

**Malla Reddy College of Engineering & Technology**

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with ‘A’ Grade)

**C Programming Guide**

**C Technical Questions**

**1. Q: What is the difference between compiler and interpreter?**

A: Compiler converts the entire program into machine code at once. Interpreter converts and executes code line-by-line.

**2. Q: Explain the structure of a C program.**

A: Includes headers, main() function, declarations, statements, and return statement.

**3. Q: What are keywords in C?**

A: Reserved words with special meaning (e.g., int, return).

**4. Q: Define a variable and constant in C.**

A: Variable: memory location with name and value. Constant: fixed value (e.g., #define PI 3.14).

**5. Q: Explain data types in C.**

A: int, float, char, double, etc. determine type of data a variable can hold.

**6. Q: What is the precedence of operators?**

A: Determines the order of evaluation. E.g., \*, / have higher precedence than +, -.

**7. Q: What is type conversion?**

A: Converting one data type into another. Implicit and explicit.

**8. Q: Difference between = and ==?**

A: = is assignment, == is comparison.

**9. Q: What is the use of printf and scanf?**

A: printf: output, scanf: input.

**10. Q: What are control structures?**

A: Direct flow of execution: if, else, switch, loops.

**11. Q: Explain if-else and switch with example.**

A: if-else evaluates condition; switch handles multiple constant cases.

**12. Q: Difference between while and do-while loop?**

A: while checks condition before loop; do-while executes once before checking.

**13. Q: What is the purpose of break and continue?**

A: break exits loop, continue skips to next iteration.

**14. Q: What is a goto statement?**

A: Transfers control to a labeled statement. Use cautiously.

**15. Q: What are arrays?**

A: Collection of elements of same type stored in contiguous memory.

**16. Q: How to declare and initialize a 2D array?**

A: int arr[2][2] = {{1,2},{3,4}};

**17. Q: What are functions?**

A: Block of code for a specific task, reused by calling.

**18. Q: What is recursion?**

A: Function calling itself. Base condition is required.

**19. Q: Call by value vs Call by reference?**

A: Value: copy passed; Reference: address passed.

**20. Q: Scope of variables?**

A: Local (within function), Global (entire program).

**21. Q: What are storage classes in C?**

A: auto, extern, static, register - define scope, lifetime, visibility.

**22. Q: What are strings in C?**

A: Array of characters ending with null ('\0').

**23. Q: Difference between gets() and scanf()?**

A: gets() reads entire line; scanf() stops at whitespace.

**24. Q: Name string manipulation functions.**

A: strcpy, strcat, strcmp, strlen, etc.

**25. Q: What is pointer?**

A: Variable storing address of another variable.

**26. Q: How to declare and initialize pointer?**

A: int \*ptr; ptr = &x;

**27. Q: Pointer to array?**

A: Pointer holding base address of array.

**28. Q: Pointer to function?**

A: Pointer holding address of function; used in callbacks.

**29. Q: Pointer to structure?**

A: Access members using -> operator.

**30. Q: Command-line arguments?**

A: Arguments passed to main(int argc, char \*argv[]).

**31. Q: What is enum in C?**

A: User-defined integer constants.

**32. Q: Dynamic memory allocation functions?**

A: malloc, calloc, realloc, free.

**33. Q: malloc vs calloc?**

A: malloc doesn’t initialize, calloc initializes with 0.

**34. Q: What are structures?**

A: Collection of variables of different data types.

**35. Q: Difference between struct and union?**

A: Struct stores all fields separately; union shares memory.

**36. Q: Array of structures?**

A: Multiple structures stored in an array.

**37. Q: File types in C?**

A: Text and Binary.

**38. Q: fopen and fclose usage?**

A: Used to open and close files.

**39. Q: Modes of file opening?**

A: "r", "w", "a", "rb", "wb", etc.

**40. Q: Reading and writing to a file?**

A: fscanf, fprintf, fgetc, fputc, fgets, fputs.

**41. Q: Appending data to file?**

A: Open with "a" or "a+" mode.

**42. Q: Difference between getc() and fgetc()?**

A: getc() is macro, fgetc() is function.

**43. Q: What is the use of rewind()?**

A: Sets file position to beginning.

**44. Q: What is feof()?**

A: Checks end of file.

**45. Q: Explain the flowchart.**

A: Graphical representation of algorithm using shapes.

