CS 1313 010 Fall 2025 Homework #10

Quiz to be held in class 9:00-9:15am Mon March 31 2025

Please feel free to discuss these questions with your classmates, but **NOT** to copy each other.
NOTE: Except where and as explicitly permitted in writing (for example, in a Homework),
you are **ABSOLUTELY FORBIDDEN** to **COPY EVEN A SINGLE CHARACTER** from,
or to have **ANY** shared code with, **ANY** other entity,
whether a human being (regardless of whether in CS1313 or not),
a text resource, a computing resource or anything else,
whether in person, on a local computer, online or anywhere else.
It's **INCREDIBLY EASY** for us to detect such copying, so **DON'T EVEN THINK ABOUT IT!**

1. What are the **FIVE STEPS** that describe the execution of a `for` loop?

   (a)

   (b)

   (c)

   (d)

   (e)

2. For each of these kinds of statements, mark **CAN** if it can appear in the body of a `for` loop, and mark **CANNOT** if it cannot appear in the body of a `for` loop. **EXPLAIN.**

   (a) A named constant declaration

   (b) A variable declaration

   (c) A `printf` statement

   (d) A `scanf` statement

   (e) An assignment statement

   (f) A `exit` statement

   (g) An `if` block

   (h) A `while` loop

   (i) A `for` loop

3. Convert the following count-controlled `while` loop into a `for` loop.

```
index = initial_value;
while (index <= final_value) {
    sum = sum + index * 2;
    index++;
} /* while (index <= final_value) */
```

4. Convert the following count-controlled `while` loop into a `for` loop.

```
count = initial_value;
while (count <= final_value) {
    printf("count = %d\n", count);
    count += stride;
} /* while (count <= final_value) */
```

5. Convert the following `for` loop into a count-controlled `while` loop.

```
for(count = 1; count <= n; count++) {
    n_factorial = n_factorial * count;
} /* for count */
```

6. What is the **OUTPUT** of each of these programs? If you aren't confident of an answer, type in, compile and run the program to test it.

   (a)
```
#include <stdio.h>
int main ()
{ /* main */
    int count;
    int sum;

    sum   = 0;
    count = 1;
    while (count <= 6) {
        sum   = sum   + count;
        count = count + 1;
    } /* while (count <= 6) */
    printf("sum = %d\n", sum);
    return 0;
} /* main */
```

   (b)
```
#include <stdio.h>
int main ()
{ /* main */
    int count;
    int sum;

    sum   = 0;
    count = 1;
    while (count <= 6) {
        sum   = sum   + count;
        count = count + 2;
    } /* while (count <= 6) */
    printf("sum = %d\n", sum);
    return 0;
} /* main */
```

   (c)
```
#include <stdio.h>
int main ()
{ /* main */
    int count;
    int product;

    product = 1;
    count   = 1;
    while (count <= 16) {
        product = product * count;
        count   = count   + 4;
    } /* while (count <= 16) */
    printf("product = %d\n", product);
    return 0;
} /* main */
```

(d)
```
#include <stdio.h>
int main ()
{ /* main */
    int count;
    int product;

    product = 1;
    count   = 1;
    while (count <= 17) {
        product = product * count;
        count   = count   + 4;
    } /* while (count <= 17) */
    printf("product = %d\n", product);
    return 0;
} /* main */
```

(e)
```
#include <stdio.h>
int main ()
{ /* main */
    const int lower_bound =  1;
    const int upper_bound = 18;
    const int stride      =  4;
    int count;
    int product;

    product = 1;
    count   = lower_bound;
    while (count <= upper_bound) {
        product = product * count;
        count   = count   + stride;
    } /* while (count <= upper_bound) */
    printf("product = %d\n", product);
    return 0;
} /* main */
```

7. The *combination* of *n* items taken *r* at a time, denoted ${}_{n}\mathrm{C}_{r}$ and pronounced "n choose r," is the number of ways in which you can choose a subset of r items from a set of n items:

$$ {}_{n}\mathrm{C}_{r} = \frac{n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-r+1)}{1 \cdot 2 \cdot 3 \cdot \ldots \cdot r} $$

For example, if you have a standard 52 card poker deck (assuming no jokers and nothing wild), the number of possible 5 card poker hands is "52 choose 5:"

$$ {}_{52}\mathrm{C}_{5} = \frac{52 \cdot 51 \cdot 50 \cdot 49 \cdot 48}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5} = \frac{311,875,200}{120} = 2,598,960 $$

Write a C program that takes as its input n, the total number of items available (for example, 52 cards in a deck), and r, the size of the subgroup to be created (for example, 5 cards in a poker hand), and outputs ${}_{n}\mathrm{C}_{r}$. (For this homework question, you do not need to include comments if you don't want, and you may use numeric literal constants in the body of the program.)

If you use **ANY** resources other than Dr. Neeman, the TAs/graders (Basiri, Bilal), the course textbook or the materials posted on the course webpage, you **MUST** reference them on the quiz. **THIS INCLUDES CLASSMATES, FRIENDS, PROFESSORS, ONLINE RESOURCES, ETC.**