ALLAMA IQBAL OPEN UNIVERSITY

USMAN NOTES
www.usmannotes.com

Name:

Registration No.:

Course Code: 1349

Program: HSSC

Semester: Autumn 2026

Allama Iqbal Open University, Islamabad

Department of Mathematics

Course: Introduction to Business Mathematics (1349)

Level: HSSC

Semester: Autumn 2026

Assignment No. 02

Units: 6–9

Total Marks: 100

Pass Marks: 40

SOLVED ASSIGNMENT GUIDE — For Study & Reference

Note for students: This guide is prepared for study and reference only. Read it carefully, understand every step, and then write your assignment in your own words and your own hand. Submitting copied material is penalised under the AIOU plagiarism policy, and tutors reward your own analysis and presentation.

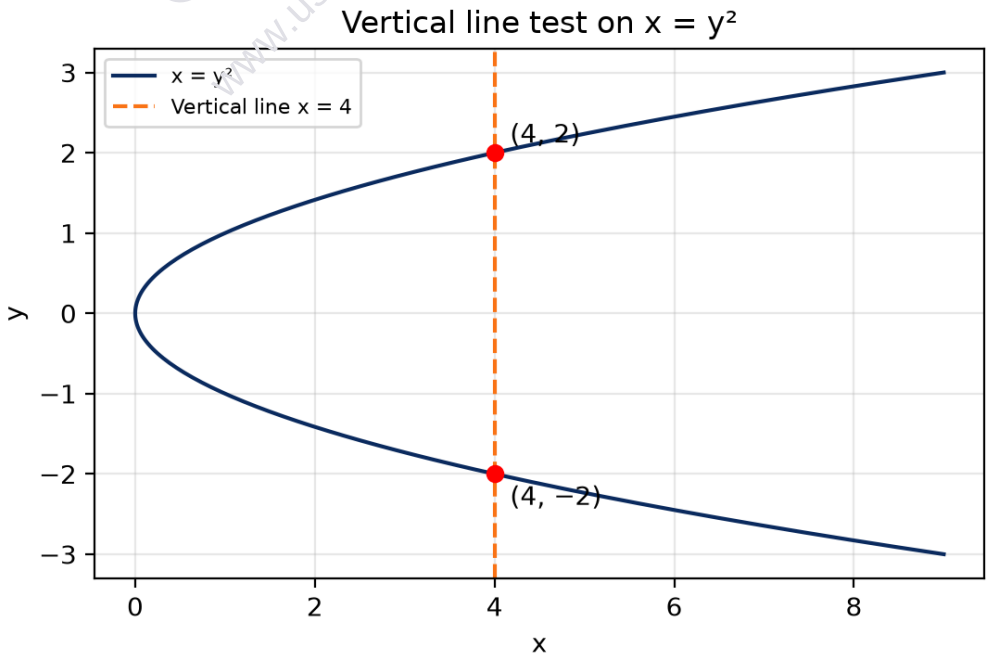
Question No. 1

Question 1 (20 marks): (a) Define a function and provide an example; explain the vertical line test and why $y^2 = x$ is not a function. (b) What is the difference between even and odd functions? (c) Explain the concept of a singular matrix. (d) Find the transpose of the matrix A = [[1, 2], [3, 4]]. (e) Calculate the cofactor of the element in the first row and first column of the matrix B = (5, 7; 8, 6).

Introduction

This question opens the second half of the course by joining its two main themes: functions, which describe how one quantity depends on another, and matrices, which organise numbers into arrays that can be added, multiplied, transposed and inverted. Part (a) defines a function, illustrates it, and introduces the vertical line test, the graphical test that decides whether a relation is a function, applying it to the relation $y^2 = x$. Part (b) distinguishes even and odd functions by their symmetry. Parts (c) to (e) turn to matrices: the idea of a singular matrix, the transpose of a given 2×2 matrix, and a cofactor calculation. These definitions are not decoration. Functions are the language in which cost, revenue and profit relationships of business are written, and singular matrices mark exactly the point at which a system of equations stops having a unique solution, so each idea is explained with its reason as well as its rule, and each small computation is set out in full and checked.

USMAN NOTES www.usmannotes.com



(a) Functions and the Vertical Line Test

A function is a rule that assigns to each input value exactly one output value. If the input is denoted by x and the output by y, we write y = f(x), read “f of x”; x is the independent variable, y is the dependent variable, the set of permitted inputs is the domain, and the set of outputs actually obtained is the range. The decisive word in the definition is “exactly one”: a function may certainly give the same output for different inputs, but it may never give two different outputs for the same input. An example makes the idea concrete: the function f(x) = 2x + 3 assigns to the input 0 the output 3, to the input 1 the output 5, to the input 4 the output 11, and to the input −1 the output 1 — one clear output for every input. A business example is a taxi fare rule or a commission rule, or, more formally in this course, a cost function C(x) giving the total cost of producing x units: for each level of output there is one definite total cost. By contrast, a mere relation is any pairing of inputs with outputs; every function is a relation, but not every relation is a function, and telling the two apart is the business of the vertical line test.

