

LIST OF PRACTICE PROGRAMS FOR "Programmimg Using C"

(copyright: Jitender Kumar Chhabra, NIT Kurukshetra)

Odd Sem 2019-20

Printf/scanf based:

- 1) Add, Subtract, Multiply two numbers.
- 2) Area of a circle, area & perimeter of triangle, rectangle, square, circle, etc
- 3) Simple interest of an amount

If-else based

- 1) Input 2 distinct numbers . Using if (no else), show which is bigger.
- 2) Input 2 numbers. Using if (not else)
 - (i) If first is bigger add both.
 - (ii) If 2nd is bigger subtract.
- 3) Do above question using if-else.
- 4) Input 2 numbers (non-distinct). Using if-else, find which is bigger. Print EQUAL if they are equal
- 5) Input a number and find sum of that number with a number one greater than it.
- 6) Check whether a number is even/odd
- 7) Convert time given in 24 hour format into 12 hour format.
- 8) Length is given in centimeters. Convert it into feet and inches.
- 9) Using if-else and && operator, find max out of 3 distinct numbers.
- 10) Using if-else and && operator, find max out of 3 non-distinct numbers.
- 11) Using if-else, find max out of 3 distinct numbers. Use of && etc is not allowed.
- 12) Without using && etc, find max as well as min out of 3 numbers. Same if should be used to find max as well as min. Aim to use minimum number of if statements.
- 13) Given numbers of days, Compute numbers of years, months & remaining days. Can you handle leap years as well?
- 14) Find roots of quadratic equation. Roots can be real/equal/complex.
- 15) Print grades of a student A+/A/B/C/D/E using if-else ladder
- 16) **Find max out of 4 distinct numbers without using && etc

**means difficult problem

Loop Based:

- 1) Using while, print Hello N times
- 2) Using while, find sum of series:

S= 1+2+3... N times	S= 11+12+13+... upto N
S=5+7+9+... N times	S=5+7+9+... upto N
S= 1+5+10+15 ... N times	S= 1+5+10+15 ... upto N
S=2+4+8+... N times	S=2+4+8+...upto N
S=1 -3 +5 -7 ... N times	S=1 -3 + 5 -7 ...upto N
- 3) Using for, implement above programs

- 4) Find Sum of n odd nos.
- 5) Sum of even numbers up to n (n being even).
- 6) Find factorial of a number. Number can be +ive/zero/-ive.
- 7) Using do-while, keep on reading numbers and find their sum till number entered is not -1.
- 8) Using do-while, implement above series sum
- 9) Find ASCII code of a symbol (say \$), of alphabet A, of alphabet a, of digit 0 etc.
- 10) Print following patterns, assuming each pattern is of N lines

*	*	*	A	1
**	**	***	ABC	121
***	***	*****	ABCDE	12321
****	****	...N	...N Lines	(after reaching to 9, next number will be 0)
...N linesN lines	lines	(assume N <=13)	123456789010987654321
		Also print above pattern in reverse order		12345678901210987654321
				... N lines

- 11) **Find whether a given number is prime or not.
- 12) Using if-else, implement calculator operations for supporting choices for four arithmetic operations +, -, *, / Program should continue repeating the calculations of the operations till Stop choice is not given.
- 13) Using loop, find sum of digits of a given number. Number will be read as integer.
- 14) **Using loops, find sum of digits of a given number, which is of float type. (Hint: integer part of the float number can be separated by assigning it to an int; fractional part can also be separated; find sum of digits of both parts separately)
- 15) Find sum of series $S = x - x^3/3! + x^5/5! - x^7/7! \dots$ upto $x^n/n!$. Do this program (i) by calculating numerator and denominator of each term separately for each iteration (ii) **implement efficiently, so that each numerator and denominator can be computed using previous numerator & denominator respectively.
- 16) **Find whether a given integer number is palindrome or not e.g. 12321 is a palindrome, 1221 is a palindrome, but 12331 is not.

Switch-case Based:

- 1) Given a day number, print day of the week assuming 1 means Monday... 7 meaning Sunday
- 2) Given a month number, print its Month name.
- 3) Read a date as collection of 3 integers (d,m,y). Using switch-case, find number of remaining days in that month. For example 15 Aug 2019 means, 16 days remaining. Implement it using (i) calculating number of remaining days in all 12 different cases (ii) think of doing these calculations min time i.e. clubbing together same calculation cases and calculating once only for the clubbed cases

- 4) Implement simple calculator using loops and switch. Previously calculated can get used in next repetition. Or user may reset it to 0, by pressing C(for clear).Calculator should keep on working till user does not decide to Exit.
- 5) Given a date, find day (Sun/Mon/Tues/...) of that date.
- 6) Given a number of max 4 digits, print it in words e.g. 2345 should be printed as two thousand three hundred forty five
- 7) **Given two dates (first date is more recent than second), find number of days in between these two. For ex. date 12,08,2019 and 10, 06, 2019 means 62 days (11 days of Aug+31 days of July+20 days of June)
- 8) **Add number of days to a given date and find new date.

