**Problem 19.1**

Complete the following tasks using pointers.

Assume

int x = 20, y = 80, arr[3] = {2, 6, 15};

int\* ptr1 = &x, \*ptr2 = &y;

int\* ptr3 = arr, \*ptr4 = &arr[1];

1. Print \*ptr1+1 and \*(ptr1+1)
2. Copy ptr1 to ptr2, \*ptr2 = 0, printf(“%d, %d, %d, %d \n”, x, y, \*ptr1, \*ptr2);
3. ptr1++; printf(“%x, %d\n”, ptr1, \*ptr1);
4. ptr3+=2; printf(“%x, %d\n”, ptr3, \*ptr3);
5. ptr4-=1; printf(“%x, %d\n”, ptr4, \*ptr4);
6. printf(“%x, %d\n”, ptr4 – ptr3, \*ptr4 – \*ptr3);
7. printf(“%x, %d\n”, ptr3 + ptr4, \*ptr3 + \*ptr4);
8. printf(“%x, %d\n”, ptr3[1], ptr4[1]);
9. printf(“%x, %d\n”, \*(ptr3+1), \*(ptr4+1));
10. printf(“%d, %d, %d\n”, ptr3>ptr4, ptr3==ptr4, ptr3<ptr4);
11. printf(“%x, %d\n”, \*(arr+1), \*(ptr3+1));
12. printf(“%x, %d\n”, arr[1], ptr3[1]);
13. \*(ptr3+1) = -1; printf(“%x, %d\n”, arr[1], ptr3[1]);
14. ptr3[1] = -1; printf(“%x, %d\n”, arr[1], ptr3[1]);

**Problem 19.2**

Write down a function called doubleIt, whose function is to simply multiply every value of the array passed to it with 2. Call this function in main and print the values of the array before and after calling the function to verify that arrays are always passed by reference.

**Problem 19.3**

Hint: Accessing array elements using pointer/offset notation: point the pointer to starting address of array, then use \*(ptr +i) to access the value of ith element of the array.

Complete the following tasks using pointer notation to access array.

1. Input an array using pointer/offset notation and then print the array using pointer/offset notation.
2. Input an array using pointer/offset notation and print the array in reverse using pointer/offset notation.

**Home Task**

1. Implement the quadratic roots program using call by reference, i.e. Write down a function called computeRoots(int a, int b, int c, int \*x1, int \*x2), which should compute the two roots using quadratic equation and return their values in x1 and x2, which are passed by reference.
2. Implement the array linear search algorithm in a function called linearSearchUsingPointer using pointer/offset notation, i.e. the function should print whether a key value is found inside the array or not, but this should be done using pointer/offset notation.