**46. Q: What is an algorithm?**

A: Step-by-step solution for a problem.

**47. Q: Difference between algorithm and flowchart?**

A: Algorithm is textual, flowchart is visual.

**48. Q: Preprocessor directives?**

A: Begin with # (e.g., #include, #define).

**49. Q: What is typecasting?**

A: Explicit conversion of one data type into another.

**50. Q: What is dangling pointer?**

A: Pointer pointing to a deallocated **memory.**

**C Coding Questions**

**1. // Print Hello World**

#include<stdio.h>

int main() {

printf("Hello, World!\n");

return 0;

}

**2. // Add two numbers**

#include<stdio.h>

int main() {

int a, b;

scanf("%d%d", &a, &b);

printf("%d\n", a + b);

return 0;

}

**3. // Find maximum of two numbers using if-else**

#include<stdio.h>

int main() {

int a, b;

scanf("%d%d", &a, &b);

if(a > b)

printf("%d is greater\n", a);

else

printf("%d is greater\n", b);

return 0;

}

**4. // Calculate factorial using while loop**

#include<stdio.h>

int main() {

int n, fact = 1;

scanf("%d", &n);

while(n > 0) fact \*= n--;

printf("%d\n", fact);

return 0;

}

**5. // Sum of digits**

#include<stdio.h>

int main() {

int n, sum = 0;

scanf("%d", &n);

while(n) {

sum += n % 10;

n /= 10;

}

printf("%d\n", sum);

return 0;

}

**6. // Find reverse of a number**

#include<stdio.h>

int main() {

int n, rev = 0;

scanf("%d", &n);

while(n) {

rev = rev \* 10 + n % 10;

n /= 10;

}

printf("%d\n", rev);

return 0;

}

**7. // Check prime number**

#include<stdio.h>

int main() {

int n, i, flag = 1;

scanf("%d", &n);

for(i = 2; i <= n/2; i++) {

if(n % i == 0) {

flag = 0; break;

}

}

printf(flag ? "Prime\n" : "Not Prime\n");

return 0;

}

**8. // Use switch case to perform arithmetic operations**

#include<stdio.h>

int main() {

int a, b, choice;

scanf("%d%d%d", &a, &b, &choice);

switch(choice) {

case 1: printf("%d\n", a + b); break;

case 2: printf("%d\n", a - b); break;

case 3: printf("%d\n", a \* b); break;

case 4: printf("%d\n", a / b); break;

default: printf("Invalid\n");

}

return 0;

}

**9. // Fibonacci series using for loop**

#include<stdio.h>

int main() {

int n, a = 0, b = 1, c;

scanf("%d", &n);

for(int i = 0; i < n; i++) {

printf("%d ", a);

c = a + b; a = b; b = c;

}

return 0;

}

**10. // GCD of two numbers**

#include<stdio.h>

int main() {

int a, b;

scanf("%d%d", &a, &b);

while(a != b) {

if(a > b) a -= b;

else b -= a;

}

printf("GCD is %d\n", a);

return 0;

}

**11. // LCM of two numbers**

#include<stdio.h>

int main() {

int a, b, max;

scanf("%d%d", &a, &b);

max = (a > b) ? a : b;

while(1) {

if(max % a == 0 && max % b == 0) {

printf("LCM is %d\n", max);

break;

}

max++;

}

return 0;

}

**12. // Check palindrome number**

#include<stdio.h>

int main() {

int n, rev = 0, temp;

scanf("%d", &n);

temp = n;

while(n) {

rev = rev \* 10 + n % 10;

n /= 10;

}

printf(temp == rev ? "Palindrome\n" : "Not Palindrome\n");

return 0;

}

**13. // Count digits in a number**

#include<stdio.h>

int main() {

int n, count = 0;

scanf("%d", &n);

do {

count++;

n /= 10;

} while(n);

printf("%d\n", count);

return 0;

}

**14. // Armstrong number check**

#include<stdio.h>

#include<math.h>

int main() {

int n, sum = 0, temp, digit;

scanf("%d", &n);

temp = n;

while(n) {

digit = n % 10;

sum += digit \* digit \* digit;

n /= 10;

}

printf(temp == sum ? "Armstrong\n" : "Not Armstrong\n");

return 0;

