

MYCOLLEGEVERSE.IN

BUILDING ACADEMIC IDENTITY · THE STUDENT OS

AKTU · B.TECH 1ST YEAR · PPS (C)

Programming for Problem Solving – Smart Prep Doc

PPS (BCS101/BCS201) exam-oriented notes based on recent 5–6 years AKTU papers and latest important-topic lists.

C LANGUAGE PYQ PATTERN SCORE 40+ / 50

Start with loops and number patterns, then arrays and functions, and finally pointers, structures, DMA and file handling. Ye doc tumhe direct scoring topics pe focus karne me help karega.

What You Get

- Unit-wise most repeated topics for PPS (C).
- List of must-do programs that appear again and again in AKTU exams.
- Smart steps to reach 40+ marks even with little time.

Contents

PPS units arranged by exam priority, not just syllabus order.

Unit 1 – Basics & Logic

Algorithms, flowcharts, languages, data types, operators.

Unit 2 – Control & Loops ★★★

if-else, switch, loops, classic number programs.

Unit 3 – Functions & Recursion

★★

Functions, parameter passing, recursion patterns.

Unit 4 – Arrays, Strings, Sorting

★★★

Unit 5 – Pointers, Structures, DMA, Files ★★

1D/2D arrays, strings, searching,
sorting.

Pointer basics, structures,
malloc/calloc, file I/O.

Unit 2 – Control Statements & Loops

WEIGHTAGE: ★★★ · MOST SCORING

MUST-KNOW CONCEPTS

- Decision statements: if, if-else, nested if, switch with menu-driven examples.
- Loops: for, while, do-while – syntax, flow and small examples.
- Jump statements: break, continue, goto – exam theory + short examples.

MOST REPEATED PROGRAMS

- Check whether a number is **prime**.
- Check whether a number is **palindrome**.
- Check whether a number is an **Armstrong** number.
- Generate **Fibonacci series** up to n terms.
- Find **factorial** of a number using loop.
- Find **sum of digits** of a number.
- Menu-driven **simple calculator** using switch.

Prime Number Pattern

```
int n, i, flag = 0;
printf("Enter a number: ");
scanf("%d", &n);
for (i = 2; i <= n/2; i++) {
    if (n % i == 0) {
        flag = 1;
        break;
    }
}
if (flag == 0 && n > 1)
    printf("Prime");
else
    printf("Not prime");
```

Exam Hints

- Har program me clear input-output messages likho (user-friendly).
- Loop ke andar sahi initialization/update check karo – off-by-one error se bach.
- Comments se logic explain karo (2–3 lines max) for better presentation.

Unit 4 – Arrays, Strings, Searching & Sorting

WEIGHTAGE: ★★★ · FREQUENTLY ASKED

KEY EXAM TOPICS

- 1D array: input, output, sum, average, largest and smallest element.
- 2D array (matrix): addition, subtraction and multiplication programs.
- String handling: length, copy, compare, reverse using library and without library.
- Searching: linear search and binary search in an array.
- Sorting: bubble sort program (very high frequency), selection sort sometimes.

STANDARD ASKED PROGRAMS

- Find largest element in an array.
- Search an element using linear search / binary search.
- Sort an array of numbers using bubble sort.
- Multiply two matrices (check order before multiplication).
- Reverse a string and check palindrome string.

Bubble Sort Pattern

```
int a[50], n, i, j, temp;
printf("Enter n: ");
scanf("%d", &n);
for (i = 0; i < n; i++)
    scanf("%d", &a[i]);

for (i = 0; i < n-1; i++) {
    for (j = 0; j < n-1-i; j++) {
        if (a[j] > a[j+1]) {
            temp = a[j];
            a[j] = a[j+1];
            a[j+1] = temp;
        }
    }
}
printf("Sorted array: ");
for (i = 0; i < n; i++)
    printf("%d ", a[i]);
```

Matrix Multiplication Tips

- Check condition: columns of first matrix = rows of second matrix.
- Use three nested loops: i (rows), j (columns), k (for multiplication and sum).
- Initialize result matrix elements to 0 before inner loop.

Unit 3 – Functions & Recursion

WEIGHTAGE: ★★

IMPORTANT CONCEPTS

- Function definition, declaration and calling.
- Return type, parameters, local and global variables.
- Call by value vs call by reference (pointer use).
- Recursion: idea, base case, recursive case.

COMMON PROGRAMS

- Factorial of a number using recursion.
- Fibonacci series using recursion.
- Sum of digits using recursion.
- Swapping two numbers using call by reference (pointers).

Factorial Using Recursion

```
int fact(int n) {
    if (n == 0 || n == 1)
        return 1;
    else
        return n * fact(n - 1);
}
```

Exam Hints

- Recursion me base case clearly likhna zaroori hai, warna infinite recursion ho sakti hai.
- Call by reference wale program me address pass karo, pointer se value change karo.

Unit 5 – Pointers, Structures, DMA & Files

WEIGHTAGE: ★★

KEY THEORY POINTS

- Pointer definition, declaration, initialization, * and & operators.
- Pointers and arrays, pointer arithmetic basics.
- Structures: definition, declaration, accessing members, array of structures.
- Difference between structure and union.
- Dynamic Memory Allocation: malloc, calloc, realloc, free.
- File handling: FILE*, fopen, fprintf, fscanf, fclose.

TYPICAL QUESTION STYLES

- Explain pointers in C with examples. Why are they needed?

- Write a program to store and display information of students using structure.
- Write short notes on malloc, calloc, realloc and free.
- Write a program to create a text file and write some content into it.

Simple File Write Pattern

```
FILE *fp;
fp = fopen("data.txt", "w");
if (fp == NULL) {
    printf("File not created");
} else {
    fprintf(fp, "Hello PPS!");
    fclose(fp);
}
```

DMA Pattern

```
int *arr, n, i;
printf("Enter n: ");
scanf("%d", &n);
arr = (int *)malloc(n * sizeof(int));
if (arr == NULL) {
    printf("Memory not allocated");
} else {
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
}
free(arr);
```

Unit 1 – Basics, Algorithms & Flowcharts

WEIGHTAGE: ★ (EASY THEORY)

OFTEN ASKED THEORY

- Definition of algorithm and flowchart; difference between them.
- Advantages and limitations of flowcharts.
- Types of programming languages: machine, assembly, high-level.
- Compiler vs interpreter.
- Data types in C and ranges; types of operators and precedence.

Quick Tip

- Ye unit se short answers aate hain, jinhe rat ke 10–15 minute revision se pakka marks milte hain.
- At least ek simple flowchart practice karo (sum of first n natural numbers jaisa).

Score 40+ / 50 Strategy

Use this simple sequence if exam nazdeek hai.

PRIORITY STEPS

1. Day 1–2: Master all loop-based programs (prime, palindrome, Armstrong, factorial, Fibonacci, sum of digits).
2. Day 3: Do arrays and sorting (largest in array, search, bubble sort, matrix multiplication).
3. Day 4: Functions and recursion (factorial, Fibonacci, sum using recursion; call by reference).
4. Day 5: Pointers, structures, DMA and simple file program.
5. Last day: Only theory revision (algorithm vs flowchart, data types, operators) + PYQ practice.

MYCOLLEGEVERSE RESOURCES

- [MyCollegeVerse Notes](#) – more subject-wise notes and prep docs.
- [MyCollegeVerse.in](#) – explore tools, notes and academic utilities.
- [Resume Builder](#) – use after 1st year to build your resume fast.

© MyCollegeVerse.in – The Student OS

Designed as a smart prep doc for AKTU B.Tech PPS (Programming for Problem Solving)