

ME 333 Quiz 1: C

Marshall Johnson

January 13, 2022

Quiz 1

- 1) The following code will overflow. Explain what that means and edit the code to prevent it:
- ```
unsigned char a = 175; unsigned char b = 100;
unsigned char c = a + b;
```

An unsigned char has a range of 0 to 255. Both a and b fall within that range, but the sum  $a + b$  does not. This results in overflow, since  $a + b = 275$ , which is out of the unsigned char range. To have the correct sum, a 1 in the  $2^8$  column would be needed. Since this falls outside of the 8 bits of the unsigned char, only the remainder is left — **19**.

To prevent this, the code could be modified as follows:

```
int a = 175;
int b = 100;
int c = a + b;
```

This would give the correct value — 275.

- 2) In a sentence or two, describe the difference in using
- ```
int numStates = 3;
and
#define NUMSTATES 3;
```

The first line defines a **variable** of type int equal to 3. This variable can be reassigned throughout the program. In contrast, the second line defines a **constant** equal to 3. This cannot be assigned a different value in the program. Additionally #define is a preprocessor command that results in the substitution of the constant value throughout the program during the preprocessing step of compilation.

- 3) Describe what happens during the compilation process when you build a project with multiple C files.

When building a project with multiple C files, each of the C files is compiled independently (typically this relies on an associated header file). The compilation and assembly of each C file results in an object code file. These object code files are then linked into a final executable.

- 4) For the following array, state the result if it exists or say "unknown":
- ```
unsigned char q[5] = {2, 12, 'j', 128, 10};
unsigned char a; int b;
```

- a.  $a = *(q+1); \rightarrow 12$
- b.  $b = ((int)q[3]) << 2; \rightarrow 512$
- c.  $a = q[q[0]]+1; \rightarrow 107$
- d.  $b = q[3]*2; \rightarrow 256$
- e.  $b = q[4] / q[5]; \rightarrow \text{unknown}$

- 5) Use a while loop to print all lowercase letters of the alphabet, each on its own line.

```
int i = 97;
while (i < 123) {
 printf("%c", i);
 i++;
}
```

- 6) Create a structure called *car* that contains an interger called *id*, a character array with 50 elements called *brand*, and a float array with 10 elements called *miles*. Write the code that creates an instance of *car*, and initializes the *id* to 1, the first element of *brand* to the null character, and every element of *miles* to 0.0

```
typedef struct {
 int id;
 char brand[50];
 float miles[10];
} car;

car c;
c.id = 1;
c.brand[0] = 0;
for (i=0; i<10; i++){
 c.miles[i] = 0.0;
}
```

- 7) Write the code that asks the user for the name of a car brand and puts the response in the brand field of your answer to #5.

```
printf("Please enter the name of a car brand (max 50 characters).");
scanf("%s", c.brand);
```

- 8) Write two functions that produce the same result, one that uses type car by value and one that uses car by reference. In the functions, calculate the value of each element of miles from element 2 to 10, assuming the number of miles doubles from each previous index, and returns to main with the calculated values stored inside the variable:

By value:

```
typedef struct {
 int id;
 char brand[50];
 float miles[10];
} car;

car calc_miles(car c);

int main(void) {
 car c;
 c.miles[0] = 1;

 c = calc_miles(c);
 return 0;
}

car calc_miles(car c) {
 int i;
 for (i=1; i<10; i++) {
 c.miles[i] = c.miles[i-1]*2;
 }
 return(c);
}
```

By reference:

```
typedef struct {
 int id;
 char brand[50];
 float miles[10];
} car;

car calc_miles(car c);

int main(void) {
 car *cp;
 car c;
 c.miles[0] = 1;

 cp = &c;
 c = calc_miles(cp);
 return 0;
}

car calc_miles(car *cp) {
 int i;
 for (i=1; i<10; i++) {
 cp->miles[i] = cp->miles[i-1]*2;
 }
 return(*cp);
}
```

- 9) Write a complete program (here on the page, you don't have to write or test the code with gcc) that asks the user to enter a string and a number, and then prints the string back with each character shifted over in the ASCII table by the number they entered. For example, if the user enters "ABCD 4", the result is "EFGH". Limit the user to entering a shift value between 1 and 8 (inclusive), keep asking for a shift value until it is in the correct range.

```
/*
 * shift.c
 *
 * This program takes an input string and number and shifts
 * each character of the string over by the number given,
 * according to the ASCII table.
 *
 * This program was written to answer Question 9 of ME333 Quiz 1.
 */

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

static char s[100];
static int shift = 0;

int main(void) {

 // Take user input and assign to variables
 printf("Please enter a string followed by a number with a space\nin between.\nThe number should be between 1 and 8 (inclusive):\n");
 scanf("%s%d", s, &shift);

 // If user inputs number out of range, ask for another number
 while (shift < 1 || shift > 8) {
 printf("Your number is out of range. Please enter a number\nbetween 1 and 8:\n");
 scanf("%d", &shift);
 }

 // Print out adjusted string
 printf("Your adjusted string is:\n");
 int i;
 for (i=0; i<strlen(s); i++) {
 printf("%c", s[i]+shift);
 }
 printf("\n");

 return 0;
}
```