Array Based

- 1) Find average of element in an array.
- 2) Find largest element in array, now find smallest element of the array.
- 3) Count number of -ive numbers, no of even numbers, no of odd numbers in an array.
- 4) Find sum of odd positioned elements of an array, also find sum of even positioned elements of the array (i) using two separate loops (ii) using same loop.
- 5) Input an array and find pair wise sum i.e. sum of first two values, next two value and so on. If n is odd, double the last value.
- 6) Input an array & find sum of first & last, 2nd & 2nd last term. If n is odd, print middle term alone.
- 7) **Change the above program such that for n being odd, middle value is added with itself. This is to be done without using any if statement. Use only one for loop and don't use any if inside it. Same loop should work correctly for n being even as well as odd.
- 8) Given an array, find index of first negative number. now find sum of all numbers from that index onwards (i) using break (ii) without using break, and using flag (iii) without break, without flag (hint: use loop from last backwards to find startindex)
- 9) **Compare whether two arrays are equal or not.
- 10) For a class of n students (roll number 1..N), store their physics, chem, math marks in 3 separate 1-dimensional arrays. Find the total marks scored by the student having highest marks in (i) physics (ii) chem (iii) math. Report if same student has highest marks in more than one subject. Assume that there are no duplicity in highest marks of a subject.
- 11) From an array, find the highest value, find compute how many values it appears, how many values are lesser than it, how many are greater than it.
- 12) Find second highest element of an array without sorting it.
- 13) Store the addition of two matrices into a new matrix, if they are addition-compatible. Display it row-wise on the screen.
- 14) Assume a class of N students having roll numbers 0..N-1. Their P, C, M marks are to be inputted. Use one 2-d array While reading the input, give suitable message for each input like "Enter Chemistry Marks of Roll No 5". Now compute and store in

the same 2-d array, the total marks of each student, average marks of each subject i.e. P, C, M. Also compute & store average total marks of the whole class.

- 15) Find and store transpose of a non-square matrix (i) into a new matrix, (ii) into itself without using any extra array/matrix. Display the result as row-wise matrix.

String based

- 1) Read a string using %s and find its length using loop.
- 2) Read a string and count how many times 'A' appears in it.
- 3) Using loop, copy a string into another string. Display the copied string using %s.
- 4) Using loop, reverse a string into (i) a new string (ii) into itself without using another string.
- 5) Concatenate two strings into a new string. Concatenated string must be stored and then displayed using %s
- 6) From a string, count number of upper case letters in it.
- 7) From a string, find number of vowels, number of consonants, number of digits and number of other symbols in it.
- 8) Concatenate second string on the right of first string. First string should get changed and no third string should be used.
- 9) Concatenate second string on the left of first string. First string should get changed and no third string should be used.

Function based

(No global variables to be used for these programs, unless specified. Use parameters. Don't take input in the function, except when the function is specifically being made for reading input. Don't print output in the same function, except when compulsory. Use return type)

- 1) Using function for factorial with suitable parameters & return type, compute nC_r .
- 2) Using function pow(a,b). compute a^b .
- 3) Write a function maxfn to find max out of 3 numbers. Call this function multiple times, to find max out of 9 numbers.
- 4) Using function, display pattern of above Q 10 (c), where the pattern should be printed using * or ? or \$ or any other symbol. symbol to be used will be input from the user and should be parameter to the function. N will also be another (obviously) parameter to the function.
- 5) Using function series_sum, find sum of series : $S = x - x^3/3! + x^5/5! - x^7/7! \dots$ upto $x^n/n!$. Implement it in two ways: (i) series_sum function internally calls fact and pow function (ii) series_sum function does not use pow & fact functions and each term is generated from previous term iteratively.
- 6) Write a function to check whether a number is prime or not, find first N prime numbers starting from 2 onwards
- 7) Write a function for find roots of quadratic equation. You are allowed to use two global variables to store the results. Roots could be real & unequal/equal /complex-conjugate. Output should clearly reflect which type of roots (real-unequal//equal/complex) are printed.

- 8) For 1-dimensional array, Write functions for each of the following: (i) average (ii) max, (iii) min (iv) second highest, assuming highest value is not duplicated. Array reading will be done in main.
- 9) Consider string as array of characters. Write your own functions for (i) strlen to compute length of the string (ii) strcpy to copy a string into another string (iii) strcmp to compare two strings (iv) strrev to reverse a string into itself (v) finding whether a given string is palindrome or not.
- 10) Write a function to find average of an array of float values. Write another function to find in
- 11) **Write a modular program to find sum of 2 matrices. Matrices could be non-square and orders should be read in main. Make separate functions for (i) reading a matrix (ii) adding two matrices (iii) displaying a matrix