Practice 8 – User Defined Functions

Tasks 1: Write functions:

A. to print first 50 positive even numbers
B. to print any character n times (function has two parameters character and count)
   #function to print character c, n times
   `def print_char(c, n):`
C. to print counting first n integers with distance d
   #function to print n integers 1, 2, 3 or 1, 3, 5 etc., distance between integers is parameter d
   `def print_counting(n, d):`
D. to find and return sum of first n integers
E. to find next vowel character
   #function will run a loop and find next vowel character
   `def next_vowel(c):` # if c is A To D, the result will be E. if c is E to H, the result will be I

Task 2: Create following functions:

A. print rectangular box according to rows and columns

Sample Run:

```
**********                          for function call (5, 10)
*        *
*        *
*        *
**********
```

B. print k multiples of n. Function has two parameters n & k. The function will print first k multiples of n

Sample Run:

```
3 6 9 12 15 18 21                   for function call (3, 7)
7 14 21 28                          for function call (7, 4)
```

C. print n random values in range a to b. Function has three parameters. The first parameter is n; whereas second and third parameters are a & b. The function will print n random numbers in range a to b. For example, in the first sample run, there are five random numbers in range 10 to 50. In the second sample run, there are three random numbers in the range 200 to 500.

Sample Run:

```
34 46 22 37 29                      for function call (5, 10, 50)
354 467 282                         for function call (3, 200, 500)
```

D. print quadratic roots in floating point. Recall, we did a program of quadratic roots in earlier classes. Later, we have repeated, when we have discussed if-else statements.

Sample Run:

```
x1=1.366025   x2=-0.366025          for function call (-2, 2, 1)
x1=1.0        x2=-0.666667          for function call (3, 5, 2)
```

Task 3: Create following functions:

A. return middle value. The function has three parameters in any order. The function has to return the middle value.

Sample Run:

```
5                                   for function call (3, 8, 5)
6                                   for function call (6, 8, 5)
8                                   for function call (9, 8, 5)
```

B. return average value. The function has three parameters. The function has to return the average in floating point.

Sample Run:

```
5.33333                                 for function call (3, 8, 5)
6.33333                                 for function call (6, 8, 5)
8.0                                     for function call (9, 8, 7)
```

C. return xy. The function has two parameters x & y. The function will calculate and return x times y. For example 25 = 32, 34 = 81

Sample Run:

```
64                                  for function call (2, 6)
125                                 for function call (5, 3)
```

D. return the next vowel. The function has only one parameter: a character. The function will run a loop and return the next vowel. In case usersends 'c', the function will return 'e'. If the user sends 'm', the function will return 'o'. If user will send 'w', the function will return 'a'

Sample Run:

```
i                                   for function call ('e')
u                                   for function call ('s')
a                                   for function call ('x')
```