**Baku Higher Oil School**

**Process Automation Engineering Department**

**Programming in C**

**Laboratory 5 – Pointer**

**P.S** Add comment for each task; submit the file in LMS before the deadline.

1. Find the error in each of the following program segments. Assume

**int** \*zPtr; // zPtr will reference array z

**int** \*aPtr = **NULL;**

**void** \*sPtr = **NULL;**

**int** number;

**int** z[**5**] = {**1**, **2**, **3**, **4**, **5**};

sPtr = z;

1. ++zptr;
2. // use pointer to get first value of array; assume zPtr is initialized

number = zPtr;

1. // assign array element 2 (the value 3) to number;

assume zPtr is initialized

number = \*zPtr[**2**];

1. // print entire array z; assume zPtr is initialized

**for** (size\_t i = **0**; i <= **5**; ++i) {

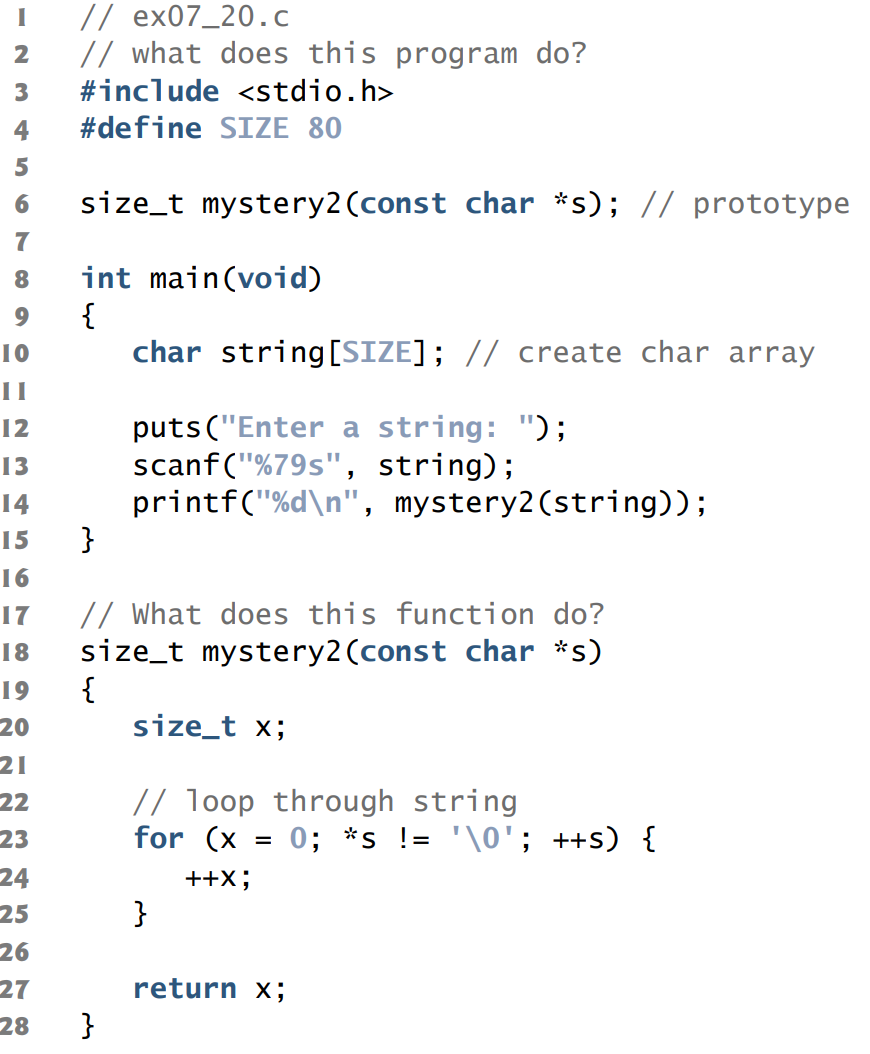
printf(**"%d "**, zPtr[i]);

