Concepts:

- Define "base-b expansion" of an integer.
- Convert integers from decimal to binary, octal, hexadecimal and vice versa.
- Convert integers from decimal to any base and vice versa.
- Convert integers directly between binary, octal and hexadecimal by grouping digits.
- Determine numbers of digits of base-b expansions of given integers, and ranges of numbers represented in base-b by a given number of digits.
- Express the information that n has a last digit k in base-b as an equation, using the division algorithm.
- Carry out addition and multiplication directly in different base systems.
- Perform multiplication and division by 2 on binary numbers by bit shifting.
- Carry out the "slow" and "fast" modular exponentiation algorithms.

Problems:

1. Check whether the following statements are true or false.

   (a) To convert a binary number to an octal number we can group 3 binary bits together, adding 0's to the front if it is necessary, and block convert each group of 3 binary digits to an octal digit.

   (b) To convert a octal number to a binary number we can reverse the procedure and convert each octal digit to a block of three bits together.

   (c) To convert a binary number to a decimal number we can group 3 binary bits together, adding 0's to the front if it is necessary, and block convert each group of 3 binary digits to a decimal digit.

2. Complete the following conversions.

   (a) $(34267)_{10}$ to duodecimal (base 12) using repeated application of the division algorithm.

   (b) $(A57B)_{16}$ to decimal.

   (c) 1011001 binary number to octal and to a hexadecimal number.

   (d) $A9F5$ hexadecimal number to binary.

   (e) The hexadecimal number $F7A5_{16}$ to an octal number using binary conversion as an intermittent step.

3. Evaluate the following arithmetic operations.

   (a) The sum of the hexadecimal numbers $BA5 + 8FC$. Do not convert the hexadecimal numbers to decimal.

   (b) The sum and the product of 1 1000 1001 and 1000 0001. Do not convert the binary numbers to decimal.

(c) The sum of $(510567)_7$ and $(063104)_7$. Do not convert to decimal base.

4. Use fast modular exponentiation to evaluate $5^{1349} \mod 13$. Show all your steps and take advantage of repetition of remainders.

5. Find the decimal value of the octal number $(\underbrace{33\ldots3}_{10\text{ digits}})_8$. (Hint: Use the formula for the sum of the terms of a geometric sequence.)

6. Find the base 52 representation of the following decimal numbers

   (a) 52.

   (b) 52^2.

7. Find the base n representation of the decimal number n^k, where n and k are positive integers.

8. Multiply

   (a) $(24306)_3$ by $(10)_3$.

   (b) the octal number $(257)_8$ by $(100)_8$.

9. What is the remainder and the quotient when you carry out the division algorithm and divide

   (a) $(24306)_3$ by $(10)_3$?

   (b) the hexadecimal number $2A4B$ by $(100)_{16}$?

10. Find the number of binary digits of the decimal number

    (a) 32768.

    (b) 35379.

11. Find the hexadecimal representation of the decimal number

    (a) $2^{32} - 1$.

    (b) $2^{32} - 2$.

12. In general, what is the minimum number of bits required to store

    (a) the sum of any two 8-bit binary numbers?

    (b) the product of any two 8-bit binary numbers?

Solutions:

1. Check whether the following statements are true or false.

   (a) To convert a binary number to an octal number we can group 3 binary bits together, adding 0's to the front if it is necessary, and block convert each group of 3 binary digits to an octal digit.

   True. For example: The binary number $(110\,010)_2$ is $(62)_8$ in octal system.

   (b) To convert a octal number to a binary number we can reverse the procedure and convert each octal digit to a block of three bits together.

   True. For example: The octal number $(357)_8$ is $(011\,101\,111)_2$ in binary system.

   (c) To convert a binary number to a decimal number we can group 3 binary bits together, adding 0's to the front if it is necessary, and block convert each group of 3 binary digits to a decimal digit.

   False. The binary number $(110\,010)_2$ is $(62)_8$ in octal system and 50 in decimal system.
   In general, we can use block conversion from base-b to base-b^k by grouping k digits together and block-converting each group of k digits into a b^k digit.

2. Complete the following conversions.

   (a) $(34267)_{10}$ to duodecimal (base 12) using repeated application of the division algorithm.

   We apply the division algorithm with divisor 12:

$$\begin{aligned} 34267 &= 12 \cdot 2855 + 7 \\ 2855 &= 12 \cdot 237 + 11 \\ 237 &= 12 \cdot 19 + 9 \\ 19 &= 12 \cdot 1 + 7 \\ 1 &= 12 \cdot 0 + 1 \end{aligned}$$

   Thus, $(34267)_{10} = (179B7)_{12}$.

   (b) $(A57B)_{16}$ to decimal.

   $10 \cdot 16^3 + 5 \cdot 16^2 + 7 \cdot 16 + 11 = 42363$ in decimal.

   (c) 1011001 binary number to octal and to a hexadecimal number.

   We block convert a group of 3 bits to an octal digit. Thus, $(001\,011\,001)_2$ is $(131)_8$.
   We block convert a group of 4 bits to a hexadecimal digit to obtain $(0101\,1001)_2$ is $(59)_{16}$.

