PDS Lab (Assignment – 1)
Date: 31st July, 2017 (2-6 pm)

(1) Write a C program to print your name.

(2) Write a C program to convert the temparature from Fahrenheit to Celsius. Take the temparature in Fahrenheit as imput from the keyboard, and print the temparature in Celsius.

(3) Write a C program to find the average of the given 3 numbers. Your program need to accept these 3 numbers from key board, and print the 3 numbers and their average as the output.

(4) Write a C program to swap two given numbers. You need to assign the input numbers (from the keyboard) to two variables “a” and “b”. You need to perform the swapping of the values associated to the variables and print the variables before and after swapping.

(5) Write a C program to compute the sum of the squares of the first “N” natural numbers. You need to provide the value of N, through keyboard and print the output as sum of the squares of the N natural numbers.

(6) Write a C program to calculate the area of the sphere. Take the radius as imput from the keyboard, and print the area of the sphere as output.

(7) Write a C program to compute the distance between given 2 points and find the slope of the line joining these 2 points. Take (x,y) coordinates of the 2 points from key board and print the length of the line joining those 2 points and slope of the line.

(8) Write a C program to carry out the following matrix operations on 2x2 matrices: (i) addition (A+B), (ii) subtraction (A-B) and (iii) multiplication (AxB). The program should read the elements of the matrices as A = {a11, a12, a21, a22} and B = {b11, b12, b21, b22}. The program should print both input and output in matrix notation.

(9) Write a C program to enter the 2 angles of a triangle and find the 3rd angle.

(10) From the given Cartesian coordinates, calculate the area and perimeter of the following regular geometric shapes.
- (a) Circle with centre (3,1) and cutting the Y-axis at (0,5) and (0, -3)
- (b) Triangle with vertices (-2,3), (-3,-4), (5,1)
- (c) Square with vertices (0,0), (1,1), (2,0), (1,-1)
- (d) Rhombus with vertices (3,5), (7,6), (2,1), (6,2)
- (e) Trapezoid with vertices (-3,3), (1,5), (4,1), (1, -5)

(11) Consider a bank that offers fixed deposit accounts with cumulative (annually) interest on the balance available in the account. Write a C program that reads the amount initially invested (called Principal amount) in an account and interest rate. The program generates the balance available in the account at the end of each year for first five years.

(12) Write a C program to print the given pattern using printf statements.

PDS Lab (Assignment – 2)
Date: 7th August, 2017 (2-6 pm)

(1) Write a C program that will print your mailing address in the following format. The fields need to be accepted through keyboard.
<Name> (K. Sreenivasa Rao)
<Quarter No>, <Street Name> (C1-87, Tech market road)
<City>, <Pincode> (Kharagpur, 721302)

(2) Write a C program to display the multiplication table (with 10 rows). The user has to specify the desired multiplication table through keyboard.
6 X 1 = 6
6 X 2 = 12
.
.
.
6 X 10 = 60

(3) Write a C program to generate output of AND and OR operations using 3 variables. User need to provide these variables through keyboard. The output has to be displayed as per the following format:
Input: a, b, c

Output: a	b	c	a || b || c
4	0	4	1
a	b	c	a && b && c
4	0	4	0

(4) Write a C program to declare an integer and floating point number. Assign the values to the variables through the keyboard and print the value of the variables and its address. Print the floating point value with different precisions as shown below:
Input: X = 12345.872345
Output: X = 123; X = 12345; X = 12345.87; X = 12345.872345

(5) Write a C program to calculate the area of the sphere. Take the radius as input from the keyboard, and print the area of the sphere as output. Note: Define the value of PI as floating point constant.

(6) Write a C program to evaluate the following expression: P = ++X + Y-- * --Z / 4 % 2 - -1. Take the values of X, Y, Z through keyboard. The program should print the outcome of each step based on the priority of the operators. Print the values of X, Y, Z at each step.

(7) Write a C program to compute the roots of quadratic equation of the form $ax^2 + bx + c = 0$. Where a, b, c are real numbers. Hint: use math function for evaluating square root function. If the discriminant function (b^2 - 4ac) is less than zero, print the roots are imaginary. For checking this condition, you may use ternary operator.

(8) Write a C program to swap two given numbers. You need to assign the input numbers (from the keyboard) to two variables "a" and "b". You need to perform the swapping of the values associated to the variables and print the variables before and after swapping.

(9) Write a C program to compute the sum of the squares of the first "N" natural numbers. You need to provide the value of N, through keyboard and print the output as sum of the squares of the N natural numbers.

(10) Write a C program to compute the distance between given 2 points and find the slope of the line joining these 2 points. Take (x,y) coordinates of the 2 points from key board and print the length of the line joining those 2 points and slope of the line.

(11) Write a C program to carry out the following matrix operations on 2x2 matrices: (i) addition (A+B), (ii) subtraction (A-B) and (iii) multiplication (AxB). The program should read the elements of the matrices as A = {a11, a12, a21, a22} and B = {b11, b12, b21, b22}. The program should print both input and output in matrix notation.

(12) Write a C program to enter the 2 angles of a triangle and find the 3rd angle.

(13) From the given Cartesian coordinates, calculate the area and perimeter of the following regular geometric shapes.
(a) Circle with centre (3,1) and cutting the Y-axis at (0,5) and (0, -3)
(b) Triangle with vertices (-2,3), (-3,-4), (5,1)
(c) Square with vertices (0,0), (1,1), (2,0), (1,-1)
(d) Rhombus with vertices (3,5), (7,6), (2,1), (6,2)
(e) Trapezoid with vertices (-3,3), (1,5), (4,1), (1, -5)

(14) Consider a bank that offers fixed deposit accounts with cumulative (annually) interest on the balance available in the account. Write a C program that reads the amount initially invested (called Principal amount) in an account and interest rate. The program generates the balance available in the account at the end of each year for first five years.

(15) Write a C program to print the given pattern using printf statements.

PDS LAB – 3 (Section-13) Date: 14th August 2017
Decision Making & Branching (if-else, switch, goto, break, continue)

1. Write a C program to compute the sum of the squares of 1st 'N' natural numbers. Read the value of 'N' from the key board.

2. Write a C program to determine the number of positive and negative numbers entered through keyboard. The key-board entry will be terminated by 0.

3. Write a C program to determine whether the entered number is a palindrome or not.

4. Write a C program to compute the second largest from the given sequence of numbers through key board.

5. Write a C program to display the awarded grade of a student in a particular course based on the marks. The grading will be carried out based on the following criteria:

Marks	Grade
>= 90	E
90 <= && >= 80	A
80 <= && >= 70	B
70 <= && >= 60	C
60 <= && >= 50	D
50 <= && >= 40	P
40 <	F

   Use the conditional operator (? :) for solving this problem.

6. Write a C program to compute the sum of the square roots of the numbers entered through key board. First enter the number of values (N) to be entered through key board, and then enter values one by one.
   a. If negative number encounters, skip the computation of square root and sum. Continue to accept the next number until you reach N.
   b. If negative number encounters, break the entire process (computation of square root, sum and entry of other numbers) and print the sum that was accumulated till that point.