The vertical line test states: a curve in the xy-plane is the graph of a function of x if and only if every vertical line (a line x = constant) meets the curve at most once. The reasoning follows straight from the definition: a vertical line x = k gathers together all the points that have the same input k, so if it cuts the curve twice, that single input has two different outputs and the relation is not a function; if no vertical line cuts it twice, every input has at most one output and the relation is a function. Apply this to the relation $y^2 = x$. Its graph is a parabola lying on its side, opening to the right, as shown in the figure. Solving for y gives $y = \pm\sqrt{x}$, which already announces the problem: at x = 4, for example, y can be +2 or −2, two outputs for one input. Geometrically, the vertical line x = 4 cuts the parabola at the two points (4, 2) and (4, −2), so the relation fails the vertical line test and $y^2 = x$ is not a function of x. It is still a perfectly good relation, and it can even be viewed as describing x as a function of y; the failure is only about which variable is the

input. Notice the contrast with the ordinary parabola $y = x^2$, which passes the test and is a function: a vertical line meets it exactly once.

Kind	Test	Symmetry of graph	Examples
Even function	$f(-x) = f(x)$	Symmetrical about the y-axis	x^2, x^4, cos x
Odd function	$f(-x) = -f(x)$	Symmetrical about the origin	x, x^3, sin x
Neither	Neither identity holds	No such symmetry in general	$x^2 + x$, 2^x

(b) Even and Odd Functions

Even and odd functions are classified by what happens when the input changes sign. A function f is even if $f(-x) = f(x)$ for every x in its domain: replacing x by $-x$ leaves the value unchanged. Its graph is symmetrical about the y-axis, so the left half is the mirror image of the right half. The standard example is $f(x) = x^2$, since $(-x)^2 = x^2$; other examples are $f(x) = x^4$ and $f(x) = \cos x$. A function f is odd if $f(-x) = -f(x)$ for every x in its domain: replacing x by $-x$ reverses the sign of the value. Its graph is symmetrical about the origin, remaining unchanged under a half-turn (rotation through 180°) about the origin. The standard example is $f(x) = x^3$, since $(-x)^3 = -x^3$; other examples are $f(x) = x$ and $f(x) = \sin x$. The figure shows the two symmetries side by side. Many functions are neither even nor odd — $f(x) = x^2 + x$, for instance, satisfies neither identity, as can be seen at $x = 1$ and $x = -1$ — and the classification is decided by the algebraic test, not by whether the function "looks" symmetrical. The idea is useful well beyond this course:

USMAN NOTES www.usmannotes.com

symmetry halves the work of sketching a graph, and in later mathematics the even and odd parts of a function behave differently under integration over a symmetrical interval.

(c) The Singular Matrix

A square matrix is called singular if its determinant is equal to zero. The concept matters because the determinant controls the existence of the inverse: a square matrix A has an inverse A^{-1} exactly when its determinant is not zero, so a singular matrix is precisely a square matrix that has no inverse. The word "singular" itself signals the exception — such a matrix is the unusual, degenerate case. For a 2 × 2 matrix [[a, b], [c, d]] the determinant is ad − bc, so the matrix is singular exactly when ad = bc; geometrically this means one row is a multiple of the other (the rows are proportional), as in [[2, 4], [1, 2]], whose determinant is 2 × 2 − 4 × 1 = 0. The algebraic meaning appears when matrices are used to solve equations, as in Question 3: if the coefficient matrix of a system is singular, the system does not have one unique solution; it has either no solution at all (the corresponding lines are parallel and distinct) or infinitely many solutions (the lines coincide). Singularity is therefore the matrix signal of a system that cannot be untangled into a single answer, and checking the determinant before attempting to invert a matrix, as is done throughout this assignment, is the professional habit this concept teaches. A non-singular matrix, by contrast, has determinant different from zero and possesses a unique inverse.

(d) Transpose of A and (e) A Cofactor of B

The transpose of a matrix, written A^T, is the matrix obtained by interchanging its rows and columns: the first row becomes the first column, the second row becomes the second column, and in general the element in row i and column j moves to row j and column i. For A = [[1, 2], [3, 4]], the first row (1, 2) becomes the first column and the second row (3, 4) becomes the

second column, so A^T = [[1, 3], [2, 4]]. The check is that transposing twice returns the original matrix, $(A^T)^T$ = A, which is quickly verified here. The diagonal elements 1 and 4 stay in place, while the off-diagonal elements 2 and 3 exchange positions — a pattern that always holds and makes 2 × 2 transposes easy to write and to check at a glance.