}

**15. // Print pattern - triangle**

#include<stdio.h>

int main() {

int i, j, n;

scanf("%d", &n);

for(i = 1; i <= n; i++) {

for(j = 1; j <= i; j++) {

printf("\* ");

}

printf("\n");

}

return 0;

}

**16. // Convert Celsius to Fahrenheit**

#include<stdio.h>

int main() {

float celsius, fahrenheit;

scanf("%f", &celsius);

fahrenheit = (celsius \* 9 / 5) + 32;

printf("%.2f\n", fahrenheit);

return 0;

}

**17. // Check even or odd**

#include<stdio.h>

int main() {

int n;

scanf("%d", &n);

printf(n % 2 == 0 ? "Even\n" : "Odd\n");

return 0;

}

**18. // Sum of array elements**

#include<stdio.h>

int main() {

int n, arr[100], sum = 0;

scanf("%d", &n);

for(int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

sum += arr[i];

}

printf("%d\n", sum);

return 0;

}

**19. // Find largest element in array**

#include<stdio.h>

int main() {

int n, arr[100], max;

scanf("%d", &n);

for(int i = 0; i < n; i++)

scanf("%d", &arr[i]);

max = arr[0];

for(int i = 1; i < n; i++)

if(arr[i] > max) max = arr[i];

printf("%d\n", max);

return 0;

}

**20. // Count vowels in a string**

#include<stdio.h>

int main() {

char str[100];

int i, count = 0;

scanf("%s", str);

for(i = 0; str[i] != '\0'; i++) {

if(str[i]=='a'||str[i]=='e'||str[i]=='i'||str[i]=='o'||str[i]=='u'||

str[i]=='A'||str[i]=='E'||str[i]=='I'||str[i]=='O'||str[i]=='U')

count++;

}

printf("%d\n", count);

return 0;

}

**21. // Reverse a string**

#include<stdio.h>

#include<string.h>

int main() {

char str[100];

int i, len;

scanf("%s", str);

len = strlen(str);

for(i = len - 1; i >= 0; i--)

printf("%c", str[i]);

return 0;

}

**22. // Swap using pointers**

#include<stdio.h>

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x, y;

scanf("%d%d", &x, &y);

swap(&x, &y);

printf("%d %d\n", x, y);

return 0;

}

**23. // Recursive factorial**

#include<stdio.h>

int factorial(int n) {

if(n == 0) return 1;

return n \* factorial(n - 1);

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", factorial(n));

return 0;

}

**24. // Linear search**

#include<stdio.h>

int main() {

int n, key, arr[100];

scanf("%d", &n);

for(int i = 0; i < n; i++)

scanf("%d", &arr[i]);

scanf("%d", &key);

for(int i = 0; i < n; i++) {

if(arr[i] == key) {

printf("Found at index %d\n", i);

return 0;

}

}

printf("Not Found\n");

return 0;

}

**25. // Copy string without strcpy()**

#include<stdio.h>

int main() {

char src[100], dest[100];

int i;

scanf("%s", src);

for(i = 0; src[i] != '\0'; i++)

dest[i] = src[i];

dest[i] = '\0';

printf("%s\n", dest);

return 0;

}

**26. // Palindrome check**

#include<stdio.h>

#include<string.h>

int main() {

char str[100];

int i, len, flag = 1;

scanf("%s", str);

len = strlen(str);

for(i = 0; i < len/2; i++) {

if(str[i] != str[len-1-i]) {

flag = 0;

break;

}

}

printf(flag ? "Palindrome\n" : "Not Palindrome\n");

return 0;

}

**27. // Count digits in a number**

#include<stdio.h>

int main() {

int n, count = 0;

scanf("%d", &n);

while(n != 0) {

n /= 10;

count++;

}

printf("%d\n", count);

return 0;

}

**28. // Count frequency of elements in array**

#include<stdio.h>

int main() {

int n, arr[100], freq[100] = {0};

scanf("%d", &n);

for(int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

freq[arr[i]]++;

}

for(int i = 0; i < n; i++) {

if(freq[arr[i]] != -1) {

printf("%d occurs %d times\n", arr[i], freq[arr[i]]);

freq[arr[i]] = -1;

}

}

return 0;

}

**29. // Print Fibonacci series using recursion**

#include<stdio.h>

int fibonacci(int n) {

if(n <= 1) return n;

return fibonacci(n-1) + fibonacci(n-2);

