Course:	Computer Application in Physics (MPC 33)
Semester:	M.Sc. III, 2018
Question Set No.:	I
Compiled by:	Pijush K. Ghosh
Date :	July 23, 2018

Sorting, Searching etc.

1. Write a C++ programme to arrange a given set of N numbers in ascending/descending order.

2. Write a C++ program to obtain the maxima, minima and the number of data from a set of numbers written in a file.

3. A class of N students appeared for an examination. Write a C++ program to implement the following with $N \geq 10$:

   - Read the number of students from a file.
   - Read from the file the marks obtained by each student.
   - Find the average marks of the class.
   - Count the number of students who scored more than the average.

4. Write a programme in C++ to find the square-root of an integer up to three decimal places. You are not allowed to use the in-built C++ functions *sqrt(x), pow(x,p)*. You may use any search-algorithm to find the root.

Preparing a Data file and Plotting

1. Consider the functions: $\psi_1(x) = 5sin(x)$, $\psi_2(x) = 5sin(2x)$, $\psi_3(x) = 5sin(3x)$, $\psi(x) = \psi_1(x)+\psi_2(x)+\psi_3(x)$. Write a C++ program to tabulate the values of $\psi_1(x)$, $\psi_2(x)$, $\psi_3(x)$ and $\psi(x)$ at intervals of 0.01 for $0 \leq x \leq 6.3$ and plot using Gnuplot.

2. **Wave-functions of a quantum particle**: Write a program in C++ to tabulate the values at equal intervals of the first three normalized eigenstates $\psi_n(x), n = 1, 2, 3$ of a particle moving in (a) an infinite square well potential and (b) a simple harmonic oscillator potential. Parameters like mass(m) of the particle, angular frequency(ω) of the oscillator, length(L) of the well etc. may be taken as inputs.
   Plot $\psi_n(x)$ vs. x and study how $\psi_n(x)$ changes with varying (i) m, (ii) L and (iii) ω.

3. **Lissajous curves**: Write a program in C++ to tabulate the values of the following trigonometric functions,

$$\begin{aligned} X(t) &= A_X cos(2 f_X t), \\ Y(t) &= A_Y sin(2 f_Y t + \phi), \\ Z(t) &= X(t) + Y(t) \end{aligned}$$

at the equally spaced times $t = n\Delta t$, with n = 0, ... , N. Take the parameters $f_X, f_Y, A_X, A_Y, \phi, \Delta t$ and N as inputs and write the outputs $X(t), Y(t)$ and $Z(t)$ in a file.

- Plot $X(t)$ vs. $Y(t)$. Give graphical evidences in support of the fact that closed curves are obtained for rational $\frac{f_X}{f_Y}$.
- Plot $Z(t)$ $vs.$ t and demonstrate the phenomenon of beats by taking $f_X \approx f_Y$.

4. **Butterfly curve**: Write a program in C++ to tabulate the values of $X(t)$ and $Y(t)$:

$$X(t) = r(t)cost(t),\ Y(t) = r(t)sin(t), r(t) = e^{cos(t)} - 2cos(4t) + sin^5(\frac{t}{12})$$

at the equally spaced time-intervals Δt for $-T \leq t \leq T$, where Δt and T are to be treated as inputs. Plot $X(t)$ vs. $Y(t)$.

5. **Surface plots: Sphere, Torus, Spherical Harmonics etc.** Use the parametric plot('set parametric') and surface plot('splot') features of gnuplot to implement the following:

(a) Plot for (i) $c = 0, a > 0$, (ii) $c > a, a, c > 0$, (iii) $c = a > 0$, (iv) $c < a, c, a > 0$.

$$x = (c+acosv)cosu, y = (c+acosv)sinu, z = asinv,\ u, v \in [0, 2\pi), a, c \in \Re$$

(b) Plot spherical harmonics: (i) $Y_0^0, Y_1^0, Y_1^1, Y_1^{-1}$.

Sum, Infinite Series etc.

1. Write a C programme to find the sum $S_{N,k} = \sum_{i=1}^{N} i^k$ for pre-fixed integer values of N and k.
   - Check your numerical result by comparing it with the known exact results for $k = 1, 2$.
   - Allow N to take very large values and show that the series $S_{\infty,k}$ converges to a finite value for $k \leq -2$ and diverges for $k \geq -1$. Tabulate the results for at least six different values of k distributed equally in the two ranges specified above.

2. Evaluate the sum

$$S = \sum_{r=0}^{N} (-1)^r\ {}^NC_r, \quad {}^NC_r \equiv \frac{N!}{r!(N-r)!}$$

for five different values of $N \geq 10$. Does your result depend on N? Note that ${}^NC_r$ is to be evaluated by using a C programme.