For part (e), recall the definitions. The minor of an element of a square matrix is the determinant of the smaller matrix left after deleting the element's row and column. The cofactor is the minor with a sign attached according to the checkerboard pattern (+ −; − +), the sign for row i, column j being (−1)^(i+j). In B = (5, 7; 8, 6), the element in the first row and first column is 5. Deleting the first row and the first column leaves the single element 6, whose determinant is simply 6, so the minor is 6. The sign for position (1, 1) is (−1)^(1+1) = $(-1)^2$ = +1, positive, as the top-left corner of the checkerboard always is. Hence the required cofactor is (+1) × 6 = 6. For completeness, the other cofactors of B are: of 7 (row 1, column 2): −8; of 8 (row 2, column 1): −7; and of 6 (row 2, column 2): +5, giving the cofactor matrix [[6, −8], [−7, 5]], whose transpose is the adjugate used in finding inverses in Question 2. The asked-for answer, however, is the single value: the cofactor of the first-row, first-column element of B is 6.

Conclusion

A function assigns exactly one output to each input, and the vertical line test translates that definition into a picture: $y^2 = x$ fails the test, since the line x = 4 meets its graph at both (4, 2) and (4, −2), so it is not a function. Even functions satisfy f(−x) = f(x) and are symmetrical about the y-axis; odd functions satisfy f(−x) = −f(x) and are symmetrical about the origin. A singular matrix is a square matrix whose determinant is zero and which therefore has no inverse. The transpose of A = [[1, 2], [3, 4]] is A^T = [[1, 3], [2, 4]], and the cofactor of the first-row, first-column element of B is +6. These are the working tools for the matrix calculations that follow.

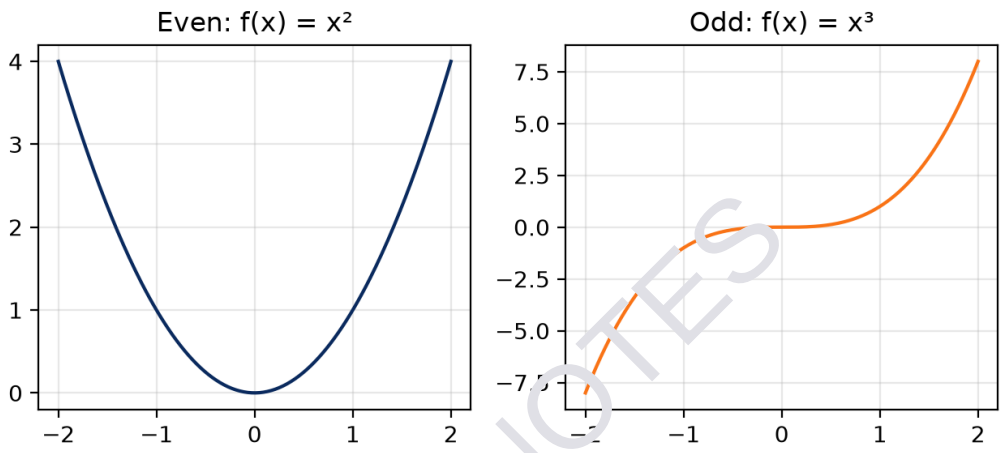


Figure: an even function, f(x) = x², symmetrical about the y-axis, and an odd function, f(x) = x³, symmetrical about the origin.

Question No. 2

Question 2 (20 marks): (a) Matrix types: Differentiate between a scalar matrix and an identity matrix. Provide examples. (b) What is the determinant of a 2×2 matrix? Find the determinant of [[2, 3], [1, −5]]. Is this matrix singular? (c) Inverse matrix: Find the inverse of [[5, 9], [4, 1]].

Introduction

Matrices earn their place in business mathematics because they package a whole table of numbers into a single object on which arithmetic can be performed: sales of several products over several months, costs of several inputs, or the coefficients of several equations can each be held in one matrix and manipulated together. This question examines three of the basic ideas that make that machinery work. Part (a) distinguishes two special square matrices, the scalar matrix and the identity matrix, which look alike but play different roles. Part (b) defines the determinant of a 2 × 2 matrix, computes one, and uses its value to decide whether the matrix is singular, the concept introduced in Question 1. Part (c) finds the inverse of a given matrix, the matrix

analogue of a reciprocal, and verifies it by multiplication. The order of the parts is the order of practice: recognise the special types, compute the determinant, and only then attempt an inverse, because the determinant decides in advance whether an inverse exists at all.

