

## DSA

### Week-1-Exercise

1. Write a function to swap two integers. The function should take two pointer arguments. Display the values before and after the swap.
2. Create a simple function `print_addr(int x)` whose sole purpose is to print the address of the integer `x` passed to it. Create an integer variable in main, print out its address, and then pass that variable to `print_addr`.
3. Create a function `new_integer()` that declares and initializes an integer inside the function and returns the address of that integer. Print out the integer value associated with this memory address in main.
4. Write a program in C to store `n` elements in an array and print the elements using pointer.
5. Write a program in C to find the factorial of a given number using pointers.
6. What will be the output of the following program:

Main()

```
{
int a, *b, **c, ***d, ****e ;
a=10;
b=&a;
c=&b;
d=&c;
e=&d;
printf( "\na = %d b = %u c = %u d = %u e = %u", a, b, c, d, e ) ;
printf( "\n%d%d %d", a, a + *b, **c + ***d + ****e);
}
```

7. Write a program in C to compute the sum of all elements in an array using pointers.
8. Write a C program to search an element in array using pointers.
9. Write a C program to find the max and min of an integral data set. The program will ask the user to input the number of data values in the set and each value. The program prints on screen a pointer that points to the max value.
10. Give the value of the left-hand side variable in each assignment statement. Assume the lines are executed sequentially. Assume the address of the `blocks` array is 4434.

```
int main()
{
char blocks[3] = {'A','B','C'};
char *ptr = &blocks[0];
char temp;

temp = blocks[0];
temp = *(blocks + 2);
temp = *(ptr + 1);
temp = *ptr;

ptr = blocks + 1;
temp = *ptr;
temp = *(ptr + 1);

ptr = blocks;
temp = ++*ptr;
temp = ++*ptr;
temp = *ptr++;
```

```
temp = *ptr;  
return 0;  
}
```

11. Develop an a program for the following:

1. Get the list of n numbers.
2. Scan and print the odd numbers along its positions while you scan the numbers from 1 to n.
3. Scan and print the even numbers along its positions while you scan the numbers from n to 1.