7. Write a C program that reads two positive real numbers. Now, enter an operator from the keyboard and perform the operation. The operator can be any of the following:
   + (addition)
   - (subtraction)
   * (multiplication)
   / (division) and
   % (remainder)

If the operator is not the valid one, print the same message. Otherwise print the expression as follows: num1 OP num2 = result (ex: x + y = z, x – y = z, …..).

8. Write a C program to test whether the given number (consider a max of 999999) is divisible by 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12. (use the first principles of division)

Divisibility by	Rule
2	Last digit (digit in one's place) should be even number
3	Sum of all digits of the number to be divisible by 3
4	The number formed by last 2 digits (one's and ten's place) should be divisible by 4
5	Last digit (digit in one's place) of the number should be either 0 or 5.
6	The number should be divisible by both 2 and 3
7	Double the last digit and subtract it from the remaining leading truncated number. If the result is divisible by 7, then so was the original number. Apply this rule over and over again as necessary.
8	If hundred's place is even check whether the number formed by last 2 digits is divisible by 4. If hundred's place is odd, add 4 to the number formed by last 2 digits and check whether it is divisible by 4.
9	Sum of all digits of the number to be divisible by 9
10	Last digit should be zero
11	Difference of sum of even and sum of odd digits should be divisible by 11.
12	The number should be divisible by both 3 and 4

PDS LAB – 3 (Section-13) Date: 14th August 2017
Decision Making & Branching (if-else, switch, goto, break)

1. Write a C program to compute the sum of the squares of 1st 'N' natural numbers. Read the value of 'N' from the key board.

2. Write a C program to determine the number of positive and negative numbers entered through keyboard. The key-board entry will be terminated by 0.

3. Write a C program to hat reads a positive real number *x*, computes the fractional part of *x*, and also the rounded value of *x* (to the nearest integer). Your program may not make any math library calls.

4. Write a C program to determine whether the entered number is a palindrome or not.

5. Write a C program to compute the second largest from the given sequence of numbers through key board.

6. Write a C program to display the awarded grade of a student in a particular course based on the marks. The grading will be carried out based on the following criteria:

Marks	**Grad**
>= 90	E
90 <= && >=	A
80 <= && >=	B
70 <= && >=	C
60 <= && >=	D
50 <= && >=	P
40 <	F

   Use the conditional operator (? :) for solving this problem.

7. Write a C program to compute the sum of the square roots of the numbers entered through key board. First enter the number of values (N) to be entered through key board, and then enter values one by one.
   a. If negative number encounters, skip the computation of square root and sum. Continue to accept the next number until you reach N.
   b. If negative number encounters, break the entire process (computation of square root, sum and entry of other numbers) and print the sum that was accumulated till that point.

8. Write a C program that reads two positive real numbers. Now, enter an operator from the keyboard and perform the operation. The operator can be any of the following:
   + (addition)
   - (subtraction)
   * (multiplication)
   / (division) and
   % (remainder)

If the operator is not the valid one, print the same message. Otherwise print the expression as follows: num1 OP num2 = result (ex: x + y = z, x – y = z, …..).

9. Write a C program to print the salary of an IITKGP employee. Consider 2 types of employees: Faculty (F or f) and Staff (S or s). Within the faculty there are 3 divisions (Assistant Professor, Associate Professor and Professor) and inside staff there are 3 divisions (Assistant Registrar, Deputy Registrar, and Registrar). Salaries of the employees are calculated as follows:

Employee	**Salary**
Assistant Professor	100000*(1+number of years/10)
Associate Professor	100000*1.5*(1+number of years/10)
Professor	100000*2*(1+number of years/10)
Assistant Registrar	100000*0.4*(1+number of years/10)
Deputy Registrar	100000*0.8*(1+number of years/10)
Registrar	100000*1.2*(1+number of years/10)

(Hint: Use nested switch statements)

10. Write a C program to test whether the given number (consider a max of 999999) is divisible by 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12. (use the first principles of division)

Divisibility by	Rule
2	Last digit (digit in one's place) should be even number
3	Sum of all digits of the number to be divisible by 3
4	The number formed by last 2 digits (one's and ten's place) should be divisible by 4
5	Last digit (digit in one's place) of the number should be either 0 or 5.
6	The number should be divisible by both 2 and 3
7	Double the last digit and subtract it from the remaining leading truncated number. If the result is divisible by 7, then so was the original number. Apply this rule over and over again as necessary.
8	If hundred's place is even check whether the number formed by last 2 digits is divisible by 4. If hundred's place is odd, add 4 to the number formed by last 2 digits and check whether it is divisible by 4.
9	Sum of all digits of the number to be divisible by 9
10	Last digit should be zero
11	Difference of sum of even and sum of odd digits should be divisible by 11.
12	The number should be divisible by both 3 and 4

PDS LAB – 3 (Section-13) Date: 14th August 2017
Decision Making & Branching

1. Write a C program that reads the coordinates of three points in the plane and reports which two of the three points are closest. Do not use math library calls.

2. Write a C program to hat reads a positive real number *x*, computes the fractional part of *x*, and also the rounded value of *x* (to the nearest integer). Your program may not make any math library calls.

3. Write a C program to compute the second largest from the given sequence of numbers through key board.

4. Write a C program to display the awarded grade of a student in a particular course based on the marks. The grading will be carried out based on the following criteria:

Marks	Grade
>= 90	E
90 <= && >= 80	A
80 <= && >= 70	B
70 <= && >= 60	C
60 <= && >= 50	D
50 <= && >= 40	P
40 <	F

   Use the conditional operator (? :) for solving this problem.

5. Write a C program that reads two positive real numbers. Now, enter an operator from the keyboard and perform the operation. The operator can be any of the following:

   + (addition)
   - (subtraction)
   * (multiplication)
   / (division) and
   % (remainder)

   If the operator is not the valid one, print the same message. Otherwise print the expression as follows: num1 OP num2 = result (ex: x + y = z, x – y = z, …..).

6. Write a C program to test whether the given six digit number is divisible by 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12. (use the first principles of division)

Divisibility by	Rule
2	Last digit (digit in one's place) should be even number
3	Sum of all digits of the number to be divisible by 3
4	The number formed by last 2 digits (one's and ten's place) should be

	divisible by 4
5	Last digit (digit in one’s place) of the number should be either 0 or 5.
6	The number should be divisible by both 2 and 3
7	Double the last digit and subtract it from the remaining leading truncated number. If the result is divisible by 7, then so was the original number. Apply this rule over and over again as necessary.
8	If hundred’s place is even check whether the number formed by last 2 digits is divisible by 4. If hundred’s place is odd, add 4 to the number formed by last 2 digits and check whether it is divisible by 4.
9	Sum of all digits of the number to be divisible by 9
10	Last digit should be zero
11	Difference of sum of even and sum of odd digits should be divisible by 11.
12	The number should be divisible by both 3 and 4

PDS LAB – 4 (Section-13) Date: 21st August 2017
Decision Making, Branching & Looping

1. Write a C program to compute the sine of x approximately by summing the first n terms of the infinite series $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$ where *x* is expressed in radians. Compute sin(x)
   (a) Sum the first n terms, where n is a positive integer read through keyboard along with x
   (b) Continue adding successive terms in the series until the value of the next term becomes smaller than 10^{-5} . Test your program for different values of x and display the number of terms used and final answer.