(a) Scalar Matrix versus Identity Matrix

Both matrices in this part are diagonal matrices, that is, square matrices in which every element off the main diagonal is zero, and they are distinguished entirely by what sits on the diagonal. A scalar matrix is a diagonal matrix whose diagonal elements are all equal to one and the same number, called the scalar, which may be any number at all: for example [[5, 0], [0, 5]] and [[3, 0, 0], [0, 3, 0], [0, 0, 3]] are scalar matrices, with scalars 5 and 3 respectively. The name comes from its behaviour in multiplication: multiplying any matrix by such a matrix (of a suitable size) simply multiplies every element of that matrix by the scalar, a uniform stretching. An identity matrix (or unit matrix), written I, is the special scalar matrix whose diagonal elements are all exactly 1: for example I_2 = [[1, 0], [0, 1]] and I_3 = [[1, 0, 0], [0, 1, 0], [0, 0, 1]]. Its role is the role of the number 1 in ordinary arithmetic: for any square matrix A of the same order, A × I = I × A = A, so multiplying by the identity changes nothing. Every identity matrix is a scalar matrix (the one with scalar 1), but a scalar matrix is an identity matrix only when its scalar happens to be 1 — that one-way relationship, and the contrast between “any equal diagonal value” and “diagonal value exactly 1”, together with the different jobs they do in multiplication, is the differentiation the question asks for. The table summarises the comparison.

Feature	Scalar matrix	Identity matrix
Form	Square, diagonal, all diagonal elements equal	Square, diagonal, all diagonal elements equal
Diagonal value	Any one number k (e.g. 5, 3, −2)	Exactly 1

Example (2 × 2)	[[5, 0], [0, 5]]	[[1, 0], [0, 1]]
Effect in multiplication	Multiplies every element of the other matrix by k	Leaves the other matrix unchanged: AI = IA = A
Relationship	The wider family	The member of the family with k = 1

(b) Determinant of a 2 × 2 Matrix and the Singularity Test

The determinant of a square matrix is a single number computed from its elements, written det A or |A|. For a 2 × 2 matrix A = [[a, b], [c, d]], the determinant is defined by det A = ad − bc: multiply the elements of the main diagonal (a and d), multiply the elements of the other diagonal (b and c), and subtract the second product from the first. The mnemonic "down-diagonal minus up-diagonal" makes the formula easy to retain. For the given matrix [[2, 3], [1, −5]], we have a = 2, b = 3, c = 1 and d = −5. The main-diagonal product is ad = 2 × (−5) = −10, and the other product is bc = 3 × 1 = 3, so the determinant is −10 − 3 = −13. Care with the sign of d = −5 is the one point of danger in the computation: dropping the minus sign would give −7, a different and wrong number. The determinant is −13, which is not zero, and so, by the definition established in Question 1(c), this matrix is not singular. It is non-singular, and consequently it possesses an inverse; it could serve, for example, as the coefficient matrix of a system of two equations with a unique solution, since the determinant tested by Cramer's rule in the next question is exactly this number.

(c) Inverse of [[5, 9], [4, 1]]

The inverse of a square matrix A, written A^{-1}, is the matrix (of the same order) such that $A \times A^{-1} = A^{-1} \times A = I$, the identity matrix; it is the matrix counterpart of the reciprocal of a number, since

a number times its reciprocal gives 1. For a 2 × 2 matrix the inverse is given by the formula A^{-1} = (1 / det A) × [[d, −b], [−c, a]]: swap the two diagonal elements, change the signs of the two off-diagonal elements, and divide every element by the determinant. Let A = [[5, 9], [4, 1]]. First the determinant: det A = 5 × 1 − 9 × 4 = 5 − 36 = −31. Since the determinant is not zero, the inverse exists, confirming the routine of checking first. Now swap the diagonal elements 5 and 1, and change the signs of 9 and 4, to obtain the adjugate matrix [[1, −9], [−4, 5]]. Dividing by the determinant −31 gives A^{-1} = (1/−31) × [[1, −9], [−4, 5]] = [[−1/31, 9/31], [4/31, −5/31]], which in decimals is approximately [[−0.0323, 0.2903], [0.1290, −0.1613]]. The verification by multiplication is essential and quick. $A \times A^{-1}$ has first row entries 5(−1/31) + 9(4/31) = (−5 + 36)/31 = 1 and 5(9/31) + 9(−5/31) = (45 − 45)/31 = 0, and second row entries 4(−1/31) + 1(4/31) = 0 and 4(9/31) + 1(−5/31) = (36 − 5)/31 = 1, so the product is [[1, 0], [0, 1]] = I, exactly as the definition demands. The inverse is therefore confirmed, and the fractions, not the rounded decimals, are its exact form.

Conclusion

A scalar matrix is diagonal with all diagonal elements equal to an arbitrary common value, while an identity matrix is the particular scalar matrix with diagonal value 1, acting in multiplication exactly as the number 1 does. The determinant of a 2 × 2 matrix [[a, b], [c, d]] is ad − bc; for [[2, 3], [1, −5]] it equals −13, which is not zero, so that matrix is non-singular. For [[5, 9], [4, 1]] the determinant is −31, and the inverse is [[−1/31, 9/31], [4/31, −5/31]], verified by the product with the original matrix giving the identity. Determinant first, inverse second, verification last: that sequence is the reliable routine for all matrix work.

