Full Name Lab Section
CprE 288 – HW2
Due Thursday, September 6, 2012

Question 1: Variable names (10 pt)

Indicate if each variable name is legal (Yes), or not legal (No)

YES MyNumber

YES Dr_Jones

NO switch **(this is a keyword)**

NO 12months **(not allowed to start with a number)**

YES _pie

NO while **(this is a keyword)**

NO if **(this is a keyword)**

NO struct **(this is a keyword)**

YES ENUM **(C is case sensitive so this is not a keyword)**

YES IF **(C is case sensitive so this is not a keyword)**

Full Name Lab Section
CprE 288 – HW2
Due Thursday, September 6, 2012

Question 2: Array sizes and element values (10 pt)

a. How many bytes are allocated for each of the arrays? (5pts)

12 char num_days[12];

13*4 = 52 signed long num_months[13];

50*2 = 100 unsigned int my_dates[50] = {5, 56, 7, 9};

9*1 = 100 char my_string1[9] = {'C', 'P', 'R', 'E', ' ', '2', '8', '8', '!'};

17 char my_string2[] = "CPRE 288 is fun!";

17 because the compiler will allocate space for the string literal + 1 byte of zero. Were the string literal is defined to be the text between the double quotes.

b. What is the value of each of the following (in decimal), using the array declarations from part a? Use N/A if the value is unknown (5pts)

7 my_dates[2]

N/A num_days[10]

'E' = 69 my_mystring1[3]

N/A my_string1[9**]** **Index 9 is outside the my_string1 array boundary.**

0 my_string2[16**]** **Index 0-15 contain the string literal and the complier adds a 0 byte to the end at index 16.**

Full Name Lab Section
CprE 288 – HW2
Due Thursday, September 6, 2012

Question 3: C-strings (10 pt)

a. Explain the importance of the NULL character (i.e. '\0' or 0) in a C-string (2pts)

Ans.

The importance of the NULL character is to signal the end of a valid C-string . String manipulation functions such as "strlen" use the NULL character as an indicator that the end of a C-string has been reached. Without the NULL character, the result returned by the string manipulation functions is unpredictable. Which could cause the program to crash, hang, or misbehave.

b. Give the value assigned to the variable. Use N/A if the value is unknown (8 pts)

Note: strlen() is a C library function that returns the length of a string.

i. The value of my_len after executing the code below is **7** .

```
char my_string[] = "CPRE288";
int my_len;
my_len = strlen(my_string);
```

ii. The value of my_len after executing the code below is **N/A** .

NULL character (i.e. 0) is missing, so strlen does not know when to stop.

```
char my_string[7] = {'C', 'P', 'R', 'E', '2', '8', '8'};
int my_len;
my_len = strlen(my_string);
```

iii. The value of my_len after executing the code below is **4** .

```
char my_string[15] = {'C', 'P', R', 'E', 0};
int my_len;
my_len = strlen(my_string);
```

iv. The value of my_len after executing the code below is **N/A** .

NULL character (i.e. 0) is missing, so strlen does not know when to stop.

```
char my_string[2] = {'C', 'D'};
int my_len;
my_len = strlen(my_string);
```

Full Name Lab Section
CprE 288 – HW2
Due Thursday, September 6, 2012

Question 4: Practice with arrays (10 pt)

i. Write a C program that copies the contents of array1 to array2 (5 pts)

```
char array1[] = "hello";
char array2[12];

main()
{
  int len;
  len= strlen(array1);   // Get length of C-string

  for(int i=0;i<len;i++)  // Copy element by element
  {
    array2[i]=array1[i];

}
```

ii. Write a simple C program that copies the contents of array1 to array2 (5 pts)

```
char array1[12] = {'h','e','l','l','o', 0,'w','o','r','l','d', '\0'};
char array2[12];

main()
{

// Note: cannot use strlen, because it will only count up to the first
// 0 (i.e. NULL character).  Thus, returning a length of only 5.

  for(int i=0;i<12;i++)   // Copy element by element
  {
    array2[i]=array1[i];
  }

}
```

Full Name Lab Section
CprE 288 – HW2
Due Thursday, September 6, 2012

Question 5: Debug Practice (10 pt)

Visit this website: http://www.thesupercars.org/top-cars/most-expensive-cars-in-the-world-top-10-list/

Your job is to verify code responsible for the "auto stop" safety system of the "Lamborghini Reventon" sports car. If the code fails, the cost of car repair comes out of your paycheck. Unfortunately, the writer of the code was not paying attention when they took CPRE 288!

For the following code a) indicate two bugs in the program (4pts), b) describe the type of issue each bug will cause (4pts), and c) fix the two bugs (2pts).

```
// Assume the appropriate #include files have been added.

// Assume get_distance returns a char indicating distance in feet,
// and has been defined for you.

// Assume stop_car stops the car, and has been defined for you.

// Assume message_array is placed by the compiler at address 0xFF00
// Assume distance_to_object is placed by the compiler at address 0xFF03
```

BUG: `char message_array[3]; // Hold message to print`

ISSUE: message array is not large enough to hold the C-string STOP. The array must be at least size 5. 5 = 4 + 1 NULL byte.

IMPACT: The assignment of 'P' (80 in decimal) will be written outside of the boundary of message_array and will overwrite the variable distance_to_object. When we check if the car is closer than 2 feet to an object, the variable will have a value of 80 instead of the value that was returned by get_distance. Thus allowing the car to crash.

FIX: `char message_array[5]; // Hold message to print`

```
char distance_to_object =  0;  // Distance in feet

int main(void)
{
  while(1)
  {
    distance_to_object = get_distance(); //Distance of car from object
                                         // Assume function returned 1
```

Full Name Lab Section
CprE 288 – HW2
Due Thursday, September 6, 2012

```
    //Fill message_array with the C-string STOP if 1 foot from an object
    if(distance_to_object == 1)
    {
      message_array[0] = 'S';
      message_array[1] = 'T';
      message_array[2] = 'O';
      message_array[3] = 'P';
```

BUG: Forgot to end the C-string with a null byte

ISSUE: printf will not know where to stop printing if there is no NULL byte

IMPACT: printf will print garbage.

FIX: message_array[4] = 0; // assign NULL byte

```
      printf("%s", message_array);
    }

    // If the car is closer than 2 feet to an object, stop
    if(distance_to_object <  2)
    {
      stop_car();
    }
  }
  return 0;  // Program should never get here
}
```