2. Write a C program to display the salary of IITKGP's employee based on his nature of job (faculty, administrative staff and technical staff), designation and years of experience. The inputs to be given to the program are Employee type, Designation, Basic and years of experience. You may need to use switch statements (2-level) to solve this problem.

Employee (type)	**Designation (code)**	**Basic Pay**
Faculty ('F' or 'f')	Assistant Professor (1)	Basic
	Associate Professor (2)	Basic × 1.5
	Professor (3)	Basic × 2
Administrative Staff ('A' or 'a')	Assistant Registrar (1)	Basic × 0.5
	Deputy Registrar (2)	Basic × 0.8
	Registrar (3)	Basic × 1.2
Technical Staff ('T' or 't')	Lab Technician (1)	Basic × 0.35
	System Analyst (2)	Basic × 0.75
	Chief System Analyst (3)	Basic × 1.25
Additional Increment (AI) = 0.1 × Basic × years		
Total Pay = Basic + AI		

3. Write a C program to check whether the given number is a palindrome or not? (Note: A number/string is said to be palindrome if the given number/string and its reversed version are same. Ex: 12321 is a palindrome)

4. Write a C program to check whether the given number is a perfect number or not? (Note: A number is said to be perfect, if the sum of all the factors (except the number) is equal to the number itself. Ex: 6 = 1+2+3).
5. Write a C program to print the following pattern of numbers by giving the number of rows as input:

1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
6 7 8 9 10 11 10 9 8 7 6

6. Write a C program to determine the number of words, number of lower case letters, number of upper case letters and number of digits of a given string. After computing the counts, reverse the input string. Enter the input text through keyboard and terminate it with new line (enter button).

7. Write a C program to convert the given binary number to decimal number and vice versa.
8. Write a C program to print prime numbers below 100.
9. Write a C program to compute the factorial of a given number.
10. Write a C program to print the sequence of Fibonacci numbers up to N.
11. Write a C program to compute the sum of the square roots of the numbers entered through key board. First enter the number of values (N) to be entered through key board, and then enter values one by one.
    (a) If negative number encounters, skip the computation of square root and sum. Continue to accept the next number until you reach N.
    (b) If negative number encounters, break the entire process (computation of square root, sum and entry of other numbers) and print the sum that was accumulated till that point.

PDS LAB – 4 (Section-13) Date: 21st August 2017
Decision Making, Branching & Looping

1. Write a C program to compute the sine of x approximately by summing the first n terms of the infinite series $sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$ where x is expressed in radians. Continue adding successive terms in the series until the value of the next term becomes smaller than 10^{-5}. Test your program for different values of x and display the number of terms and value of sin(x).

2. Write a C program to display the salary of IITKGP's employee based on his nature of job (faculty, administrative staff and technical staff), designation and years of experience. The inputs to be given to the program are Employee type, Designation, Basic and years of experience. You may need to use switch statements (2-level) to solve this problem. Display the input data as well as output in proper format.

Employee (type)	**Designation (code)**	**Basic Pay**
Faculty ('F' or 'f')	Assistant Professor (1)	Basic
	Associate Professor (2)	Basic × 1.5
	Professor (3)	Basic × 2
Administrative Staff ('A' or 'a')	Assistant Registrar (1)	Basic × 0.5
	Deputy Registrar (2)	Basic × 0.8
	Registrar (3)	Basic × 1.2
Technical Staff ('T' or 't')	Lab Technician (1)	Basic × 0.35
	System Analyst (2)	Basic × 0.75
	Chief System Analyst (3)	Basic × 1.25
Additional Increment (AI) = 0.1 × Basic × years		
Total Pay = Basic + AI		

3. Write a C program to find the list of perfect-numbers below a given number N (input through keyboard) (Note: A number is said to be perfect, if the sum of all the factors (except the number) is equal to the number itself. Ex: 6 = 1+2+3).

4. Write a C program to print the following pattern of numbers by giving the number of rows as input:

1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
6 7 8 9 10 11 10 9 8 7 6

5. Write a C program to compute the sum of the square roots of the numbers entered through key board. First enter the number of values (N) to be entered through key board, and then enter values one by one. Incorporate the following two options by appropriately using break and continue statements.

   Option-1: If negative number encounters, skip the computation of square root and sum. Continue to accept the next number until you reach N.

   Option-2: If negative number encounters, break the entire process (computation of square root, sum and entry of other numbers) and print the sum that was accumulated till that point.

Additional Problems for Practice

6. Write a C program to find the list of palindrome-numbers below a given number N (input through keyboard) (Note: A number/string is said to be palindrome if the given number/string and its reversed version are same. Ex: 12321 is a palindrome)

7. Write a C program to convert the given binary number to decimal number and vice versa. At the beginning the user has to select one of the options (say 1 for binary to decimal and 2 for decimal to binary) from the keyboard.

8. Write a C program to print all prime numbers below the given number say N (enter through keyboard). Use the break statement and evaluate the number of computations by displaying them with and without break statement.

PDS LAB – 4 (Section-13) Date: 21st August 2017
Decision Making, Branching & Looping

1. Write a C program to compute the sine of x approximately by summing the first n terms of the infinite series $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$ where *x* is expressed in radians. Continue adding successive terms in the series until the value of the next term becomes smaller than 10^{-5}. Test your program for different values of x and display the number of terms and value of sin(x).

2. Write a C program to display the salary of IITKGP's employee based on his nature of job (faculty, administrative staff and technical staff), designation and years of experience. The inputs to be given to the program are Employee type, Designation, Basic and years of experience. You may need to use switch statements (2-level) to solve this problem.

Employee (type)	**Designation (code)**	**Basic Pay**
Faculty ('F' or 'f')	Assistant Professor (1)	Basic
	Associate Professor (2)	Basic × 1.5
	Professor (3)	Basic × 2
Administrative Staff ('A' or 'a')	Assistant Registrar (1)	Basic × 0.5
	Deputy Registrar (2)	Basic × 0.8
	Registrar (3)	Basic × 1.2
Technical Staff ('T' or 't')	Lab Technician (1)	Basic × 0.35
	System Analyst (2)	Basic × 0.75
	Chief System Analyst (3)	Basic × 1.25
Additional Increment (AI) = 0.1 × Basic × years		
Total Pay = Basic + AI		

3. Write a C program to find the list of palindrome-numbers below a given number N (input through keyboard) (Note: A number/string is said to be palindrome if the given number/string and its reversed version are same. Ex: 12321 is a palindrome)

4. Write a C program to find the list of perfect-numbers below a given number N (input through keyboard) (Note: A number is said to be perfect, if the sum of all the factors (except the number) is equal to the number itself. Ex: 6 = 1+2+3).

5. Write a C program to print the following pattern of numbers by giving the number of rows as input:

```
1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
6 7 8 9 10 11 10 9 8 7 6
```

6. Write a C program to determine the number of words, number of lower case letters, number of upper case letters and number of digits of a given string. After computing the counts, reverse the input string. Enter the input text through keyboard and terminate it with new line (enter button).

7. Write a C program to convert the given binary number to decimal number and vice versa. At the begining the user has to select one of the options (say 1 for binary to decimal and 2 for decimal to binary) from the keyboard.