}

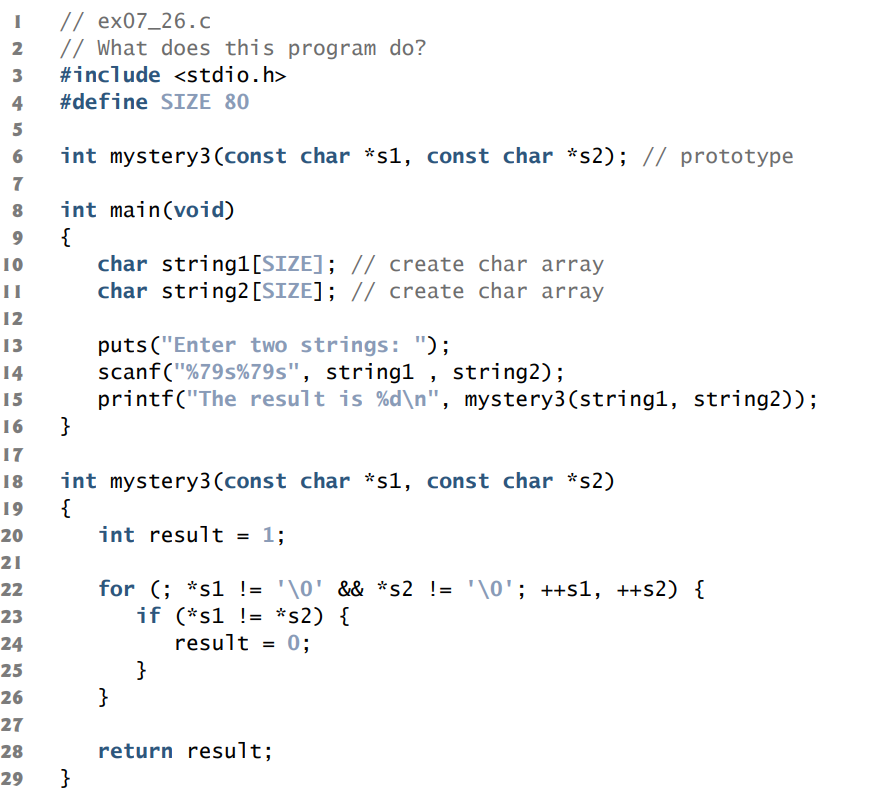
1. // assign the value pointed to by sPtr to number

number = \*sPtr;

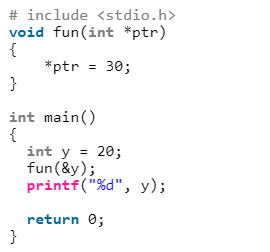
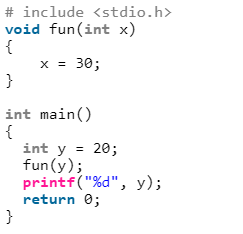
1. ++z;
2. int \*number;  
    printf("%d\n", \*number);
3. float \*realPtr;   
   long \*integerPtr;   
   integerPtr = realPtr;
4. int \* x, y;   
   x = y;
5. char s[] = "this is a character array";   
   int count;   
   for (; \*s != '\0'; ++s)   
   printf("%c ", \*s);
6. short \*numPtr, result;   
   void \*genericPtr = numPtr;   
   result = \*genericPtr + 7;
7. float x = 19.34;   
   float xPtr = &x;   
   printf("%f\n", xPtr);
8. char \*s;  
   printf("%s\n", s);
9. What does this program do?

