move the bot forward for 50 cm

**e) sprintf(message, “The ASCII value for %c is decimal %d and hex %X”,**

**ch1, ch1, ch1);**

The ASCII value for 2 is decimal 50 and hex 32

## **Question 4: Struct and Union (5 pts)**

What is the size of the following data structures in bytes.

a) Size of **point3D** 9 (1 pts)

struct point3D {

signed long x;

unsigned int y;

char z;

};

b) Size of **val** 8 (1 pts)

union val {

char cval;

char str;

int ival[8];

float fval;

double dval;

};

c) Size of **compound** 9 (1 pts)

struct compound {

char \*mystring;

long y;

union {

char \*c;

int \*i;

float \*f;

} u;

};

d) Size of **more\_compound** 19 (2 pts)

struct more\_compound {

char \*name;

int age;

short pay;

long \*height;

union {

char short\_id;

int normal\_id;

char my\_id;

} id;

union {

char \*text\_data;

int \*numeric\_data;

long l\_numeric\_data;

} data;

};

## **Question 5: Pointers (10 pt)**

What is the value of a, b, c\_ptr at the end of each program. **Give N/A** if a value is undefined. It is recommended that you practices drawing a memory map for solving these questions.

**Assume the following memory locations for the variables:**

- Location of a is 0xFFFF\_FF00

- Location of b is 0xFFFF\_FE00

- Location of c\_ptr is 0xFFFF\_FD00

**a. 3pts**

void main()

{

char a = 5;

char b = 10;

char \*c\_ptr;

c\_ptr = &a;

\*c\_ptr = 12;

\*c\_ptr = b;

c\_ptr = &b;

\*c\_ptr = a;

}

a is 10 b is 10 c\_ptr is 0xFFFF\_FE00 .

**b. 3pts**

void main()

{

char a = 5;

char b = 10;

char \*c\_ptr = 0;

c\_ptr = &a;

c\_ptr = &b;

(\*c\_ptr)++;

c\_ptr++;

}

a is 5 b is 11 c\_ptr is 0xFFFF\_FE01 .

**c. 4pts (note all variables were changed from chars to ints or pointers to ints).**

void main()

{

int a = 5;

int b = 10;

int \*c\_ptr = 0;

c\_ptr = &b;

a = \*c\_ptr + b;

(\*c\_ptr)++;

c\_ptr++;

}

a is 20 b is 11 c\_ptr is 0xFFFF\_FF04 .

## **Question 6: Small Program (10 pts)**

Write a function that counts the occurrences of char1 in str1, called char\_count: (5pts)

// Count the number of occurrences of char1 in str1

int char\_count(char \*str1, char char1)

{

int count = 0;

for(int i = 0; i < strlen(str1); i++){

if(str1[i] == char1)

count++;

}

return count;

}

// example of char\_count()

#include <stdio.h>

int main(void)

{

char my\_str1[] = “hello world”;

char my\_char = ‘l’;

int my\_count = 0;

my\_count = char\_count(my\_str1, my\_char);

printf(“%s. has %c %d times \n”, my\_str1, my\_char, my\_count);

}

**Collaboration Documentation**

List the people (First and Last name) you collaborated with: .

For each collaborator, describe the manner in which you collaborated:

1)