}

int main() {

int n;

scanf("%d", &n);

for(int i = 0; i < n; i++)

printf("%d ", fibonacci(i));

return 0;

}

**30. // Count words in a string**

#include<stdio.h>

int main() {

char str[100];

int i, count = 1;

fgets(str, sizeof(str), stdin);

for(i = 0; str[i] != '\0'; i++) {

if(str[i] == ' ' || str[i] == '\n')

count++;

}

printf("%d\n", count);

return 0;

}

**31. // Reverse array elements**

#include<stdio.h>

int main() {

int arr[100], n;

scanf("%d", &n);

for(int i = 0; i < n; i++) scanf("%d", &arr[i]);

for(int i = n-1; i >= 0; i--) printf("%d ", arr[i]);

return 0;

}

**32. // Prime numbers in a range**

#include<stdio.h>

int main() {

int i, j, start, end, flag;

scanf("%d%d", &start, &end);

for(i = start; i <= end; i++) {

if(i < 2) continue;

flag = 1;

for(j = 2; j <= i/2; j++) {

if(i % j == 0) {

flag = 0; break;

}

}

if(flag) printf("%d ", i);

}

return 0;

}

33. // Power of a number

#include<stdio.h>

int main() {

int base, exp, res = 1;

scanf("%d%d", &base, &exp);

for(int i = 1; i <= exp; i++)

res \*= base;

printf("%d\n", res);

return 0;

}

**34. // Replace all spaces in a string with hyphens**

#include<stdio.h>

int main() {

char str[100];

fgets(str, sizeof(str), stdin);

for(int i = 0; str[i] != '\0'; i++) {

if(str[i] == ' ') str[i] = '-';

}

printf("%s", str);

return 0;

}

**35. // Count lowercase, uppercase, digits, and special characters**

#include<stdio.h>

int main() {

char str[100];

int i, lower=0, upper=0, digit=0, special=0;

fgets(str, sizeof(str), stdin);

for(i = 0; str[i] != '\0'; i++) {

if(str[i] >= 'a' && str[i] <= 'z') lower++;

else if(str[i] >= 'A' && str[i] <= 'Z') upper++;

else if(str[i] >= '0' && str[i] <= '9') digit++;

else if(str[i] != '\n') special++;

}

printf("Lowercase: %d\nUppercase: %d\nDigits: %d\nSpecial: %d\n", lower, upper, digit, special);

return 0;

}

**36. // Find LCM of two numbers**

#include<stdio.h>

int main() {

int a, b, max;

scanf("%d%d", &a, &b);

max = (a > b) ? a : b;

while(1) {

if(max % a == 0 && max % b == 0) {

printf("LCM = %d\n", max);

break;

}

max++;

}

return 0;

}

**37. // Find GCD of two numbers using Euclidean Algorithm**

#include<stdio.h>

int gcd(int a, int b) {

if(b == 0) return a;

return gcd(b, a % b);

}

int main() {

int a, b;

scanf("%d%d", &a, &b);

printf("GCD = %d\n", gcd(a, b));

return 0;

}

**38. // Convert decimal to binary**

#include<stdio.h>

int main() {

int n, binary[32], i = 0;

scanf("%d", &n);

while(n > 0) {

binary[i++] = n % 2;

n /= 2;

}

for(int j = i - 1; j >= 0; j--)

printf("%d", binary[j]);

printf("\n");

return 0;

}

**39. // Sum of all elements in a 2D array**

#include<stdio.h>

int main() {

int a[10][10], i, j, r, c, sum = 0;

scanf("%d%d", &r, &c);

for(i = 0; i < r; i++)

for(j = 0; j < c; j++) {

scanf("%d", &a[i][j]);

sum += a[i][j];

}

printf("Sum = %d\n", sum);

return 0;

}

**40. // Swap two numbers using pointers**

#include<stdio.h>

void swap(int \*x, int \*y) {

int temp = \*x;

\*x = \*y;

\*y = temp;

}

int main() {

int a, b;

scanf("%d%d", &a, &b);

swap(&a, &b);

printf("a = %d, b = %d\n", a, b);

return 0;

}

**41. // Find factorial using recursion**

#include<stdio.h>

int factorial(int n) {

if(n == 0) return 1;

return n \* factorial(n - 1);