Question No. 3

Question 3 (20 marks): (a) Describe the process of solving simultaneous linear equations using matrices. (b) Use Cramer's rule to solve the system of equations: (2x + 3y = 5) and (4x − y = 11).

Introduction

A single linear equation in two unknowns has infinitely many solutions, pictured as a line of points; two such equations together usually pin the unknowns down to one pair of values, the point where the two lines cross. Assignment 1 solved such a pair by elimination. This question presents the two systematic matrix approaches to the same problem, which generalise to any number of equations and are the methods a computer uses. Part (a) describes the general process of solving simultaneous linear equations with matrices, covering both the inverse-matrix method and the row-reduction idea, together with the cases in which no unique solution exists. Part (b) applies the third, closely related method named in the question, Cramer's rule, which expresses each unknown as a ratio of two determinants, to the system 2x + 3y = 5 and 4x − y = 11. The answer is checked by substitution into both original equations, as always. Understanding all three approaches as variations on one theme — isolate the unknowns by undoing the coefficient matrix — is the real objective.

(a) Solving Simultaneous Equations with Matrices: The Process

The process begins by rewriting the system in the matrix form AX = B. Each equation is first arranged with the unknowns in the same order on the left and the constant on the right. The coefficients of the unknowns are then collected into the coefficient matrix A, the unknowns into the column matrix X, and the constants into the column matrix B, so that the multiplication AX reproduces exactly the left-hand sides of the equations. For two equations $a_1x + b_1y = c_1$ and $a_2x + b_2y = c_2$, this gives A = [[a_1, b_1], [a_2, b_2]], X = (x, y) and B = (c_1, c_2). The second step is to

compute the determinant of A. This is the decision point of the whole method: if det A ≠ 0, the matrix A has an inverse and the system has exactly one solution, obtained in the third step; if det A = 0, A is singular, no inverse exists, and the system has either no solution or infinitely many, according as the equations represent parallel distinct lines or the same line twice, and other methods must be used to describe what happens.

In the third step, the inverse-matrix method multiplies both sides of AX = B on the left by A^{-1}. Since $A^{-1}A = I$, the left side becomes simply X, and the solution is $X = A^{-1}B$: one matrix multiplication delivers all the unknowns at once. For a 2 × 2 system the inverse is found by the formula of Question 2 (swap the diagonal elements, change the signs of the off-diagonal elements, divide by the determinant). A closely related process, used for larger systems, is row reduction: the augmented matrix [A | B], formed by attaching the constants as an extra column, is reduced by elementary row operations — interchanging rows, multiplying a row by a non-zero number, adding a multiple of one row to another — which are exactly the elimination moves of Assignment 1 performed on the whole array at once, until A has been turned into the identity; the extra column then reads off the solution. Whichever route is taken, the final step is invariable and indispensable: substitute the values found into the original equations and confirm that both are satisfied. This check distinguishes the one correct solution from the near misses produced by a sign slip, and it also confirms, in the singular case, any claim that a proposed pair works.

Quantity	How it is formed	Value
D	Determinant of the coefficient matrix |2, 3; 4, −1|	−14
Dx	Replace the x-column by the constants: |5, 3; 11, −1|	−38

Dy	Replace the y-column by the constants: |2, 5; 4, 11|	2
x = Dx/D	−38 / −14	19/7 ≈ 2.714
y = Dy/D	2 / −14	−1/7 ≈ −0.143

(b) Cramer's Rule Applied to 2x + 3y = 5 and 4x − y = 11

Cramer's rule solves a system by determinants alone. For the system $a_1x + b_1y = c_1$, $a_2x + b_2y = c_2$, let D be the determinant of the coefficient matrix, let Dx be the determinant obtained by replacing the x-coefficients in D by the constants, and let Dy be the determinant obtained by replacing the y-coefficients by the constants. Then, provided D ≠ 0, x = Dx/D and y = Dy/D. For our system the coefficient determinant is D = |2, 3; 4, −1| = 2(−1) − 3(4) = −2 − 12 = −14. Since D ≠ 0, a unique solution exists and the rule applies. Replacing the first column by the constants (5, 11) gives Dx = |5, 3; 11, −1| = 5(−1) − 3(11) = −5 − 33 = −38, so x = Dx/D = −38/−14 = 19/7 ≈ 2.714. Replacing the second column by the constants gives Dy = |2, 5; 4, 11| = 2(11) − 5(4) = 22 − 20 = 2, so y = Dy/D = 2/−14 = −1/7 ≈ −0.143. The solution is x = 19/7 and y = −1/7. The substitution check in the first equation gives 2(19/7) + 3(−1/7) = 38/7 − 3/7 = 35/7 = 5, correct; in the second equation, 4(19/7) − (−1/7) = 76/7 + 1/7 = 77/7 = 11, correct. Both equations are satisfied exactly, and the fractions, not the rounded decimals, are the exact answer. The steps are laid out in the table so that the pattern of the rule — same denominator D, numerators built by substitution of the constants — can be seen at a glance and reused.