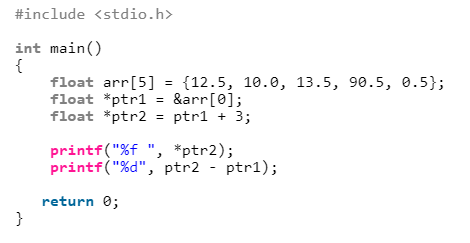
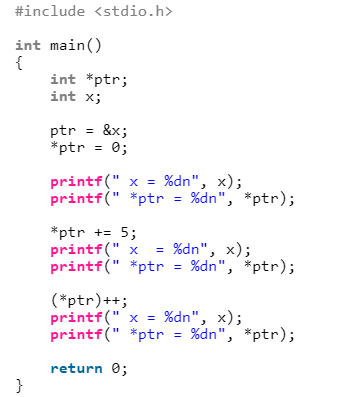


1. What does this program do, assuming that the user enters two strings of the same length?



1. What is the output of following program?

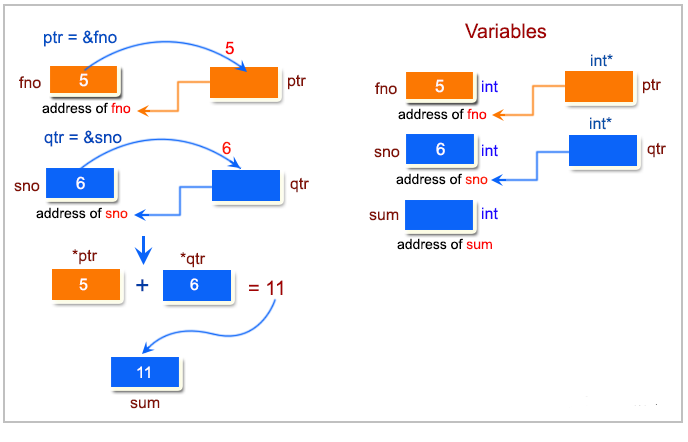




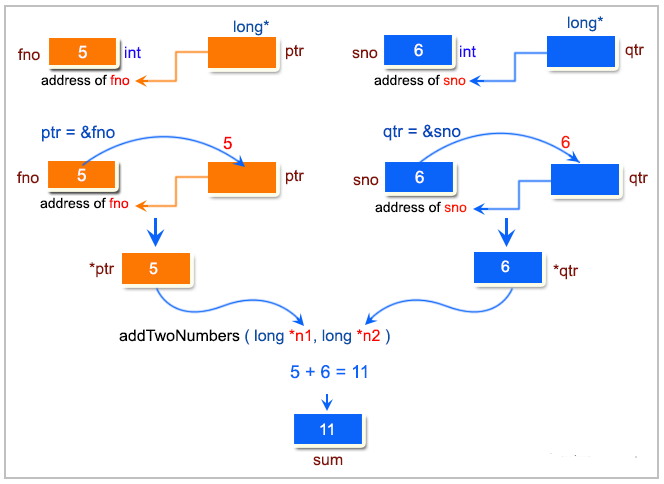
1. For each of the following, write a statement that performs the indicated task. Assume that floating-point variables *number1* and *number2* are defined and that number1 is initialized to 7.3.
2. Define the variable *fPtr* to be a pointer to an object of type float.
3. Assign the address of variable *number1* to pointer variable *fPtr*.
4. Print the value of the object pointed to by *fPtr*.
5. Assign the value of the object pointed to by *fPtr* to variable *number2*.
6. Print the value of *number2*.
7. Print the address of *number1*. Use the %p conversion specifier.
8. Print the address stored in *fPtr*. Use the %p conversion specifier. Is the value printed the same as the address of *number1*?
9. Write Answer each of the following. Assume that single-precision floating-point numbers are stored in 4 bytes, and that the starting address of the array is at location 1002500 in memory. Each part of the exercise should use the results of previous parts where appropriate.
10. Define an array of type float called numbers with 10 elements, and initialize the elements to the values 0.0, 1.1, 2.2, …, 9.9. Assume the symbolic constant SIZE has been defined as 10.
11. Define a pointer, nPtr, that points to an object of type float.
12. Print the elements of array numbers using array index notation. Use a for statement. Print each number with 1 position of precision to the right of the decimal point.
13. Give two separate statements that assign the starting address of array numbers to the pointer variable nPtr.
14. Print the elements of array numbers using pointer/offset notation with the pointer nPtr.
15. Print the elements of array numbers using pointer/offset notation with the array name as the pointer.
16. Print the elements of array numbers by indexing pointer nPtr.
17. Refer to element 4 of array numbers using array index notation, pointer/offset notation with the array name as the pointer, pointer index notation with nPtr and pointer/offset notation with nPtr.
18. Assuming that nPtr points to the beginning of array numbers, what address is referenced by nPtr + 8? What value is stored at that location?
19. Assuming that nPtr points to numbers[5], what address is referenced by nPtr –= 4?

