Computer Science by AP Tutorial

A class under Aryan Singh

Notes on Number Systems, ASCII, Algorithms, and Problem Solving

Number Systems

Number systems are mathematical notations used to represent numbers. Common types include:

1. **Decimal System (Base-10)**:

    - The most commonly used system in daily life.
    - Digits range from 0 to 9.

2. **Binary System (Base-2)**:

    - Used in computers and digital systems.
    - Digits are 0 and 1.

3. **Octal System (Base-8)**:

    - Uses digits from 0 to 7.
    - Often used in computing as a shorthand for binary numbers.

4. **Hexadecimal System (Base-16)**:

    - Uses digits 0-9 and letters A-F.
    - Commonly used in programming and memory addressing.

Conversions:

- Decimal to Binary: Divide by 2 and record remainders.
- Binary to Decimal: Multiply each bit by $$2^n$$ (where $$n$$ is the position index).
- Similar methods apply for octal and hexadecimal conversions.

ASCII (American Standard Code for Information Interchange)

ASCII is a character encoding standard used in computing to represent text.

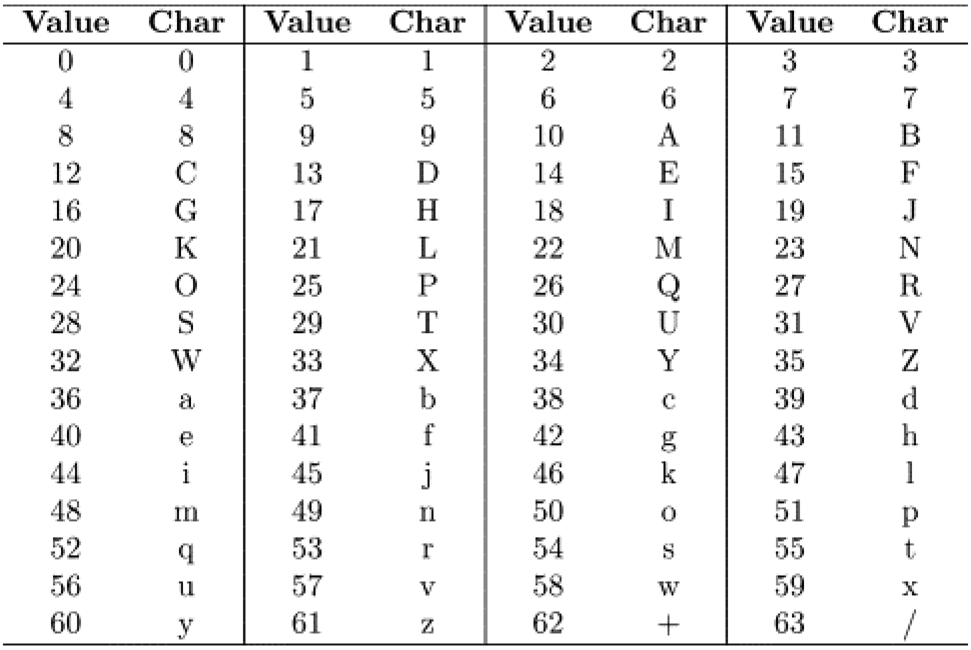
1. **Basic Features**:

    - Maps 128 characters (0–127) to numbers.
    - Includes control characters (e.g., newline, tab) and printable characters (letters, digits, symbols).

2. **Extended ASCII**:

    - Expands the set to 256 characters (0–255).
    - Includes additional symbols and non-English characters.

3. **Applications**:
    - Used in text files, programming languages, and data transmission protocols.
4. **Table for reference**



Value	Char	Value	Char	Value	Char	Value	Char
0	0	1	1	2	2	3	3
4	4	5	5	6	6	7	7
8	8	9	9	10	A	11	B
12	C	13	D	14	E	15	F
16	G	17	H	18	I	19	J
20	K	21	L	22	M	23	N
24	O	25	P	26	Q	27	R
28	S	29	T	30	U	31	V
32	W	33	X	34	Y	35	Z
36	a	37	b	38	c	39	d
40	e	41	f	42	g	43	h
44	i	45	j	46	k	47	l
48	m	49	n	50	o	51	p
52	q	53	r	54	s	55	t
56	u	57	v	58	w	59	x
60	y	61	z	62	+	63	/

Figure 1: ASCII table showing a subset of ascii table only. There are more characters in the table

Example:

- ASCII code for 'A' is 65.
- ASCII code for 'a' is 97.

Algorithms

An algorithm is a step-by-step procedure to solve a problem or perform a task.

1. **Characteristics of Good Algorithms**:
    - Finite: Must terminate after a finite number of steps.
    - Clear: Each step should be unambiguous.
    - Efficient: Minimizes time and space complexity.
2. **Example Algorithm (Finding Maximum Number)**:

```
Start
  let the first number in the list be called Max
  for each Number in List
    if Number > Max
      Update Max := Number

  Print Max
End
```

Problem Solving

Problem-solving involves identifying solutions to challenges using logical reasoning and structured approaches.

1. **Steps in Problem Solving**:
    - Understand the Problem: Identify inputs, outputs, constraints.
    - Plan the Solution: Break the problem into smaller parts.
    - Implement the Solution: Write code or perform calculations.
    - Test the Solution: Verify correctness with different cases.
2. **Techniques**:
    - Brainstorming, flowcharts or pseudocode.
    - Debugging errors systematically during implementation.
3. **Example Problem-Solving Approach**: *Problem*: Find the sum of all even numbers in a list. *Solution*:

```
Start
  for each Number in the list
    if Number is divisble by 2
      then Number is even
    else
      Number is odd
End
```