ESC101: Fundamentals of Computing
Lab 4 (January 22-28, 2014)

Duration: 3 hours **Total Marks: 50**

1. **Exponent and Logarithm**

   Define two functions, `myex()` and `mylogx()`, to compute an approximation of **exponent** and **logarithm** of a float value. The `main` function should first ask the user to present a choice:

   - On an input "1", the program should read a floating point number x from the user and should call the function `myex()` to print the value of e^x using the following expansion:

     $$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad \text{for all } x.$$

     Consider only the first 3 terms of the expansion.

   - On the input "2", it should read a floating point number x from the user and should call the function `mylogx()` to print the value of $\log(x)$ using the following expansion:

     $$\log(x) = 2\left[\left(\frac{x-1}{x+1}\right) + \frac{1}{3}\left(\frac{x-1}{x+1}\right)^3 + \frac{1}{5}\left(\frac{x-1}{x+1}\right)^5 \ldots\right] \quad \text{for } x > 0.$$

     Consider only the first 3 terms of the expansion.

   Here are some sample interactions of the program:

   ```
   $ ./a.out
   Enter 1 or 2: 1
   Enter the number: 3.2
   e^x of 3.200000 is 9.320000

   $ ./a.out
   Enter 1 or 2: 1
   Enter the number: 5.7
   e^x of 5.700000 is 22.945000

   $ ./a.out
   Enter 1 or 2: 2
   Enter the number: 3.2
   log(3.200000) is 1.159207

   $ ./a.out
   Enter 1 or 2: 2
   Enter the number: 1024.08
   log(1024.079956) is 3.054983
   ```

   [Total 10 marks]

2. **Income Tax**

Write a function `mytax()` which takes into input the taxable income (say `ti`) of a person in the assessment year and compute the net tax for the person using the following rules:

(a) If the amount is upto $200,000$, tax $=0$.

(b) If the amount is between $200,001$ and $500,000$, tax $=$ 10% of (`ti` - 200000)

(c) If the amount is between $500,001$ and $1,000,000$, tax $=$ 30000 $+$ 20% of (`ti` - 500000)

(d) If the amount is above $1,000,000$, tax $=$ 130000 $+$ 30% of (`ti` - 1000000)

The `main` function should take the taxable amount as input from the user, and using the function `mytax` it should output the income tax payable upto 2 decimal places.

You may assume that the taxable income is an integer.

Here are some sample interactions of the program:

```
$ ./a.out
Enter taxable income: 710536
Income tax is 72107.20

$ ./a.out
Enter taxable income: 1515150
Income tax is 284545.00

$ ./a.out
Enter taxable income: 222222
Income tax is 2222.20
```

[Total 10 marks]

3. **Compound interest**

Write a function `compound_interest` which takes as arguments, the principal amount (`p`), nominal annual rate of interest (`r`, expressed in decimal i.e. 6% = 0.06), total time in years (`t`), and number of compounding periods per year (`n`) and returns the amount after the interest using the following formula:

$$A = p\left(1+\frac{r}{n}\right)^{\lfloor nt \rfloor}$$

The `main` function should take values of principal, rate, time and number of compounding periods as input from the user, and using the function `compound_interest` it should output the final amount upto 2 decimal places.

Note:

(a) `p, r, t` are float and `n` is an integer which is either 1, 2, 3 or 4.

(b) Also, the product `nt` will never be more than 4.

Here are some sample interactions of the program:

```
$ ./a.out
Enter principal: 1000
Enter rate of interest: .06
Enter time in years: .5
Enter compounding periods per year: 4
Amount after interest is 1030.22
```

```
$ ./a.out
Enter principal: 2000
Enter rate of interest: .9
Enter time in years: 2
Enter compounding periods per year: 1
Amount after interest is 7220.00
```

[Total 15 marks]

4. Manhattan Distance

The Manhattan Distance between two points (x_1, y_1) and (x_2, y_2) is defined as,

$$|x_1 - x_2| + |y_1 - y_2|.$$

Write a function `man_dist` that takes the coordinates of two points as arguments (x_1, y_1, x_2, y_2) and returns the Manhattan Distance between the two points. You can assume that all values are integral.

The `main` function should take coordinates of three points as input from the user (say point A, point B and point C), and using the function `man_dist` it should output which pair of points has the greatest Manhattan distance (there can be more than one pairs).

DO NOT use any function from the `math.h` library for this problem.

Here are some sample interactions of the program:

```
$./a.out
Enter x coordinate of point A: 0
Enter y coordinate of point A: 0
Enter x coordinate of point B: -1
Enter y coordinate of point B: 4
Enter x coordinate of point C: 6
Enter y coordinate of point C: -8
Point B and C.
```

```
$./a.out
Enter x coordinate of point A: -1
Enter y coordinate of point A: 1
Enter x coordinate of point B: 2
Enter y coordinate of point B: -2
Enter x coordinate of point C: 0
Enter y coordinate of point C: 2
Point A and B.
Point B and C.
```

[Total 15 marks]