

**International Institute of Information Technology, Hyderabad**  
**ICS101:M-16:Computer Programming**  
**Final Exam**

Max. Points: 170

[Time: 180 Mins]

~~This question paper has 11 questions. Please check both sides and make sure that your question paper is complete.~~

**Part I:** Give brief and precise answers to the following.

[5 × 10 = 50]

1. Give the address-related operators that are used in dealing with pointers (not pointer arithmetic). Explain each with a simple example.
2. Explain the meaning of `#ifndef` statement, its processing, and use in the context of multi-file programs.
3. You are given a character array: `char line[128];`, where every character is assigned a non-zero value. What does the following code do for a given integer,  $k$ ?  

```
int *i = (int *)(&line[k]); printf("%d", *i);
```
4. Explain the difference between *declaration* and *definition* in the context of functions, and how they are relevant to the processes of compilation and linking.
5. Explain with the help of an example, the usage of the keyword: `typedef`.

**Part II:** Write functions to solve the following problems with comments where required. You should write your logic in sentences before writing the function itself

[6 × 20 = 120]

1. Given a positive integer, `num`, write a function that computes the number of unique prime factors of the number. Note that the return value is the *number of prime factors* and not the factors themselves. e.g., if `num` is 12, the return value should be 2, as the unique prime factors of 12 are: {2, 3}.  

```
int uniquePrimeFactors(int num)
```
2. You are given a 2-d array of integers: `int A[M][N];`. Complete the function below that finds the size,  $k$ , of the *largest symmetric square* sub array,  $k \times k$  ( $k < M, N$ ). Divide your code into appropriate functions.  

```
int findSymmetric(int array[][N], int M, int N);
```
3. Write a function to print all the unique permutations of a set of integers given in an array. Note that the array might contain multiple entries of the same number.  

```
void printPermutations(int array[], int N);
```
4. Write a program that takes an integer of two digits or less as a command line argument and prints it in words. For e.g., 15 should be printed as *fifteen*, 37 as *thirty seven* and 8 as *eight*. If the command line argument is incorrect, you should print a helpful message.

5. You are given a pointer to a node in a circular linked list of integers. Write the function that computes the number of repeating integers in the circular linked list: (`int countRepetitions(Node *n);`). For e.g., if List is: {2, 18, 4, 2, 591, 27, 3, 2, 18, 4, 2, 62, 2}; Output is: 3.

The *Node* structure is defined as:

```
typedef struct node{  
    int data;  
    struct node *next;  
} Node;
```

6. Write a program that reads a list of numbers from a file (*data.txt*), creates a dynamic array to store the numbers, dynamically resize the array if it runs out of space, and then sorts the numbers in the array. Note that the file (*data.txt*), only contains the numbers and not the count of numbers, so you need to read the data until the end of file.