3. Write a C programme to find the sum

$$(i) S_1 = \sum_{i \neq j \neq k \neq i; i,j,k=1}^{N} \frac{1}{(i-j)(j-k)}, \quad (ii) S_2 = \sum_{i \neq j \neq k \neq i; i,j,k=1}^{N} \frac{i+j+k}{(i-j)(j-k)}$$

for a fixed value of N

4. Write a C programme to find the value of $f(x)$ at a given value of x by using the power-series expansion of the function $f(x)$.

$$(i)\ f(x) = e^{\pm x},\ (ii)\ cosx,\ (iii)\ sinx,\ (iv)\ tanhx$$

- Tabulate your results for at least ten different values of x between $-100 \leq x \leq 100$.
- Check the accuracy of your result up to four decimal places.
- Plot the function $f(x)$ for the range $-100 \leq x \leq 100$ by using Gnuplot. Mark the points $f(x_i), i = 1, 2, \ldots, 10$ on the same plot, where $f(x_i)$ correspond to the numerically obtained values of the function.

5. Write a C programme to generate Fibonacci sequences:

$$F_n = F_{n-1} + F_{n-2},\ n \geq 2, \quad F_0 = 0, F_1 = 1$$

- Tabulate the results for first 100 elements in this sequence.
- Check numerically while implementing the previous task that the identity holds $\sum_{i=0}^{N} F_i^2 = F_N F_{N+1}$ for all N.

6. The Kummer's function is defined as

$$\begin{aligned} M(a, b, x) &= 1 + \frac{ax}{b} + \frac{(a)_2 x^2}{(b)_2 2!} + \ldots + \frac{(a)_n x^n}{(b)_n n!} + \ldots, \\ (a)_n &\equiv a(a+1)(a+2)\ldots(a+n-1),\ (a)_0 \equiv 1 \end{aligned}$$

Evaluate $M(a, b, x)$ for pre-fixed values of $a > 0, b > a$ and x with an accuracy of the order of 10^{-4}.
Check the relation $M(a, b, x) = e^x M(b - a, b, -x)$
Plot $M(a, b, x)$ for fixed a, b and $-1 \leq x \leq 1$.

7. The νth order Bessel function $J_\nu(x)$ has the following series expansion:

$$J_\nu(x) = \left(\frac{x}{2}\right)^\nu \sum_{n=0}^{\infty} \frac{(-\frac{1}{4}x^2)^n}{n!\Gamma(\nu + n + 1)}, \quad \Gamma(1) = 1,\ \Gamma(n+1) = n\Gamma(n)$$

Evaluate $J_1(x)$ for 20 distinct points within the range $-1 \leq x \leq 1$ with an accuracy of the order of 10^{-4}. Plot $J_1(x)$ for $-1 \leq x \leq 1$.

8. Use the recursion relation satisfied by the Legendre functions $P_n(x)$,

$$(n+1)P_{n+1}(x) = (2n+1)\ x\ P_n(x) - nP_{n-1}(x), \quad P_0(x) = 1, \quad P_1(x) = x$$

to find $P_5(x)$ on 20 distinct points within the interval $-1 \leq x \leq 1$.
Plot $P_5(x)$ vs. x using Gnuplot.

Logic Gates & Truth Table

1. Write a C programme for n-input truth table for (i) OR and (ii) AND gate.

Complex Numbers

1. Write a C++ program to evaluate the sum, difference and product of two given complex numbers.

2. Write a C++ program to compute the square and cube roots of a complex number given in the form $x + iy$ numerically.

Number Theory

1. **Primality Test**: Write a C programme to determine whether a given number is a prime or not. Find all the primes between 1 and 1000 by using your programme.

2. Write a C programme to find the greatest common divisor of three given integers less than 1000. [**Hint**: You may use the following result from number theory for writing the programme. Any number can be factorized as a product of powers of prime numbers. This factorization is unique for a given number.]

3. **Change of bases**: Write a C programme to find the binary equivalent of a given decimal number.

4. Generalize the above problem to the case when a decimal number is expressed in a number system with base $b, 2 \leq b \leq 8$.

5. **Change of bases**: Write a C programme to find the decimal equivalent of a given binary number.

6. **Diophantine equations**: Write a C programme to find the integer-valued solutions for X, Y, Z of the equation, $X^2 + Y^2 = Z^2$.

Recreational Mathematics, Games, Calenders etc.

1. **Magic square**: Read in a square array of positive integers $n(1 \leq n \leq N^2)$, and determine if it is a magic square or not. If yes, find its magic constant.

2. **Constructing a Magic square**: The ij^{th} elements of a magic square of odd order may be constructed by using the following formula:

$$M_{ij} = n\left\{\left(i + j - 1 + \left[\frac{n}{2}\right]\right) mod\ n\right\} + \{(i + 2j - 2)\, mod\ n\} + 1$$

   where $[x]$ denotes the integral part of x and $p\, mod\, q \equiv p - q[p/q]$. Construct a magic square of order 9 and print it in the form of a square array.

3. **Playing Cards**: Consider a standard deck of 52 playing cards. Assign the numbers $0 - 51$ in order to each card. Read in a number x $(0 \leq x \leq 51)$ and identify the corresponding card.

4. **Chess**: Assign the numbers $0 - 63$, row by row, to the various squares on a 8×8 chess-board. Read in two numbers $x, y(0 \leq x, y \leq 63)$ and determine if the queen at x can capture the queen at y.