8. Write a C program to print all prime numbers below the given number say N (enter through keyboard). Use the break statement and evaluate the number of computations by displaying them with and without break statement.

9. Write a C program to compute the sum of the square roots of the numbers entered through key board. First enter the number of values (N) to be entered through key board, and then enter values one by one. Incorporate the following two options by appropriately using break and continue statements.
   Option-1: If negative number encounters, skip the computation of square root and sum. Continue to accept the next number until you reach N.
   Option-2: If negative number encounters, break the entire process (computation of square root, sum and entry of other numbers) and print the sum that was accumulated till that point.

PDS LAB – 5 (Section-13) Date: 28th August 2017
Loops and Arrays (Practice Problems)

1. Write a C program that reads an array A[] of n integers and a separate element X from keyboard. Now add those elements of the array A which are greater than X and print the sum.

2. Write a C program that reads an array of 10 numbers. Find the minimum element of the array and print the minimum as well as all elements of the array.

3. Write a C program that reads the sentence consisting sequence of words. Find the number of words in the sentence and print them one below the other.

4. Write a C program that reads the elements of matrix of order 3X4. Input the elements of a matrix through key board and print the matrix in a conventional format.

5. Write a C program to find the list of palindrome-numbers below a given number N (input through keyboard) (Note: A number/string is said to be palindrome if the given number/string and its reversed version are same. Ex: 12321 is a palindrome)

6. Write a C program to convert the given binary number to decimal number and vice versa. At the beginning the user has to select one of the options (say 1 for binary to decimal and 2 for decimal to binary) from the keyboard.

7. Write a C program to accept input as number of rows, and print the output in the following pattern (Ex: output pattern for 6 rows).

```
        1
      1   1
    1   2   1
  1   3   3   1
1 4   6   4   1
1 5  10  10  5  1
```

PDS LAB – 5 (Section-13) Date: 28th August 2017
Loops and Arrays

1. Write a C program that reads an array of 10 numbers. Find the minimum and maximum elements of the array and print the minimum, maximum and all elements of the array.

2. Write a C program that reads the elements of matrix of order 4X4. Input the elements of a matrix through key board and print the matrix in a conventional format and sum of the diagonal elements.

3. Write a C program to convert the given binary number to decimal number and vice versa. At the beginning the user has to select one of the options (say 1 for binary to decimal and 2 for decimal to binary) from the keyboard.

4. Write a C program to read a sequence of numbers from a key board (terminated by some user defined constraint such as entry of some specific number) and compute (i) the average of the sequence, (ii) number of entered values greater than, less than and equal to the average of the sequence.

5. Write a C program to find all possible sub-arrays whose sum equal to zero.
   Example:
   Input: {4, 2, -3, 1, 6, -2, -1, -1, -2, 5, -2}
   Output: Number of possible sub-arrays = 5.
   S1: {2,-3,1};
   S2: {2,-3,1,6,-2,-1,-1,-2};
   S3: {-3,1,6,-2,-1,-1};
   S4: {6,-2,-1,-1,-2};
   S5: {-1,-2,5,-2}

6. Write a C program to accept input as number of rows, and print the output in the following pattern (Ex: output pattern for 5 rows).

```
    1
   123
  12345
   123
    1
```

7. Write a C program that accepts text from keyboard and print the total number of words, number of unique words and their frequencies.
   Input: IIT Kharagpur IIT Mumbai IIT Hyderabad IISC Bangalore IITG IIM Alhabad IIM Bangalore IIT Ganghinagar
   Output:
   Total Number of words = 15
   Number of Unique words = 10
   Frequency of Occerrence: IIT – 4; Kharagpur – 1; Mumbai – 1; Hyderabad – 1; IISC – 1; Bangalore – 2; IITG – 1; IIM – 2; Alhabad – 1; Ganghinagar – 1;

PDS LAB – 5 (Section-13) Date: 28th August 2017
Loops and Arrays

1. Write a C program that reads an array of 10 numbers. Find the minimum and maximum elements of the array and their positions in the array. Print the entire array, minimum, maximum and positions of maximum and minimum elements in the array.

2. Write a C program that reads the elements of matrix of order 4X4. Input the elements of a matrix through key board and print the matrix in a conventional format and sum of the elements of principal diagonal.

3. Write a C program to read a sequence of numbers from a key board (terminated by some user defined constraint such as entry of some specific number) and compute (i) the average of the sequence, (ii) number of entered values greater than, less than and equal to the average of the sequence.

4. Write a C program to find all possible sub-arrays whose sum equal to zero.
   Example:
   Input: {4, 2, -3, 1, 6, -2, -1, -1, -2, 5, -2}
   Output: Number of possible sub-arrays = 5.
   S1: {2,-3,1};
   S2: {2,-3,1,6,-2,-1,-1,-2};
   S3: {-3,1,6,-2,-1,-1};
   S4: {6,-2,-1,-1,-2};
   S5: {-1,-2,5,-2}

5. Write a C program to accept input as number of rows, and print the output in the following pattern (Ex: output pattern for 5 rows).

```
    1
   123
  12345
   123
    1
```

6. Write a C program that accepts text from keyboard and print the total number of words, number of unique words and their frequencies.
   Input: IIT Kharagpur IIT Mumbai IIT Hyderabad IISC Bangalore IITG IIM Alhabad IIM Bangalore IIT Ganghinagar
   Output:
   Total Number of words = 15
   Number of Unique words = 10
   Frequency of Occerrence: IIT – 4; Kharagpur – 1; Mumbai – 1; Hyderabad – 1; IISC – 1; Bangalore – 2; IITG – 1; IIM – 2; Alhabad – 1; Ganghinagar – 1;

PDS LAB – 6 (Section-13) Date: 4th September 2017
Functions & Recursion

1. Write C functions to (i) reverse the string and (ii) comparison of 2 strings. Through main () function enter the strings and demonstrate the above functions. Also, evaluate whether the given string is palindrome or not, by using the above functions.

2. Write C functions to (i) read the elements of a matrix, (ii) standard deviation of the elements of a matrix, and (iii) transpose of a matrix. Through main () function enter the elements of a matrix, compute their standard deviation and print the transpose of a matrix.

3. Write a C function to determine cos(x) using the following expression $\left(1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots\right)$. In this problem you are required to use recursive calls of cos function, factorial function and power function. Through main () function provide the value of x in 'degrees' and number of terms to be considered for computation. Demonstrate the C function by calling through main program.

4. Consider a three dimensional vector space over integer. Write C functions to perform following vector operations.
   i) Dot product of two vectors.
   ii) Cross product of two vectors.
   Read a set of vectors (maximum 10) and using the functions thus created, find how many of these vectors are perpendicular and parallel. Also find the maximum area of the parallelogram that can be formed by two vectors from the set of vectors read.

5. Write a C program to determine the LCM of 2 given numbers. Use co-prime function to verify the 2 number or co-prime or not. Procedure to compute LCM is successively divide the 2 numbers by a common divisor till the numbers become co-prime. At that point multiply all common divisors and left-over numbers.