   (d) $A9F5$ hexadecimal number to binary.

   We convert each hexadecimal digit to a block 4 binary digits (bits). Thus, $(A9F5)_{16}$ is $(1010\,1001\,1111\,0101)_2$.

(e) The hexadecimal number $F7A5_{16}$ to an octal number using binary conversion as an intermittent step.

We use reverse block conversion from base 16 to binary and block conversion from binary to octal. $(F7A5)_{16}$ is $(1111\,0111\,1010\,0101)_2$. Then we group the bits into groups of three bits, $(001\,111\,011\,110\,100\,101)_2$ and convert each group into an octal digit $(173645)_8$.

3. Evaluate the following arithmetic operations.

(a) The sum of the hexadecimal numbers $BA5 + 8FC$. Do not convert the hexadecimal numbers to decimal.

We add digits individually from right to left and make sure to carry the overflow: $5 + 12 = 17$ and $17 = 16 + 1$, thus we get the digit 1 we carry 1 into the next addition. $1 + A + F = 26$ and $26 = 16 + 10$, thus we get the digit A and carry over 1 into the next addition. $1 + B + 8 = 20$ and $20 = 16 + 4$, which gives the digit 4 and carry over 1.
Hence, $BA5 + 8FC = 14A1$.

(b) The sum and the product of 1 1000 1001 and 1000 0001. Do not convert the binary numbers to decimal.

The sum is given as:

$$\begin{array}{r} 1\,0000\,0010 \\ 1\,1000\,1001 \\ +0\,1000\,0001 \\ =10\,0000\,1010 \end{array}$$

The product is given as:
Since the second factor 1000 0001 has two 1's and eight bits, it is more efficient to multiply 1 1000 1001 by 1000 0001.

$$\begin{aligned} 1\,1000\,1001 \cdot 1000\,0001 &= 1100\,0100\,1000\,0000 \\ &\;+1\,1000\,1001 \\ &= 1100\,0110\,0000\,1001 \end{aligned}$$

Note that the number of additions needed equals the number of 1's in the multiplier, minus one. The number of left bit shifts needed is the number of the digits of the multiplier, minus one.

This multiplication needed 1 addition and 7 left bit shifts.
If we carry out the multiplication in reverse order, as 1000 0001 is multiplied by 1 1000 1001, then we need 3 additions and 8 left bit shifts.

(c) The sum of $(510567)_7$ and $(063104)_7$. Do not convert to decimal base.

We add digits individually from right to left and make sure to carry overflow. The overflow is indicated with red color on the top.

$$\begin{array}{r} 101110 \\ 510567 \\ +063104 \\ =604004 \end{array}$$

4. Use fast modular exponentiation to evaluate 5^{1349} mod 13. Show all your steps and take advantage of repetition of remainders.

   First we find the binary representation of $1349 = 2^{10} + 2^8 + 2^6 + 2^2 + 1$, then we perform the algorithm. We successively square base 5 ten times, in each step reduce the answer mod 13, until we obtain 5^{1024}

   $$5^{2^0} \text{ mod } 13 = 5^1 \text{ mod } 13 = 5$$

   $$5^{2^1} \text{ mod } 13 = 5^2 \text{ mod } 13 = 12$$

   In the second step, we obtain the result by squaring 5 instead of calculating 5^{2^1} mod 13.

   $$5^{2^2} \text{ mod } 12 = 12^2 \text{ mod } 13 = 1$$

   In the third step, we obtain the result by squaring 12 instead of calculating 3^{2^2} mod 13.

   $$3^{2^3} \text{ mod } 13 = 1^2 \text{ mod } 7 = 1$$

   In the fourth step we obtain the result by squaring 1 and not by calculating 3^{2^3} mod 7.

   Thus, from now on the remainders will be repeating, each remainder will be 1.

   5^{1349} mod $13 = 5^{2^{10}+2^8+2^6+2^2+1}$ mod $13 = ((5^{2^{10}}$ mod $13) \cdot (5^{2^8}$ mod $13) \cdot (5^{2^6}$ mod $13) \cdot (5^{2^2}$ mod $13) \cdot (5^{2^0}$ mod $13))$ mod $13 = (1 \cdot 1 \cdot 1 \cdot 1 \cdot 5)$ mod $13 = 5$.

   Thus, 5^{1349} mod $13 = 5$.

5. Find the decimal value of the octal number $(\underbrace{33\ldots3}_{10 \text{ digits}})_8$. (Hint: Use the formula for the sum of the terms of a geometric sequence.)

   The octal number $(\underbrace{33\ldots3}_{10 \text{ digits}})_8$ in decimal system is $\sum_{k=0}^{9} 3 \cdot 8^k = 3 \cdot \frac{8^{10}-1}{7} = 460\,175\,067$.

6. Find the base 52 representation of the decimal numbers

   (a) 52.

   The decimal number 52 in base 52 is 10.

   (b) 52^2.

   The decimal number 52^2 in base 52 is 100.

7. Find the base n representation of the decimal number n^k, where n and k are positive integers.

   The decimal number n^k in base n is $1\underbrace{00\ldots0}_{k \text{ zeros}}$.