}

int main() {

int n;

scanf("%d", &n);

printf("Factorial = %d\n", factorial(n));

return 0;

}

**42. // Count number of vowels and consonants**

#include<stdio.h>

int main() {

char str[100];

int vowels = 0, consonants = 0;

fgets(str, sizeof(str), stdin);

for(int i = 0; str[i] != '\0'; i++) {

char ch = str[i];

if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {

ch = (ch >= 'A' && ch <= 'Z') ? ch + 32 : ch;

if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') vowels++;

else consonants++;

}

}

printf("Vowels = %d, Consonants = %d\n", vowels, consonants);

return 0;

}

**43. // Copy one string to another**

#include<stdio.h>

int main() {

char str1[100], str2[100];

scanf("%s", str1);

int i = 0;

while((str2[i] = str1[i]) != '\0') i++;

printf("Copied string: %s\n", str2);

return 0;

}

**44. // Merge two arrays**

#include<stdio.h>

int main() {

int a[50], b[50], c[100], m, n, i;

scanf("%d", &m);

for(i = 0; i < m; i++) scanf("%d", &a[i]);

scanf("%d", &n);

for(i = 0; i < n; i++) scanf("%d", &b[i]);

for(i = 0; i < m; i++) c[i] = a[i];

for(i = 0; i < n; i++) c[m + i] = b[i];

printf("Merged array: ");

for(i = 0; i < m + n; i++) printf("%d ", c[i]);

return 0;

}

**45. // Matrix multiplication**

#include<stdio.h>

int main() {

int a[10][10], b[10][10], res[10][10] = {0};

int i, j, k, r1, c1, r2, c2;

scanf("%d%d", &r1, &c1);

for(i = 0; i < r1; i++)

for(j = 0; j < c1; j++)

scanf("%d", &a[i][j]);

scanf("%d%d", &r2, &c2);

for(i = 0; i < r2; i++)

for(j = 0; j < c2; j++)

scanf("%d", &b[i][j]);

if(c1 != r2) {

printf("Matrix multiplication not possible\n");

return 0;

}

for(i = 0; i < r1; i++)

for(j = 0; j < c2; j++)

for(k = 0; k < c1; k++)

res[i][j] += a[i][k] \* b[k][j];

printf("Resultant Matrix:\n");

for(i = 0; i < r1; i++) {

for(j = 0; j < c2; j++)

printf("%d ", res[i][j]);

printf("\n");

}

return 0;

}

**46. // Reverse an integer**

#include<stdio.h>

int main() {

int n, rev = 0;

scanf("%d", &n);

while(n != 0) {

rev = rev \* 10 + n % 10;

n /= 10;

}

printf("Reversed: %d\n", rev);

return 0;

}

**47. // Write a program using calloc**

#include<stdio.h>

#include<stdlib.h>

int main() {

int \*ptr, n, i;

scanf("%d", &n);

ptr = (int\*)calloc(n, sizeof(int));

for(i = 0; i < n; i++) scanf("%d", &ptr[i]);

for(i = 0; i < n; i++) printf("%d ", ptr[i]);

free(ptr);

return 0;

}

**48. // Append data to a file**

#include<stdio.h>

int main() {

FILE \*fp = fopen("file.txt", "a");

if(fp == NULL) {

printf("Error opening file\n");

return 1;

}

fprintf(fp, "\nAppended text");

fclose(fp);

return 0;

}

**49. // Find largest number using dynamic memory allocation**

#include<stdio.h>

#include<stdlib.h>

int main() {

int n, \*arr, max;

scanf("%d", &n);

arr = (int\*)malloc(n \* sizeof(int));

for(int i = 0; i < n; i++) scanf("%d", &arr[i]);

max = arr[0];

for(int i = 1; i < n; i++)

if(arr[i] > max) max = arr[i];

printf("Max = %d\n", max);

free(arr);

return 0;

}

**50. // Count number of words in a string**

#include<stdio.h>

#include<string.h>

int main() {

char str[200];

int i, count = 1;

fgets(str, sizeof(str), stdin);

for(i = 0; str[i] != '\0'; i++) {

if(str[i] == ' ' && str[i+1] != ' ' && str[i+1] != '\n') count++;

}

printf("Words = %d\n", count);

return 0;

}