   Example:
   Input: n1 = 24, n2 = 40; (Co-prime: No)
   Divide by 2: 12 and 20 (Co-prime: No)
   Divide by 2: 6 and 10 (Co-prime: No)
   Divide by 2: 3 and 5 (Co-prime: Yes)
   LCM = 2 × 2 × 2 × 3 × 5 = 120.

6. Write C functions to retrieve (i) the name of the state given the city name and (ii) the names of the cities given the name of the state. Through main () function enter the names of the states and their respective cities and call the above functions to demonstrate the retrieval process.

State	**City**			
Andhra Pradesh	Vijayawada	Guntur	Vizag	Tirupati
Tamil Nadu	Chennai	Madurai	Coimbatore	Tiruchi
Karnataka	Bangalore	Mysore	Beedar	Mangalore
Uttar Pradesh	Kanpur	Luknow	Allahabad	Banaras

PDS LAB – 6 (Section-13) Date: 4th September 2017
Functions and Recursion (Practice Problems)

1. Write C functions to (i) find the length of the string and (ii) reverse the string. Through main () function enter the string using keyboard and demonstrate the above functions.

2. Write C functions to (i) read the elements of a matrix and (ii) sum of 2 matrices. Through main () function enter two matrices and print their sum using the above written functions.

3. Read any two positive numbers *a* and *b*. Write a recursive function *int gcd(int a, int b),* which would return the greatest common divisor of *a* and *b*. You should read the numbers *a* and *b* in the main function and call *gcd(a,b)* in the main.

4. Consider a three dimensional vector space over integer. Write C functions to perform following vector operations.
   (a) Dot product of two vectors.
   (b) Cross product of two vectors.

5. Write a C functions to compute x^n and $n!$ (factorial of *n*) using recursive calls. Demonstrate its applicability through main () function.

PDS LAB – 7 (Section-13) Date: 9th October 2017
Recursion & Structures

1. Write a C function (recursion) to input the elements of an array through key board.

2. Write a C function (recursion) to compute the sum of the elements in an array and return the sum to the main function.

3. Write a C function (recursion) to merge two sorted arrays into a single sorted array.

4. Write a C program to input a student record which contains the following details: {int roll, char name[20], int age}. Define the structure to represent the details of the student, input the attributes of the student record through keyboard and print the same.

5. Modify the above program to enter and print multiple student records say "n" taken from the keyboard.
   Example Output:
   Student [1] = 25 Krish 18
   Student [2] = 28 Ram 19
   Student [3] = 22 Satish 17

6. Write a C program to read (input) multiple student records from keyboard. Write C functions separately for sorting the student records based on each attribute. Demonstrate the above functions by printing the attribute-based sorted records.
   Example Output:

Original student records (before sorting)	*Sorted records based on student roll*
Student [1] = 22 Satish 17	*Student [1] = 22 Satish 17*
Student [2] = 28 Ram 19	*Student [2] = 25 Krish 18*
Student [3] = 25 Krish 18	*Student [3] = 28 Ram 19*
Sorted records based on student name	*Sorted records based on student age*
Student [1] = 25 Krish 18	*Student [1] = 22 Satish 17*
Student [2] = 28 Ram 19	*Student [2] = 25 Krish 18*
Student [3] = 22 Satish 17	*Student [3] = 28 Ram 19*

7. Write a C program to read (input) multiple student records from keyboard. The attributes of the student record consists of {int roll, char name[20], int age, struct marks M}, marks {float sub1, float sub2, float sub3, float sub4, float total, char result}. Enter the number of students' records (say "n") through keyboard. Then call a function to enter the attributes of the students records. Write a C function to sort the records based on total marks and provide the grade as per IITKGP grading scheme.
   Example Output:

Original student records (before sorting)										
Student [1]	=	22	Satish	17	66	70	87	74	297	B
Student [2]	=	28	Ram	19	55	90	99	95	339	A
Student [3]	=	25	Krish	18	22	33	56	90	201	D

Sorted records based on student total marks										
Student [1]	=	25	Krish	18	22	33	56	90	201	D
Student [2]	=	22	Satish	17	66	70	87	74	297	B
Student [3]	=	28	Ram	19	55	90	99	95	339	A

8. Write C program to compute the interest and total amount (principle + interest) for all the account holders of a bank. The details of the account holder are as follows: {int acc_no, char name[20], float principle, struct date deposit_date, float interest, float total}. The attributes of Date are {int date, int month, int year}. Define appropriate structures and input the number of account holders, rate of interest through keyboard. Write C functions to (i) enter the details of account holders, (ii) compute the difference between deposit date and target date, (iii) compute the interest and total amount for each account holder and (iv) print the records in systematic manner.

PDS LAB – 7 (Section-13) Date: 17th October 2017
Recursion & Structures (Supplementary)

1. Expression to evaluate binomial coefficient is as follows:

$$\binom{n}{r} = \begin{cases} 1, if\ r = 0 \\ 1, if\ n = r \\ \binom{n-1}{r} + \binom{n-1}{r-1} Otherwise \end{cases}$$

   Write a C function to compute binomial coefficient using recursive calls. Through main function evaluate $\binom{n}{r}$ by inputting the values of n and r through key board. Also analyse the number of calls to evaluate $\binom{n}{r}$ and repetitions.

2. Write a Function to recursively compute the harmonic mean of an array of numbers. In the main function, create an array of size 10. Input integers from the user till a negative number is given as input or the 10 elements have been filled up. Find the harmonic mean of the elements of this array.

3. Write a function to recursively compute the sum of digits of a positive integer. The function has to be recursive. Function Protype : int sum(int n)

4. A rectangle in the X-Y plane can be specified by eight real numbers representing the coordinates of its four corners. Define a structure to represent a rectangle using eight double variables. Notice that here we do not assume the sides of a rectangle to be necessarily parallel to the X and Y axes. Notice also that by a rectangle we mean only the boundary (not including the region inside).

   (a) Write a function that, upon input a structure of the above kind, determines whether the structure represents a valid rectangle.

   (b) Write a function that, upon input a valid rectangle, determines the area of the rectangle.

   (c) Write a function that, upon input a valid rectangle and two real numbers x, y, determines whether the point (x,y) lies inside, on or outside the rectangle.

5. Define a structure customer to specify data of customer in a bank. The data to be stored is: Account number (integer), Name (character string having at most 50 characters), and Balance in account (integer).

   Assume data for all the 10 customers of the bank are stored in the structure array :
   struct customer bank[10];

The function, transaction, is used to perform a customer request for withdrawal or deposit to his/her account. Every such request is represented by the following three quantities: Account number of the customer, request type (0 for deposit and 1 for withdrawal) and amount. The function prototype is as follows: int transaction (int account_number, int amount, struct customer bank [10])

The transaction function returns 0 if the transaction fails and 1 otherwise. The transaction fails only when the account balance is less than the withdrawal amount requested.

The array bank (defined above) is another input to the transaction function and is suitably updated after every request. In case of a failed transaction no change is made in the bank array.

Write a main() function which populates the array bank with values for 5 customers. Also, the main() should take a withdrawal request from the user (i.e., read values for account number, amount), and call the transaction function, and thereby print if it is a valid transaction. If valid, it should print the balance after the withdrawal.