8. Multiply

   (a) $(24306)_3$ by $(10)_3$.

   Using base 3 expansion, $\underbrace{(2 \cdot 3^4 + 4 \cdot 3^3 + 3 \cdot 3^2 + 0 \cdot 3^1 + 6 \cdot 3^0)}_{=(24306)_3} \cdot \underbrace{(1 \cdot 3^1)}_{=(10)_3} = \underbrace{2 \cdot 3^5 + 4 \cdot 3^4 + 3 \cdot 3^3 + 0 \cdot 3^2 + 6 \cdot 3^1}_{=(243060)_3}$. Thus, $(24306)_3 \cdot (10)_3 = 243060$. For any base $b > 1$, multiplying a number by $(10)_b$ will add a "0" to the end of that number.

   (b) the octal number $(257)_8$ by $(100)_8$.

   Using octal expansion, $\underbrace{(2 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0)}_{=(257)_8} \cdot \underbrace{(1 \cdot 8^2)}_{=(100)_8} = \underbrace{2 \cdot 8^4 + 5 \cdot 8^3 + 7 \cdot 8^2}_{=(25700)_8}$. Thus, $(257)_8 \cdot (100)_8 = 25700$. For any base $b > 1$, multiplying a number by $(100)_b$ will add "00" to the end of that number.

9. What is the remainder and the quotient when you carry out the division algorithm and divide

   (a) $(24306)_3$ by $(10)_3$?

   $(24306)_3 = (2430)_3 \cdot (10)_3 + 6$. Thus, the quotient is $(2430)_3$ and the remainder is 6. For any base $b > 1$, if a number is divided by $(10)_b$, the quotient will be the same number without the last digit (a right digit shift) and the remainder will be the last digit of that number.

   (b) the hexadecimal number $2A4B$ by $(100)_{16}$?

   $(2A4B)_{16} = (2A)_{16} \cdot (100)_{16} + 4B$. Thus, the quotient is $(2A)_{16}$ and the remainder is $4B$. For any base $b > 1$, if a number is divided by $(100)_b$, the quotient will be the same number without the last two digits (two right digit shifts) and the remainder will be the last two digits of that number.

10. Find the number of binary digits of the decimal number

    (a) 32768.

    Since $32768 = 2^{15}$, the decimal number 32768 in binary system is $1\underbrace{00\ldots0}_{15\text{ zeros}}$. Thus, 32768 has 16 $(\log_2 32768 + 1)$ digits (bits) in binary system.

    (b) 35379.

    Since $35379 = 2^{15} + 2^{11} + 2^9 + 2^5 + 2^4 + 2 + 1$, the decimal number 35379 in binary system is 1000 1010 0011 0011. Thus, 35379 has 16 $(\lfloor \log_2 35379 \rfloor + 1)$ digits (bits) in binary system.

    In general, the size of a positive integer n in base b, $b > 1$, is $(\lfloor \log_b n \rfloor + 1)$.

11. Find the hexadecimal representation of the decimal number

    (a) $2^{32} - 1$.

    The binary representation of the decimal number $2^{32} - 1$ is $\underbrace{111\ldots1}_{32\text{ ones}}$, since $\sum_{k=0}^{31} 2^k = 2^{32} - 1$. Thus, grouping 4 bits together, we obtain the hexadecimal representation FFFF FFFF.

(b) $2^{32} - 2$.

Since $2^{32} - 2 = 2 \cdot (2^{31} - 1)$, the 2 multiple will produce a left bit shift and add a "0" to the end of binary representation of $2^{31} - 1$. Thus, the binary representation of $2^{32} - 2$ is $\underbrace{1111\ldots111}_{\text{31 ones}}0$. Now, we group 4 bits to obtain the hexadecimal form FFFF FFFE. Notice that we could have received the same result by subtracting 1 from FFFF FFFF.

12. In general, what is the minimum number of bits required to store

(a) the sum of any two 8-bit binary numbers?

We need minimum 9 bits considering that the sum could produce an overflow. Let x and y be two 8-bit numbers in decimal form. Then $0 \leq x \leq 2^8 - 1$, and a similar inequality holds for y. Thus, the sum $0 \leq x + y \leq 2^9 - 2$, and $2^9 - 2$ is the 9-bit number 1 1111 1110. Note that the sum of any two 8-bit numbers is not more than 1 1111 1110 and 1111 11111 + 1111 11111 = 1 1111 1110.

(b) the product of any two 8-bit binary numbers?

We need minimum 16 bits considering overflows. Let x and y be two 8-bit numbers in decimal form. Then $0 \leq x \leq 2^8 - 1$ and a similar inequality holds for y. Thus, the product $0 \leq x \cdot y \leq 2^{16} - 2^9 + 1$. The binary representation of the decimal number $2^{16} - 2^9 + 1 = 2^9(2^7 - 1) + 1$ is 1111 1110 0000 0001. The product of any two 8-bit numbers is not more than 1111 1110 0000 0001 which is the product of 1111 11111 and 1111 1111.