Name: Ashwini Kumar Singh
Enroll. No. 19UCF075
Ashwini Kumar Singh

[Section-B]

1.)

```
// Program to reverse a no. with two digits

#include <stdio.h>

int main (void)
{
    int num, reven=0
    printf ("Enter a 2 digit number: ");
    while (num!0)
    {
        revn = (revn * 10) + num % 10;
        num = num / 10;
    }
    printf ("The no. with reversed digit is: %d", revn);
    return 0;
}
```

[signature]
19UCF075

(2)

```
// Program for factorial of any given no.
#include <stdio.h>

int main(void)
{
    int n, i;
    unsigned long long fact=1;
    printf("\nEnter an integer: ");
    scanf("%d", &n);

    if(n<0)
        printf("Factorial of -ve no. does not exist.\n");
    else{
        for(i=1; i<=n; ++i)
            fact *= i;
        printf("\n Factorial of %d = %llu\n", n, fact);
    }
    return 0;
}
```

Ashwani Kumar Singh

19UCE075

5.) Given $f(x) = x^3 - 3$

Given boundary value $[a, b] = [1, 2]$

$\therefore f(a) = f(1) = -2$

$\& \ f(b) = f(2) = 5$

$\frac{a+b}{2} = \frac{3}{2} \Rightarrow F(1.5) = 0.3750$

So $f(1.5) > 0.0005$, going to next iteration,

Hence, table upto 3 iteration can be.

Itrn	a	b	f(a)	f(b)	mid$(\frac{a+b}{2})$	$f(\frac{a+b}{2})$	Update
1	1	2	-2	5	1.5	0.3750	mid = b
2	1	1.5	-2	0.3750	1.25	-1.0468	mid = a
3	1.25	1.5	-1.0468	0.3750	1.3750	-0.40039	mid = a

Hence, required root, $x = \underline{\underline{1.25}}$

19UCE075

4.)

Given,

$$5x_1 + x_2 + 2x_3 = 8$$
$$3x_1 + 2x_2 - x_3 = 4$$
$$x_1 + 3x_2 + 5x_3 = 0$$

Writing as $AX = B$.

$$\begin{bmatrix} 5 & 1 & 2 \\ 3 & 2 & -1 \\ 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 10 \end{bmatrix}$$

Now $[A:B] = \begin{bmatrix} 5 & 1 & 2 & : & 8 \\ 3 & 2 & -1 & : & 4 \\ 1 & 3 & 5 & : & 10 \end{bmatrix}$

$R_1 \leftrightarrow R_3$

$$\begin{bmatrix} 1 & 3 & 5 & : & 10 \\ 3 & 2 & -1 & : & 4 \\ 5 & 1 & 2 & : & 8 \end{bmatrix}$$

$R_2 \to R_2 - 3R_1$, $R_3 \to R_3 - 5R_1$

$$\begin{bmatrix} 1 & 3 & 5 & : & 10 \\ 0 & -7 & -16 & : & -26 \\ 0 & -14 & -23 & : & -42 \end{bmatrix}$$

$R_3 \to R_3 - 2R_2$

$$= \begin{bmatrix} 1 & 3 & 5 & : & 10 \\ 0 & -7 & -16 & : & -26 \\ 0 & 0 & 9 & : & 10 \end{bmatrix}$$

19UCE075

Now: $9x_3 = 10$ — (i) $\quad x_3 = \frac{10}{9} = \frac{70}{63}$

$-7x_2 - 16x_3 = -26$ — (ii)

$x_1 + 3x_2 + 5x_3 = 10$ — (iii)

Solving through Back Substitution,

$$x_2 = \frac{74}{9 \times 7} = \frac{74}{63}$$

From (iii)

$$x_1 + 3 \times \frac{74}{63} + 5 \times \frac{10}{9} = 10$$

$$x_1 = 10 - \frac{50}{9} - \frac{225}{63}$$

$$\Rightarrow x_1 = \frac{58}{63}$$

Ans: $\boxed{\begin{aligned} x_1 &= 58/63 \\ x_2 &= 74/63 \\ x_3 &= 70/63 \end{aligned}}$

190CE075

Section C

(2)

```
// Program to give transpose of of mxn matrix.
# include <stdio.h>
int main (void){
    int r, c, i, j;
    printf("\n Enter no. of rows and columns: ");
    scanf("%d %d", &r, &c);
    int a[r][c], transpose[c][r];
    // Assigning elements to matrix
    printf("\n Enter matrix elements: \n");
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j){
         printf(" Enter element a%d %d: " i+1, j+1);
         scanf("%d ", & a[i][j]);
        }

    // Displaying the matrix. a
    printf("\n Entered matrix: \n");
    for(i=0; i<r; i++)
        for(j=0; j<c; j++)
         { printf("%d" a[i][j]);
          if(j==c-1)
                printf("\n");
         }
    // finding the transpose
    for(i=0; i<r; ++i).
        for(j=0; j<c; ++j)
            transpose[j][i] = a[i][j];
```

19UCE075

```
//Displaying the transpose matrix
printf("\n Transpose of the matrix: \n");
for(i=0; i<c; ++i)
    for(j=0; j<r; ++j)
    {
        printf("%d", transpose[i][j]);
        if(j==r-1)
            printf("\n");
    }
return 0;
}
```

eg. entered matrix:

1 4 0
-5 2 7

Transposed matrix

1 -5
4 2
0 7

19UCE075

3.)

$$f(x) = 2x^3 - 2.5x - 5$$

$$\therefore f'(x) = 6x^2 - 2.5$$

$$\boxed{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}}$$

$$\Rightarrow x_{n+1} = x_n - \frac{2x_n^3 - 2.5x_n - 5}{6x_n^2 - 2.5}$$

Given, $x_0 = 2$

Itrn	x_n	$f(x_n)$	$f'(x_n)$	x_{n+1}
1	2	6	21.5	1.7209
2	1.7209	0.8911	15.2696	1.6626
3	1.6626	0.348	14.0849	1.6601

So, after 3 iteration the required value of x is $\boxed{x = 1.6601}$

19UCE075

Section 1:

Ans-(6) The secant method does not require derivation of the function for solving non linear function/ equations, while NR method need it and sometimes it is not possible to get derivative.

Ans (5) The order of the rate of convergence in the NR method of solving non-linear eqn is 2.

Ans-(3) The no. of significant digits in 0.0025 is 2.

Ans- (2) A decimal no. having large no. of terms or infinite no. of terms is rounded to some terms after the decimal.

e.g: 4.3256 is rounded as 4.33 causing an error of 4.33 = 0.0056.

Ans-(1) The case in which the terms at the end of number are truncated after some terms without rounding off, the error occured is called truncation error.

e.g: 4.3256 after truncation of 2 decimal we get 4.32.
The error is 4.3256-4.32 = 0.0056
This is truncation error

19UCE075

Ans 4: The condition to ensure convergence is diagonahally dominant matrix.

i.e $|a_{ii}| \geq \sum |a_{ij}|$

or $|a_{11}| \geq |a_{12}| + |a_{13}|$