PDS LAB – 8 (Section-13) Date: 16th October 2017
Pointers and Dynamic Memory Allocation
(Practice Problems)

1. Using pointers write C functions to swap (i) two values (say x and y) and (ii) two character strings (say s1=”xxx” and s2=”yyy”). Through main function call the above swap functions and print the values and strings before and after calling the functions.

2. Write a C function *int* factor_compute (int)* to return all factors of a given positive integer. Use pointers for passing the arguments and return the result. Demonstrate the utility of the above C function using main () function. Input the positive number through key board and print the number and its factors as output.

3. Using pointers write C functions to (i) *char* enter-text ( )* to enter input text terminated by enter key and return the entered text through a pointer and (ii) *int string_lenght (char *)* to determine the length of the string and return it to the main function. Through main () function, call the above 2 functions and print the input string and its length.

4. Using pointers write C functions to enter the array of integers using *int* enter_array ( ),* and compute the count of positive and negative numbers using *void compute_positive_negative_numbers (int *, int *, int *)*. Use malloc () and free () functions to allocate and release the memory.

5. Using pointers write a C program to input a student record which contains the following details: {int roll, char name[20], int age}. Define the structure to represent the details of the student. Write the C functions (with pointers) input the attributes of the student record *struct student* input()* through keyboard and return the student record to main function through pointer and print the same using *void print_student_record(struct student *).*

6. Modify the above program to enter and print multiple student records say “n” taken from the keyboard. Solve this problem using pointers. (Hint: in case of pointers, you have to allocate the memory).
   Example Output:
   Student [1] = 25 Krish 18
   Student [2] = 28 Ram 19
   Student [3] = 22 Satish 17

PDS LAB – 9 (Section-13) Date: 23rd October 2017
2D-Arrays and Linked Lists

1. Write C functions to enter and print the input matrix (A) using int ** enter_matrix (int m, int n) and void print_matrix(int **, int m, int n), respectively. Compute (i) transpose of a matrix (A^T) int ** transpose_matrix (int **, int m, int n), (ii) square of the input matrix (A^2) using A and A^T (*with modified multiplication rule*) using pointers. Demonstrate the above functions through main (), input the matrix (A), and print the matrix (A), its transpose (A^T) and square (A^2) in matrix form.

2. Write C functions (using 2D/3D pointers) to enter the names of states and cities char *** enter (int m, int n), where m and n are number of rows and columns, respectively, and print the same void_print (char***, int m, int n). Write C functions to (i) retrieve the state name for the given city name and (ii) retrieve the names of the cities for the given state name.

Andhra Pradesh	Vijayawada	Guntur	Vizag	Tirupati
Tamil Nadu	Chennai	Madurai	Coimbatore	Tiruchi
Karnataka	Bangalore	Mysore	Beedar	Mangalore
Uttar Pradesh	Kanpur	Luknow	Allahabad	Banaras

3. A linked list, where the node of the linked list (data component) holds the student record {int roll, char name[20], int age}. Write C functions to (i) create the list and fill the nodes with student records using void enter_list(struct student *, int n) and (ii) print the student records by traversing the list void traverse_print_list(struct stud *).

4. Continuation to the above problem: write C functions to (i) insertion of a node (at the beginning, at the end, in between based on some attribute of the student record), (ii) deletion of a node (at the beginning, at the end, in between based on some attribute of the student record) and (iii) print the student records in reverse order.

5. Assume that the linked list holds the data consisting of student records {int roll, char name[20], int age}. Write separate C function to sort the records based on each attribute of the student record. Print the student records in the original order (before sorting) and in sorted order as per each specific attribute.

6. Assume that the sequence of data elements are present in linked list. First create a linked list and place the data elements in nodes void enter_list(struct node *, int n). Write a C function to search the data elements present in the linked list and print the output as the given data element is present along with its position if the data element is present in the list, otherwise the output should indicate that the data element is not present in the list void_search_list(struct node *, int k).

PDS LAB – 9 (Section-13) Date: 23rd October 2017
2D-Arrays and Linked Lists
(Practice Problems)

1. Write a C function (using pointers) to count the number of positive, negative and zero values in an $M \times N$ matrix, and return the counts to main function. Use appropriate dynamic memory allocation functions to allocate the memory. Enter the values of M (rows) and N (columns) through key-board. Write C functions to enter the elements of a matrix using int ** enter_matrix(int m, int n) and print the elements of a matrix row-column form using print_matrix(int **, int m, int n) and mention the number of positive, negative and zero values present in the matrix.

2. Write C functions (using 2D/3D pointers) to enter the names of states and cities char *** enter (int m, int n), where m and n are number of rows and columns, respectively, and print the same void_print (char***, int m, int n).

Andhra Pradesh	Vijayawada	Guntur	Vizag	Tirupati
Tamil Nadu	Chennai	Madurai	Coimbatore	Tiruchi
Karnataka	Bangalore	Mysore	Beedar	Mangalore
Uttar Pradesh	Kanpur	Luknow	Allahabad	Banaras

3. Create 3 nodes, where each node holds one student record {char name[20], int roll_no, int age}. Let the student records be S1, S2 and S3. Each node carries the details of the student and pointer to the next student. Create a linked list with S1 as the 1st node and S3 as the last node. First enter the details of the student in each node and then provide the link from one node to other. Once the linked list is created, demonstrate the traversal by printing the records in sequence.

4. Assume that the sequence of data elements are present in linked list. First create a linked list and place the data elements in nodes void enter_list(struct node *, int n). Write a C functions to carry out the following operations: (i) length of the list (number of elements/nodes in the list), (ii) insert the value/node at the begin, end and at specific position (other than begin/end), (iii) delete the value/node at the begin, end and at specific position (other than begin/end).

PDS LAB – 10 (Section-13) Date: 30th October 2017
2D-Arrays using pointers and Linked Lists

1. Write C functions to implement the following operations.
    a) Create a linked list (*wordlist*) where each node stores a word.
    b) Compress the *wordlist* by removing repeated occurrences of words but maintaining the counts of each word and find the most frequently used word.
    c) Reorder a compressed wordlist by organizing the nodes in a sorted order of the words (based on lexicographical order) maintained in the nodes.
    d) Merge to linked lists as an ordered word lists.
    e) Using the above-mentioned program, do the following:

In the main program read a set of sentences (each sentence is terminated by '\n'). Create linked lists for each sentence. Then find the most frequently occurring words after merging the linked lists thus created.

2. A sparse matrix has very few non-zero elements. Instead of storing a N x M sparse matrix **A** as an array, it can be stored using N linked lists(one for each row), where the linked list corresponding to i-th row stores only non-zero column values in the row, that is, data of a node in this list stores (j, A[i,j]) provided $A[i,j] \neq 0$. Based on such representation, write C functions to perform the following operations.
    a) Read non-zero elements of a sparse matrix (say A[i,j]) and store them in respective linked list (i-th linked list).
    b) Given a sparse matrix, print its elements.
    c) With such storage implementation perform following matrix operations: (i) Addition of two sparse matrices and (ii) Multiplication of two sparse matrices.