5. **Calender**: Read in a date in the form date(D)/month(M)/Year(Y) in the Gregorian calender. Print the day(d) of the week corresponding to the date by using the formula:

$$d = D + [2.6M - 0.2] - 2a_0 + a_1 + [\frac{a_0}{4}] + [\frac{a_1}{4}](mod 7)$$

   where $a_0 = [Y/100], a_1 = Y(mod\ 100)$. Assign $March = 1, April = 2, \ldots$ and $0 = Sunday, 1 = Monday, \ldots$.

Vector and Tensor Operations

1. Write a C programme to implement the $d \geq 2$ dimensional Kronecker-delta tensor.

2. Write a C programme to implement the $d \geq 2$ dimensional Levi-Civita tensor.

3. Write a C Programme to implement dot product, cross product, scalar-triple product and vector triple-product of any given three dimensional vectors.

4. Write a C Programme to find the reciprocal lattice vectors for a given set of vectors.

Matrix Operations

1. Write a C++ program to evaluate the sum, difference and product of two given matrices of order $N \times N$.

2. Write a C++ program to evaluate the determinant and trace of a given matrix of order $N \times N$.

3. Write a C++ program to determine the inverse of a given 2×2 matrix.

4. The Pauli matrices $\sigma_{1,2,3}$ are given by,

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

(a) Write a C programme to verify the following relations:

$$\sigma_i \sigma_j = i\epsilon_{ijk}\sigma_k, \quad \{\sigma_i, \sigma_j\} = 2\delta_{ij}, \quad [\sigma_i, \sigma_j] = 2i\epsilon_{ijk}\sigma_k, \quad i, j, k = 1, 2, 3$$

(b) Define four matrices $\xi_{1,2,3,4}$ in terms of the Pauli matrices $\sigma_{1,2,3}$ and the 2×2 identity matrix I:

$$\xi_1 = i\sigma_1 \otimes \sigma_2, \quad \xi_2 = i\sigma_2 \otimes \sigma_2, \quad \xi_3 = -\sigma_3 \otimes \sigma_2, \quad \xi_4 = I \otimes \sigma_3$$

Implement the operation $A \otimes B$ using a C programme and find the form of the matrices (i)$\xi_1, \xi_2, \xi_3, \xi_4$, (ii) $\{\xi_i, \xi_j\}$, (iii) $\gamma_5 = \xi_1\xi_2\xi_3\xi_4$.

Note: The outer-product of two operators A and B is denoted as $A \otimes B$. An example of outer-product of two 2×2 matrices A and B are given below:

$$A \otimes B = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \otimes \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix}$$

5. A quantum system of two spin-$\frac{1}{2}$ particles is described by the Hamiltonian H:

$$H_{\pm} = \alpha \vec{S}_1 \cdot \vec{S}_2 + \beta \left(S_1^z \pm S_2^z\right), \quad (\alpha, \beta) \in R.$$

(a) Represent the operators $\vec{S}_{1,2}$ in terms of the Pauli matrices as $S_1^a = \frac{1}{2}\sigma^a \otimes I, S_2^a = \frac{1}{2} I \otimes \sigma^a$ and write H_+ as a 4×4 matrix.

(b) Check numerically that the following states are eigenstates of H_+:

$$|1,1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1,0\rangle = \frac{1}{\sqrt{2}}\left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right],$$
$$|1,-1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}, |0,0\rangle = \frac{1}{\sqrt{2}}\left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right],$$

and find the corresponding eigenvalues.

(c) Are all the states above also eigenstate of H_-?

Coin Tossing, Random Walks etc.

(a) An unbiased coin is thrown N times. Simulate the process numerically by writing a C++ program. Read the value of N from the terminal and write the output to a file.

- Record the sequence of 'heads'(H) and 'tails'(T).
- Identify the longest sub-sequence of (i) consecutive 'heads' and consecutive 'tails'.

- Count the total number of 'heads' and 'tails' at the end of the process.
- Repeat the computations for the case of two unbiased coins thrown simultaneously N times. Record the sequence of HH, TT and HT/TH.

(b) Write a C++ program to evaluate the value of π numerically with an accuracy up to two decimal places by computing the ratio of the areas of a circumscribed square and its in-circle.

(c) A drunkard is moving randomly on one dimensional uniform lattice of unit size. Choose the probability of moving in the forward direction to be p. Read in p and the initial position of the drunkard and simulate the process numerically. Record the position(P) of the drunkard from his/her initial position after each step for the total steps $N = 1000, 2000, \ldots, 10000$. Plot P *vs. step* for each N.

(d) A drunkard is moving randomly on a uniform square-lattice of unit lattice constant. At each vertex, there are four possible directions: forward, backward, right and left. Choose the probability to move in any one of these directions to be 1/4. Simulate the process numerically.

- Record the Position(x, y) of the drunkard after each step for the total steps $N = 1000, 2000, \ldots, 10000$. Generate a surface-plot of the variables $(x, y, step)$ for each N.
- Print the path(plot of $yvs.x$) followed by the drunkard.
- Repeat your computations for a drunkard moving randomly on a uniform cubic lattice.