1. The following **recursive function** in C language computes the factorial of a number. For example fact(5) = 5*4*3*2*1 = 120. Fill in the blanks to make this recursive function working. **(marks: 1.5)**

```
int fact(int num) {
          if(___(i)________) {
               return __________(ii)_____________;
         } else num;

}
```

2. The following function is supposed to reverse a given integer number. Fill in the code to enable the function to do that. **(marks: 2)**
   Note: Do not declare any new variable. Write only 2 statements to complete the code.

```
int reverse(int num) {
            int rev= 0;
            while (num > 0) {
                _______(i)_________;
                _______(ii)_________;
           }
           return rev;
      }
```

3. Complete the following function to print the number triangle
num_triangle(4) gives

```
   1
  121
 12321
1234321
```

(marks: 2)

```
void num_triangle(int range) {
                for(int i=1; i<=range; i++) {
                     for(________(i)_________)
                         printf(" ");
```

```
                    for(________(ii)_________)
                         printf("%d", j);

                   for(________(iii)_________)
                         printf("%d", k);

        printf("\n");
    }
}
```

4. Write a function in C language to simulate a biased coin (will get head with probability p and tail with probability 1-p). Let the function be `int tossBiasedCoin(double p)` which returns 1 if it is head and 0 if it is tail. Assume that you are given a function rand() which returns a number between 0 and 1 (precision: 1 decimal place) with equal probability. **(marks 3)**

5. Write a C function to find node in a graph with maximum degree and return the degree of that node.

```
#define MAX_NODES 100
int findMaxDegree(int adjMatrix[MAX_NODES][MAX_NODES], int numNodes)
```

This function takes the adjacency matrix and the number of nodes as argument and returns the maximum degree. **(marks 1.5)**