3. A set can be represented by a linked list where a node of the linked list contains an element of the set. Thus a set S = {4, 55, 26, 103} can be stored in a linked list having 4 nodes where 4 is stored in node 1, 55 in node 2, 26 in node 3 and 103 in node 4. Note a set cannot have duplicate elements. With this representation of sets, write C functions to perform the following operations.
    a) Read a set of integers and store them in a linked list.
    b) Insert an element in an existing set.
    c) Verify whether two sets are disjoint. Perform union and intersection of two sets.

4. A linked list, where the node of the linked list (data component) holds the student record {int roll, char name[20], int age}. Write C functions to (i) insertion of a node (at the beginning, at the end, in between based on some attribute of the student record), (ii) deletion of a node (at the beginning, at the end, in between based on some attribute of the student record) and (iii) print the student records in reverse order.

5. Write a C program to display the elements of a matrix by traversing the path such that the sum of row and column indices increases incrementally.

Input:

```
1    2    3    4
5    6    7    8
9    10   11   12
13   14   15   16
```

Output-1:

```
1
2    5
9    6    3
4    7    10   13
14   11   8
12   15
16
```

Output-2:

```
1
5    2
9    6    3
13   10   7    4
14   11   8
15   12
16
```

PDS LAB – 11 (Section-13) Date: 6th November 2017
Searching and Sorting
(Practice Problems)

1. Write a C program to search an element in a sorted array using binary search method. Input the sorted array through keyboard and print the output as “Element is present” or “Element is not present” based on whether the element is present or not?

2. Write C program to sort the unique words present in a paragraph based on alphabetic order.

3. Assume that the sequence of data elements are present in linked list. First create a linked list and place the data elements in nodes. Write a C function to search the data elements present in the linked list and print the output as the given data element is present along with its position if the data element is present in the list, otherwise the output should indicate that the data element is not present in the list.

4. Write a C program to sort the given array of elements using Insertion sort technique.

5. Write a C program to sort the given array of elements using Merge sort technique.

6. Write a C program to sort the given array of elements using Quick sort technique.

PDS LAB – 8 (Section-13) Date: 16th October 2017
Pointers & Dynamic Memory Allocation

Note: *Please use DMA functions (malloc () and free ()) suitably for allocating and releasing the memory in case of using the pointers for representing the arrays and structures.*

1. Write a C function *int* factor_compute (int)* to return all factors of a given positive integer in increasing order. Use pointers for passing the arguments and return the result. Demonstrate the utility of the above C function using main () function. Input the positive number through key board and print the number and its factors as output.

2. Write a C function (using pointers) to compute and return the common factors for the given sequence of numbers. Use the solution of above question to solve this problem.

3. Write C functions (using pointers) to reverse the input string *char * reverse(char *),* and then check whether the given string is palindrome or not using *reverse*() function *int palindrome (char *)* ?

4. Write C functions (using pointers) to enter the text *char * enter_text(),* count the number of words present in the given text *int compute_number_words (char *)* and modify the text by rearranging the words based on word length *char * modify_text(char *).* Print the input text, number of words in the input text and text with reordered words.

5. Write C functions (using pointers) to enter the array int * enter_array(), and sort the array int * sort_array(int *). Print the input array and sorted array.

6. Write a C function (using pointers) to merge the 2 sorted arrays into a single sorted array and return the sorted array to the main function *float * merge_sorted_arrays (float *, float *, int m, int n).* Input the 2 unsorted arrays from keyboard using the C functions used in problem 5. Sort the given arrays using the C functions used in problem 5

7. Write C functions (using pointers) (i) to read (input) multiple student records from keyboard, (ii) functions for sorting the student records based on each attribute. Demonstrate the above functions by printing the original student records as well as attribute-based sorted records.
   Example Output:

Original student records (before sorting)	*Sorted records based on student roll*
Student [1] = 22 Satish 17	*Student [1] = 22 Satish 17*
Student [2] = 28 Ram 19	*Student [2] = 25 Krish 18*
Student [3] = 25 Krish 18	*Student [3] = 28 Ram 19*
Sorted records based on student name	*Sorted records based on student age*
Student [1] = 25 Krish 18	*Student [1] = 22 Satish 17*
Student [2] = 28 Ram 19	*Student [2] = 25 Krish 18*
Student [3] = 22 Satish 17	*Student [3] = 28 Ram 19*

PDS LAB – 11 (Section-13) Date: 6th November 2017
Searching and Sorting

1. Write a C program to search an element in a sorted array using ternary and interpolation search methods.

2. Write a program to read an integer array containing n (input by user in a random order) elements. Also, read a key k. Now, print the elements of the array in ascending order of their difference with the key k. Assume that every element of the array is unique.

   Example:
   Input Array: 12, 17, -7, 2, 8, 15
   Key: 6
   Sorted array based on absolute difference with key : 8, 2, 12, 15, 17, -7

3. Given two strings S 1 and S 2 , write a program to print the number of times the first string S 1 appears in the second string S 2 as a substring. For example if S 1 is “atg” and S 2 is “batgccatgatatga”, then the program should print 3. Program should work correctly for all possible inputs. DO NOT use any of the library functions defined under strings described above.

4. Write a C program to find the real root of a odd-degree polynomial using binary search method. Input the coefficients of a polynomial into an array through keyboard.

5. Write the following C functions:

   (a) Enter a paragraph
   (b) Compute the unique words in the paragraph
   (c) Sort the unique words of the paragraph
   (d) Search the given key word from the paragraph using above functions.

6. Let us consider a class of students, and their details (roll number, name and grade) were provided in 3 different arrays. Write a C function to (i) sort the students based on their names as per alphabetical order and print the output as mentioned below and (ii) sort the students based on their grade and print the output.

Input		
Roll No	Name	Grade
17EC001	Kiran Kumar	B
17EC002	Radha Krishna	B
17EC003	Anand Babu	A
17EC004	Hari Prasad	D
Output-1: Sorted list based on names		
17EC003	Anand Babu	A
17EC004	Hari Prasad	D

17EC001	Kiran Kumar	B
17EC002	Radha Krishna	B
Output-2: Sorted list based on grades		
17EC003	Anand Babu	A
17EC001	Kiran Kumar	B
17EC002	Radha Krishna	B
17EC004	Hari Prasad	D

7. Implement merge-sort using linked lists.

Ternery Search:
Dividing the array into 3 parts and search for the given element.