What’s the value stored at that location?

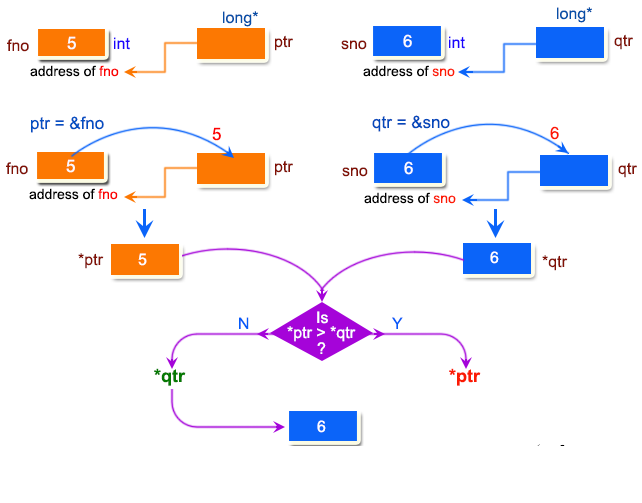
1. Write a program to add two numbers using pointers.



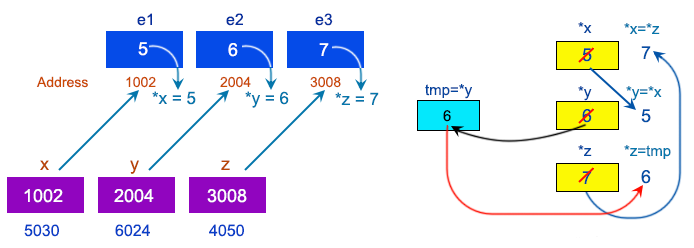
1. Write a function to add numbers using call by reference.



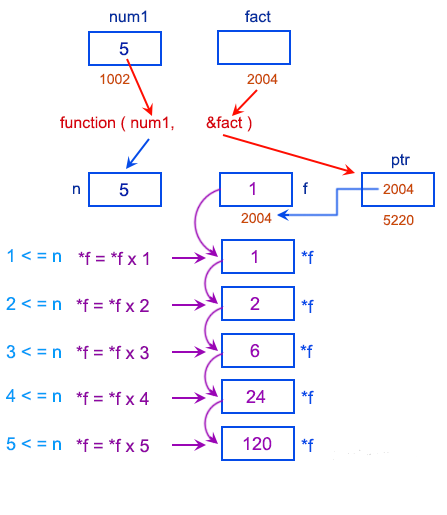
1. Write a program to find the maximum number between two numbers using a pointer.



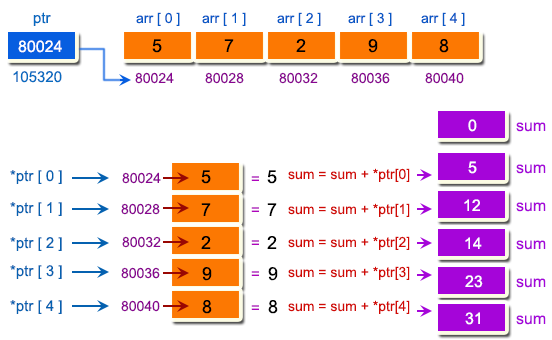
1. Write a program that swap two elements using call by reference.



1. Write a program that find the factorial of a given number using pointers.



1. Write a program to compute the sum of all elements in an array using pointers.



1. Write a program that print the elements of an array in reverse order using pointers.