A Note on When Each Method Is Best

The three methods of this question are not rivals so much as tools for different sizes of job. Elimination is transparent and is the fastest route for a single small system done by hand. The

inverse-matrix method is the most valuable when the same coefficient matrix must be solved against several different sets of constants — a common business situation, for instance when the same production structure is costed under several price scenarios — because the inverse is computed once and then reused. Cramer's rule has the unique virtue of giving each unknown as a separate formula, which makes it the best tool for theoretical work and for systems written with letters rather than numbers; however, its cost in determinants grows very fast with the size of the system, so for large numerical systems row reduction is preferred in practice. All three agree whenever the determinant is non-zero, because all three are doing the same thing: undoing the coefficient matrix. When the determinant is zero, all three refuse in their own way, and that refusal is itself information — the equations do not determine a single point. Recognising that unity behind the methods is what turns three recipes into one piece of understanding.

Conclusion

To solve simultaneous equations with matrices, write the system as AX = B, test the determinant of A, and, if it is non-zero, obtain $X = A^{-1}B$, or equivalently reduce the augmented matrix, finishing always with a substitution check. Cramer's rule applied to $2x + 3y = 5$ and $4x - y = 11$ uses $D = -14$, $Dx = -38$ and $Dy = 2$, giving $x = 19/7$ and $y = -1/7$, a pair verified exactly in both equations. The determinant is the gatekeeper of the whole topic: non-zero, and every method succeeds; zero, and no unique solution exists.

Question No. 4

Question 4 (20 marks): (a) Conversion: Convert (101101_2) to decimal. (b) Binary Arithmetic: Add (1101_2) and (1011_2).

Introduction

Every number a computer handles — every quantity, price, mark and total in business work done on a machine — is stored inside the machine in the binary system, the number system with base 2 whose only digits are 0 and 1. This question tests the two foundational skills of that system: translating a binary number into the familiar decimal (base 10) system, and adding two binary numbers directly, using the binary addition rules rather than converting first. The skills rest on one idea, place value. In decimal, the positions of a number count in powers of ten — units, tens, hundreds — because the base is 10. In binary, the positions count in powers of two — 1, 2, 4, 8, 16, 32 — because the base is 2. Once that single parallel is grasped, conversion is a matter of adding up the values of the positions where a 1 stands, and addition is the familiar column process with a different carry rule. Both parts are worked in full, and each is checked by the reverse operation, because binary results are easy to mistype and the decimal check catches any slip.

(a) Converting $(101101)_2$ to Decimal

Write the binary number 101101 with the place value of each digit above it. Reading from the right, the six positions carry the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$ and $2^5 = 32$. The digits of 101101, from the left, are 1, 0, 1, 1, 0, 1, standing in the positions 32, 16, 8, 4, 2, 1 respectively. The decimal value is the sum of the place values wherever the digit is 1, ignoring the positions where the digit is 0: value = $1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 32 + 0 + 8 + 4 + 0 + 1 = 45$. Hence $(101101)_2 = (45)_{10}$. The expansion is shown in the table that accompanies this section. The check is the reverse conversion, dividing 45 repeatedly by 2 and reading the remainders upwards: $45 \div 2 = 22$ remainder 1; $22 \div 2 = 11$ remainder 0; $11 \div 2 = 5$ remainder 1; $5 \div 2 = 2$ remainder 1; $2 \div 2 = 1$ remainder 0; $1 \div 2 = 0$ remainder 1; the remainders read from last to first give 101101, which is the original binary number, so the conversion is

confirmed. A rough sense check also helps: a six-digit binary number lies between 32 and 63, and 45 sits comfortably in that range, closer to its lower end, exactly as a number beginning 10… should.

Binary digit	**1**	**0**	**1**	**1**	**0**	**1**
Place value	32	16	8	4	2	1
Contribution	32	0	8	4	0	1