Interpolation Search:
The Interpolation Search is better over Binary Search, if the values in the sorted array are uniformly distributed. Binary Search always pick middle element to decide the following interval. On the other hand interpolation search may go to different locations according the value of key being searched. For example if the value of key is closer to the last element, interpolation search is likely to start search toward the end side.

pos = lo + [(x-arr[lo])*(hi-lo) / (arr[hi]-arr[Lo])]

Mid-Semester PDS Lab Test

Section-13; Date: 11th September 2017; Time: 2.00 to 4.30 pm; Marks: 100

(For ODD MACHINE NUMBERS)

1 Write a C program to add two binary numbers a and b (up to 8 bits) and the result of addition to be stored in variable c. The binary number should be entered through keyboard as a single number (not bit by bit). First, you need to validate the inputs (a and b), whether they are in binary form.
For binary arithmetic: 0+0 = 0; 0+1 = 1; 1+0 = 1; 1+1 = 0 (carry = 1); 1+1+1 = 1 (carry = 1)
Example:
B1 = 1011
B2 = 1111

```
      Carry-----> (1)    (1)    (1)

         (B1)      1      0      1      1

         (B2)      1      1      1      1
                  ------------------------------
B1+B2 =       1    1      0      1      0
                  ------------------------------
```

(40 Marks)

2 Write C functions to (i) read the values from keyboard and store them into an array (*void read_array(int a [], int a_size*), (ii) compute the unique elements from the array and store them into another array and return the size of the unique array
(*int compute_unique_array(int a[], int ua[], int a_size*) and (iii) merge the 2 unique arrays to single unique array and return the size of the merged array (*int merge_unique_array(int ua[], int ub [], int c[], int ua_size, int ub_size*). Print the following arrays *a, b, ua, ub,* and *c*.
Example:

a = {-25, -5, 10, -62, -5, 10, 2, 7, -5}
b = {-5, 2, 1, -25, 11, -5, 17, 2, -5}
ua = {-25, -5, 10, -62, 2, 7}
ub = {-5, 2, 1, -25, 11, 17}
c = {-25, -5, 10, -62, 2, 7, 1, 11, 17}

(60 Marks)

Mid-Semester PDS Lab Test

Section-5; Date: 27 Feb 2017; Time: 2.00 to 4.30 pm; Marks: 100

(For EVEN MACHINE NUMBERS)

1 Write a C program to generate an even parity check code word for the given message code word (up to 8 bits). The message code word should be represented in binary form. The even parity check code can be generated from the message code word based on number of ones present in the message code word. If the number of ones in the message code word is even, parity check bit zero is appended, otherwise one is appended.

After generating the parity check code word, the encoded message (the parity check code word) need to be transmitted to the receiver, through a channel. There may be chances of errors during transmission. Here the number of errors and their positions are entered through keyboard. Here introducing errors mean the original bit values will be flipped. Generate the received code word after introducing errors into original parity check code word (transmitted code word). At the end, you have to validate the received code word as whether it is erroneous or not, by again checking the number of ones in the received code word.
Example:

(a) Message code word = 10110011
Parity check bit = 1
Parity check code word = 101100111
Number of errors = 3
Error bit positions = 4,6 and 8.
Received code word = 11<u>1</u>0<u>0</u>1<u>1</u>111 (errors are underlined)
Validation of received code word = **erroneous**

(b) Message code word = 10100011
Parity check bit = 0
Parity check code word = 101000110
Number of errors = 2
Error bit positions = 4 and 7.
Received code word = 10<u>0</u>00<u>1</u>110 (errors are underlined)
Validation of received code word = **error free** **(40 Marks)**

2 Write C functions to (i) read the values from keyboard and store them into an array (*void read_array(int [], int a_size*), (ii) derive the new array from the input array such that it contains the elements of input array whose frequency of occurrence is one (i.e., only non-duplicate entries), and return the size of the new array (*int compute_new_unique_array(int a[], int ua[], int a_size*) and (iii) merge the 2 new unique arrays to single unique array such that it contains the common elements present in both the arrays and return the size of the merged array (*int merge_common_array(int ua[], int ub[], int c[], int ua_size, int ub_size*). Print the following arrays *a, b, ua, ub,* and *c*.
Example:

a = {-25, -5, 10, -62, -5, 10, 2, 7, -5}
b = {-5, 2, 1, -25, 11, -5, 17, -5}
ua = {-25, -62, 2, 7}
ub = { 2, 1, -25, 11, 17}
c = {-25, 2}

(60 Marks)

End-Semester PDS Lab Test

Section-13; Date: 13th November 2017; Time: 2.00 to 5.00 pm; Marks: 100

(For ODD MACHINE NUMBERS)

1. Write a C program to perform the following tasks using linked lists:
   a) Create a linked list and enter N positive integer values into the list such that each node carries one value.
   b) Traverse the list and print the values in the sequence as they are entered.
   c) By traversing the list, insert the new nodes such that successive nodes should not have either both even or both odd values. Suppose, if 2 successive nodes have even values 64 and 72, then we need to insert a new node in between them and place the value as one more than the value of previous node (i.e., 65). (*Don't use arrays for any manipulations. You are supposed to use only one LL*)
   d) Traverse the updated list and print the values present in the list.

   Example: 23, 44, 12, 53, 7, 15, 45, 87
   Output: 23, 44, 45, 12, 53, 54, 7, 8, 15, 16, 45, 46, 87 **(40 Marks)**

2 Consider a communication system where the sender sending the message encrypted by a security parameter *k*, called *key* to the receiver. The message is encrypted by the following way:
The message is arranged column-wise in an m x n matrix, and gives the encrypted output by reading it row-wise and the row-ordering is determined by the values of the array elements of key *k.* Every element of the array of key *k* specifies the row number. So key gives the order of reading the rows, and number of elements of the key is equal to number of rows.

Example:

Input message: PDS Lab end semester test is scheduled today
m = 5, n = 9

P	a	d	e	$	$	c	l	o
D	b	$	s	t	i	h	e	d
S	$	s	t	e	s	e	d	a
$	e	e	e	s	$	d	$	y
L	n	m	r	t	s	u	t	$

Here $ represents blank space. Key:

3	4	2	1	5

Encrypted Message: S$steseda$eees$d$yDb$st ihedPade$$cloL nmrtsut$

Write a C program to read a message, number of rows, number of columns and elements of key and print both the input and encrypted message by following the above method.

(60 Marks)

End-Semester PDS Lab Test

Section-13; Date: 13th November 2017; Time: 2.00 to 5.00 pm; Marks: 100

(For EVEN MACHINE NUMBERS)

1. Write a C program to perform the following tasks using linked lists:
   a) Create a linked list and enter N values into the list such that each node carries one value.
   b) Traverse the list and print the values in the sequence as they are entered.
   c) Access each node of linked list, if the value present in the node is negative, then remove the node from the list and place the value in the new list. (*You have to use only 2 LLs and you are not supposed to use any arrays.*)
   d) Traverse the new list as well as updated old list and print the values present in the lists.

   Example: 23, 44, -12, 53, -7, 15, -45, 87
   Output: New List = -12, -7, -45;
   Updated old list = 23, 44, 53, 15, 87 **(40 Marks)**

2. Consider a communication system where the sender sending the message encrypted by a security parameter *k*, called *key* to the receiver. The message is encrypted by the following way:
   The message is arranged row-wise in an m x n matrix and gives the encrypted output by reading it column-wise and the column-ordering is determined by the values of the array elements of key *k.* Every element of the array of key *k* specifies the column number. So key gives the order of reading the columns, and number of elements of the key is equal to number of columns.
   Example:
   Input message: Second pds lab test will be held in November.
   m = 5, n = 10

S	e	c	o	n	d	$	p	d	s
$	l	a	b	$	t	e	s	t	$
w	i	l	l	$	b	e	$	h	e
l	d	$	i	n	$	N	o	v	e
m	b	e	r	$	$	$	$	$	$

Here $ represents blank space. Key:

5	4	7	1	8	10	3	9	6	2

Encrypted Message: n$$n$oblor$eeN$$$wlmps$o$s$eescal$edthv$dtb$$elidb

Write a C program to read a message, number of rows, number of columns and elements of key and print both the input and encrypted message by following the above method.

(60 Marks)