1] Which of the following gives the memory address of the first element in array foo, an array with 100 elements? (Circle all that apply.)

A. foo[0] B. &foo[1] C. **foo**  D. **&foo[0]**

E. **&foo** F. \*foo G. foo[1]; H. 42

2] We define a pointer to bar:

**int bar[3] = {20, 30, 40};**

**int \*p\_bar = bar;**

Let’s assume that the array starts in memory at address 0x2000. What is the effect of these commands?

**printf("%d", \*p\_bar);** 20

**printf("%d", \*(p\_bar+2));** 40

**printf("%p", p\_bar);** 0x2000

**printf("%p", p\_bar++);**  0x2000, points to bar[1]-0x2004

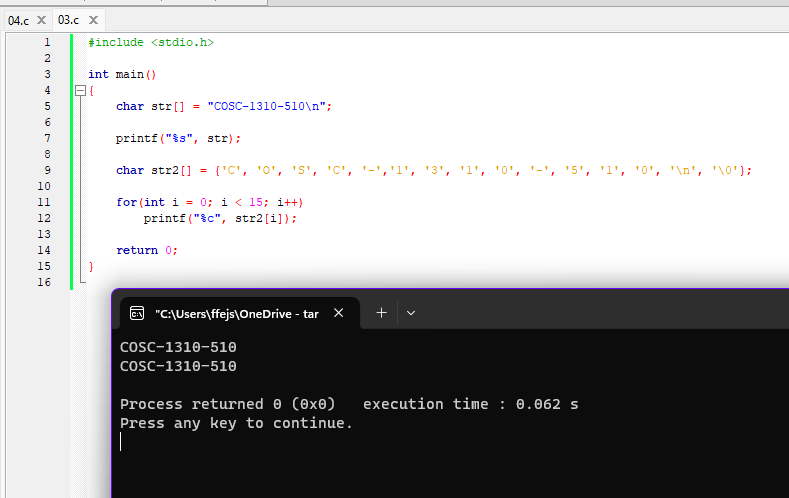
**printf("%p", bar++);**  This is an error.

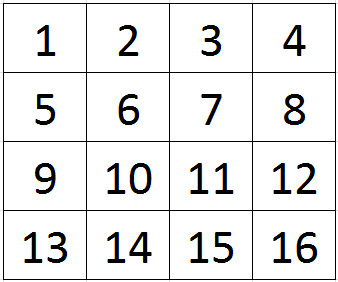
**printf("%p", ++p\_bar);**  0x2004

**printf("%d", \*p\_bar++);**  20, points to bar[1]-0x2004

3] Place in an array of characters (a.k.a. string) the number and name of our course, and then display it two ways:

* All at once, with a %s placeholder in the printf function.
* Character-by-character, with a for loop and a %c placeholder in the printf function.



4] Write a program that declares a 10x10 array (matrix) of integers, and initializes it with the pattern shown:

Hint: Use a counter variable!

A screenshot of a computer

Description automatically generatedI thought it was a fun challenge to do it in one loop.

**Problems to be reported (all of them):**

(Include code and execution screenshots of programs)

5] The array **bar** was declared and initialized like this:

**short bar[3] = {20, 30, 40};**

What does the expression **\*bar** evaluate to? 20, bar[0]

How about **\*(bar + 1)** ? 30, bar[1]

How about **\*bar + 2** ? 22, bar[0]+2

A computer screen shot of a program

Description automatically generated

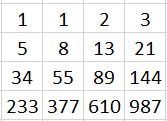
6] Write a program that asks the user to enter a string, and then uses a pointer variable to access and print the characters of the string one on a line.

A screenshot of a computer screen

Description automatically generated

7] Write a program that declares a **6x6** array (matrix) of integers, and initializes it with the pattern shown, with Fibonacci numbers.

Start with two '1's. Each Fibonacci number is the sum of the two previous numbers.



Hint: Use two variables and after each iteration, update them!

Hint: f=f1+f2; f1 = f2; f2=f;

A screenshot of a computer

Description automatically generated