(b) Adding $(1101)_2$ and $(1011)_2$ in Binary

Binary addition uses the same column-by-column method as decimal addition, but the addition table has only four entries: 0 + 0 = 0; 0 + 1 = 1; 1 + 0 = 1; and 1 + 1 = 10 in binary, that is, write 0 and carry 1, because two in binary is written 10. When a carry is also present, 1 + 1 + 1 = 11 in binary, write 1 and carry 1. Align the two numbers by their rightmost digits and work from right to left: 1101 over 1011. In the units column, 1 + 1 = 10: write 0, carry 1. In the twos column, 0 + 1 plus the carried 1 = 10: write 0, carry 1. In the fours column, 1 + 0 plus the carried 1 = 10: write 0, carry 1. In the eights column, 1 + 1 plus the carried 1 = 11: write 1, carry 1. The final carry becomes a new leading digit, 1. Reading the written digits from the last to the first, the sum is 11000, so $(1101)_2 + (1011)_2 = (11000)_2$. The decimal check is decisive: $(1101)_2 = 8 + 4 + 0 + 1 = 13$ and $(1011)_2 = 8 + 0 + 2 + 1 = 11$, and $13 + 11 = 24$; converting the answer, $(11000)_2 = 16 + 8 = 24$. The two routes agree exactly, which is the confirmation that the carries were handled correctly. The example is also a neat warning about the commonest binary error: treating 1 + 1 as 2 and writing the decimal digit 2 into a binary result, where the digit 2 can never appear; in binary the pair must always be rewritten as 0 with a carry of 1 into the next column.

Why Binary Arithmetic Matters

It is worth pausing on why a business mathematics course includes this material at all. The binary system is used by computers because a digit with only two values can be represented by a physical device with two states — a switch off or on, a voltage low or high, a spot magnetised one way or the other — and such devices are cheap, fast and reliable. Every spreadsheet total, bank balance and computerised account that a business student will ever meet is, underneath, binary numbers being added exactly as in part (b), millions of times per second. Conversion, as in part (a), is the bridge that lets a human read what the machine is holding: the machine stores 101101, the screen shows 45. The rules themselves never change with size; a 64-digit binary number is converted by the same place values and added by the same four-entry table, only with more columns. Mastering the small examples of this question therefore amounts to mastering the arithmetic of the machine age in miniature, and the habit of checking in decimal gives a permanent safety net for hand work.

Conclusion

By place values, $(101101)_2 = 32 + 8 + 4 + 1 = 45$ in decimal, confirmed by converting 45 back to binary. By the binary addition rules, with a carry wherever a column totals two or three, $(1101)_2 + (1011)_2 = (11000)_2$, confirmed by the decimal check $13 + 11 = 24 = (11000)_2$. Place value explains conversion, the carry rule explains addition, and converting to decimal explains how to check both — three ideas that together cover the binary arithmetic of this course.

Question No. 5

Question 5 (20 marks): (a) Find the product of the matrices [[5, 0], [5, 1]] and [[4, 10], [9, −8]]. (b) Discuss how the letter A is 11010001 in ASCII (a binary-based encoding). (c) Adding

two numbers in binary: Suppose a computer adds two decimal numbers 3 and 2; find the sum of these two numbers in binary.

Introduction

The closing question of the course returns to matrix multiplication, the most powerful and the most error-prone of the matrix operations, and then ties the matrix work back to the binary theme of Question 4 by looking inside the computer: how a letter is stored as a binary code, and how the machine adds two small decimal numbers in binary. Part (a) multiplies two 2 × 2 matrices by the row-by-column rule and verifies the result entry by entry. Part (b) discusses ASCII, the standard binary-based code in which text is stored, and it contains a misprint in the booklet which is identified and handled openly in a transparency note, with the correct standard codes given and proved. Part (c) performs, in binary, exactly the addition a computer performs when it is asked for 3 + 2. Together the three parts round off the course: arrays of numbers, codes for characters, and arithmetic in base 2 are the three forms in which a computer holds the business data this course teaches students to manipulate by hand.

(a) The Product of the Two Matrices

Matrix multiplication is defined by the row-by-column rule: the element in row i and column j of the product is obtained by multiplying the elements of row i of the first matrix, in order, by the corresponding elements of column j of the second matrix, and adding the results. A product AB exists only when the number of columns of A equals the number of rows of B; here both matrices are 2 × 2, so the product exists and is itself 2 × 2. Let A = [[5, 0], [5, 1]] and B = [[4, 10], [9, −8]]. The element in row 1, column 1 of AB pairs the first row of A, (5, 0), with the first column of B, (4, 9): 5 × 4 + 0 × 9 = 20 + 0 = 20. The element in row 1, column 2 pairs (5, 0) with the second column of B, (10, −8): 5 × 10 + 0 × (−8) = 50 + 0 = 50. The element in row 2,

column 1 pairs the second row of A, (5, 1), with the first column of B: 5 × 4 + 1 × 9 = 20 + 9 = 29. The element in row 2, column 2 pairs (5, 1) with (10, −8): 5 × 10 + 1 × (−8) = 50 − 8 = 42. The product is therefore AB = [[20, 50], [29, 42]]. Two observations are worth making. First, the zero in A simplified the whole of the first row of the product, a pattern worth noticing before calculating, since predicting easy entries guards against slips. Second, matrix multiplication is not commutative: multiplying in the reverse order gives BA = [[70, 10], [5, −17]], an entirely different matrix, so the order in which the question names the matrices must be respected. The row-by-column workings are set out in the table for reference.

