APS 105: Computer Fundamentals

Tutorial #1
Summer 2025

Problem 1: Printing

Write a C program that produces the following output. The output must be exactly as shown:

```
C uses escape sequences for a variety of purposes.
Some common ones are:
to print ", use \"
to print \, use \\
to jump to a new line, use \n
```

Example solution:

```c
#include <stdio.h>

int main(void) {
printf("In C, the program to print \"hello, world\" is\n");
printf("\n");
printf("#include <stdio.h>\n");
printf("\n");
printf("int main(void)");
printf(" {\n");
printf("    printf(\"hello, world\\n\");\n");
printf("    return 0;\n");
printf("}\n");
}
```

Problem 2: Unit Conversion

Write a C program that asks the user to enter a distance (assumed to be in metres). Convert this distance to yards, feet, inches and a decimal number, rounded to two decimal places, that indicates any remaining fraction of an inch. Use the conversion factor 1 inch = 2.54 cm. For example, if the user enters a value of 3.376, the output will be:

```
3 yards , 2 feet , 0 inches , 0.91 inches remainder
```

The prompt to the user should take the form:

```
Please provide a distance in metres:
```

written by itself on a single line.

Note: if the input leads to something like the following:

```
1 yards , 1 feet , 1 inches , 0.00 inches remainder
```

This is fine. You do not need to change the output to read 1 yard, 1 foot, 1 inch. Likewise, if yards, feet or inches have a value of 0, no need to remove their prints from the output.

```c
#include <stdio.h>

int main(void) {
const double CmPerInch = 2.54;
const double CmPerMetre = 100.00;
const int InchesPerFoot = 12;
const int InchesPerYard = 36;

double distance;
printf("Please provide a distance in metres: ");
scanf("%lf", &distance);

double distanceInInches = distance * CmPerMetre / CmPerInch;

// truncate fractional part to get # of inches
int inches = (int)distanceInInches;
distanceInInches = distanceInInches - inches;

int yards = inches / InchesPerYard;

// how many inches are left after extracting yards
inches = inches % InchesPerYard;

int feet = inches / InchesPerFoot;

// how many inches are left after extracting feet
inches = inches % InchesPerFoot;

printf("%d yards, %d feet, %d inches, %.2f inches remainder\n", yards,
   feet,
inches, distanceInInches);

return 0;
}
```

Problem 3: Deciphering a code

The scheme you use to encode a 4-digit number takes the real combination and swaps the 1st and the 4th digit of the combination, and replaces each of the 2nd and 3rd digits by their 9's complement. In other words, a combina- tion of the form abcd is encoded as $d(9-b)(9-c)a$. For example, if the actual combination is 0428, the encrypted combination will be 8570.

Write a C program to implement this code-decode scheme. The user (you!) will, then, enter an encrypted combination and the program will output the real combination. A sample output from an execution of the program appears below:

```
Enter an encrypted 4-digit combination: 8021<enter>
The real combination is: 1978
```

When reading the 4-digit combination from user input, you are not allowed to scan in individual characters; instead, you should scan in a single integer using scanf.

Example solution:

```c
#include <stdio.h>

int main(void) {
int encComb;
printf("Enter an encrypted 4-digit combination: ");
scanf("%d", &encComb);

// Determine the 4 digits of the encrypted combinaiton.
int d4, d3, d2, d1;
d4 = encComb / 1000;
encComb = encComb % 1000;
d3 = encComb / 100;
encComb = encComb % 100;
d2 = encComb / 10;
encComb = encComb % 10;
d1 = encComb;

// printing the decryped combination: d4 and d1 are swaped. d3 and d2 are
// are 9's complemented
printf("\nThe real combination is: %d%d%d%d\n", d1, 9 - d3, 9 - d2, d4);

return 0;
}
```

Problem 4: Simple Loop (Winter 2022, Q8)

Write a complete C program that calculates and prints the sum of all numbers between 1 and 999 (inclusive) that satisfy all the following conditions:

1. The number is divisible by 9.

2. The number is even.
3. The ten's digit of the number is not 7. For example, the ten's digit in 753 is 5, the ten's digit of 671 is 7, and the ten's digit of 9 is 0.

Example solution:

```c
#include <stdio.h>

int main() {
  int sum = 0;
  for (int i = 1; i < 1000;i++) {
    if (i%9==0 && i%2==0 && i/10%10!=7) {
    sum += i;
  }
}
printf("%d\n", sum);

return 0;
}
```