Entry of AB	Row of A × column of B	Value
Row 1, column 1	5 × 4 + 0 × 9	20
Row 1, column 2	5 × 10 + 0 × (−8)	50
Row 2, column 1	5 × 4 + 1 × 9	29
Row 2, column 2	5 × 10 + 1 × (−8)	42

(b) Letters in ASCII — and a Printed Error, Handled Openly

ASCII, the American Standard Code for Information Interchange, is the classic scheme by which a computer stores text: every character — letter, digit, punctuation mark and certain control signals — is assigned a fixed decimal code number from 0 to 127, and that number is stored in binary, nowadays usually in one byte of eight bits. The capital letters run in order from A = 65 to Z = 90, and the small letters from a = 97 to z = 122, so the code of any letter can be found by counting from A. In binary, the capital letter A, code 65, is 1000001 in seven bits, written in a full byte as 01000001, since 65 = 64 + 1. When a user presses the A key, it is this pattern of bits that travels through the machine and is stored in memory; when the character is displayed or

printed, the pattern is looked up in the ASCII table and the letter A is drawn. This is the sense in which a letter "is" a binary number in a computer: text is not stored as shapes but as numbers, and the binary arithmetic of Question 4 is the arithmetic performed on those numbers. The discussion the question invites is therefore about encoding itself: a code is an agreed table between symbols and numbers, the numbers are held in binary, and every text operation a computer performs — sorting names alphabetically, for example, which is just comparing their codes — is arithmetic on codes.

Transparency note on the printed question. part (b) is printed in the booklet in exactly these words — "Discuss how the letter A is 11010001 in ASCII (a binary-based encoding)." The eight-bit string printed there, 11010001, is not the ASCII code for the letter A, and it cannot be an ASCII code for any character: read as a binary number it equals 128 + 64 + 16 + 1 = 209, whereas ASCII codes run only from 0 to 127. The correct codes, as explained above, are: capital A = 65 decimal = 1000001 (seven bits) = 01000001 (one byte); for comparison, small a = 97 decimal = 1100001 = 01100001. The printed string is evidently a typesetting slip — most plausibly for 01000001, the byte form of A — in which a digit has been altered. This guide has therefore done three things rather than silently copying the error: it has quoted the booklet's wording exactly here, it has given and proved the correct standard codes above ($65 = 64 + 1 = 1000001_2$), and it has answered the discussion the question intends, namely how a letter is represented by a binary code in ASCII. Students writing this answer in their own hand are advised to present the correct code for A, and may add a brief note that the booklet's printed string appears to be a misprint, and to confirm the point with their tutor if in doubt. No answer that repeated "11010001" as the code for A would be correct, and presenting a wrong code without comment would mislead anyone who used this guide for revision.

Character	ASCII decimal code	Binary (byte)
A (capital)	65	01000001
a (small)	97	01100001
B (capital)	66	01000010
Digit 0	48	00110000
Space	32	00100000

(c) Adding the Decimal Numbers 3 and 2 in Binary

A computer cannot add the decimal numerals 3 and 2 directly, because it holds only binary digits; it first represents each number in binary, adds by the binary rules, and the result is the binary form of the sum. The decimal number 3 in binary is 11, since 3 = 2 + 1, that is, one “two” and one “one”. The decimal number 2 in binary is 10, since 2 is a single “two” with no “one”. Adding in columns, right-aligned: in the units column, 1 + 0 = 1, write 1 with no carry. In the twos column, 1 + 1 = 10 in binary: write 0 and carry 1. The carry becomes a new leading digit in the fours column. The result is therefore 101. So the sum of the two numbers in binary is $(101)_2$. The check by place values confirms it: $(101)_2 = 4 + 0 + 1 = 5$, and $3 + 2 = 5$ in decimal, so the binary addition is correct. Although the numbers are tiny, exactly the same routine — convert, add column by column with carries, read off the binary result — is what the processor does with numbers of any size, and part (c) is really Question 4(b) seen from the computer's side of the screen: there the binary digits were given, here the machine's user supplies decimal numbers and the machine's internal work is reproduced. Noticing that the carry in the twos column is what creates the new fours digit is the key insight, since it is the binary counterpart of carrying in decimal when a column reaches ten.

Conclusion

By the row-by-column rule, [[5, 0], [5, 1]] × [[4, 10], [9, −8]] = [[20, 50], [29, 42]], and the reverse product is different, confirming that matrix multiplication depends on order. In ASCII a character is stored as a fixed decimal code written in binary. the letter A is code 65, that is 1000001, or 01000001 in a byte; the string “11010001” printed in the booklet equals 209 and is identified above, transparently, as a misprint. Finally, 3 and 2 are 11_2 and 10_2, and their binary sum is 101_2, equal to the decimal 5. With matrices multiplied, characters encoded and binary addition performed from decimal inputs, the course's three computational themes are brought together in one closing answer.

USMAN NOTES www.usmannotes.com