1. Print hello world
2. Write a program to print 'Hello World' without using semicolon anywhere in the code.
3. Write a program to print “I Like [your_name]”.
4. Write a program to print “ I Like [your_name] and I am [your_age] years young”.
5. Write a program to add your CGPA my CPGA is [your_CGPA].
6. Redo the previous problems by storing your name, your age, your CGPA in variables.
7. Redo the previous program to accept the value from the user.
8. Write a program to accept two characters, check the equality and print. “equal” or “not equal “.
   Version : use getchar()
   Version : use scanf()
9. Write a program to accept two numbers and multiply them without using ‘*’symbol.
10. Write a program to print all letters of English [both uppercase and lowercase] repeatedly five times.
11. Write a program to accept a set of numbers and find the minimum, maximum, sum and the average.

Note: Input will be positive numbers and -1 will mark the termination

12. Write a program to accept a user input that would be either one or two, if the input was 1, program should print 2 if the input was 2, and program should print 1.
13. Write a program to accept a value in a long int variable and print them in binary, octal and hexa-decimal.

Note: Do not use format specifiers %o and %x

14. Write a program to accept input in following form

    1,1,1,1,2,2,1,2,2,2,2,1,1,1,2,2,2,2…..

    Find the total number of 1’s

    Find the total number of 2’s

    Find the total number of run for 1 and 2. A run is a consecutive flow; also find the maximum run for 1 and maximum run for 2
15. Write a program to find out the divisors of 1010101.

16. Write a program to accept a value and check whether it is 2’s power and print appropriate message.
17. Optimize the previous problem to reduce the number of instructions
18. Write a program to count a number of 1’s of a number, when in a binary form.
19. Write a program to multiply the given number by 7 without using ‘ *’ operator. Hint: use bitwise operator.
20. Write a program to accept two time value (HH:MM:SS) and display the elapsed time.
21. Swap two numbers using temporary variable
22. Swap two numbers without using temporary variable
23. Swap two numbers using bitwise operators
24. Write a program to calculate the electricity usage charges, accept initial reading and the end reading.

1 - 50	-> Rs.3.75
50 - 500	-> Rs.7.50
500 – 1000	-> Rs.10.00
1000 and above	-> Rs.30.00

FLOW CONTROL (NESTED & CASCADED IF ELSE, SWITCH)

25. Odd or even
26. Positive or negative or zero
27. Max between two numbers
28. Max among three numbers
29. If number divisible by 3 print Hi. If number divisible by 5 print Hello. If number divisible by 3 & 5 print HiHello
30. Print day of the week, for the given corresponding day number
31. Print name of the month, for the given corresponding month number
32. Leap year
33. Find the number of days remaining in a year for the given date and month of a year
34. Find the number of days passed in a year for the given date and month of a year

35. Write a program to accept the DOB of a person and calculate the age in NUMBER of days(DIFFERENCE BETWEEN TWO DATES)
36. Write a program to accept the DOB of a person and print the Day of Date.

NUMBER CRUNCHING PROBLEM & LOOPING

37. Write a C program which asks the user for a number between 1 to 9 and shows the number. If the user enters above the range show message as 'you entered number is not in range'.
38. Multiplication table
39. Generate 1 to 100
40. Generate 1 to 100 print odd no.
41. Print ASCII value using loops
42. Calculate sum of natural numbers.
43. Calculate sum of N using for Loop.
44. Print sum of odd numbers.

    1(1),4(1+3),9(1+3+5),16(1+3+5+7),25(1+3+5+7+9),…..
45. Calculate the power of a number.
46. Display odd numbers between two interval
47. Display even numbers between two intervals
48. Find factor, count number of factor and sum of factor
49. Perfect number
50. Prime number
51. Amicable numbers
52. Betrothed numbers
53. Display Prime numbers between two interval
54. HCF/GCD
55. LCM
56. Roman number to integer.
57. Split the number into digits and print it
58. Count number of digits of an integer.
59. Find sum of digits of an integer
60. Write a program to accept a number, and reverse the number
61. Check whether a number is palindrome or not

62. Write a function to find the next smallest palindrome number of the given number. for example: if given number is 12345, then the next smallest palindrome is 12421.
63. Adam number
64. Find Factorial of number
65. Strong number
66. Armstrong number
67. Convert decimal to binary; o/p is int;
68. Fibonacci series generation
69. Separate odd and even digits in a number
70. Replace digit(num, pos, digit)
71. Replace digit(num1,pos1, num2,pos2)

FUNCTION

72. Perfect number
73. Amicable number
74. Strong number
75. Prime number
76. Betroth number
77. Armstrong number

MINI PROJECTS

Calendar program

ARRAYS

78. Print left to right
79. Print right to left
80. Print sum of array
81. Replace every element with sum of its other elements
82. Replace every element with sum of its right side elements
83. Replace every element with sum of its left side elements
84. Average of the given values in an array
85. Find the repeated elements in an given array
86. Find common elements between two array

87. Count the number of zero’s and one’s in an array
88. Segregate 0’s to the left, 1’s in middle and 2’s to the right in an array
89. Reverse the array inPlace.
90. Reverse the half of the array
91. Reverse the partial array
92. Reverse the array as triplet
93. Rotate the array
94. Rotate the half of the array
95. Rotate the partial array
96. Rotate the array as triplet
97. Merging the two sorted array
98. Find the GCD of array
99. Given an array of integers, every element appears twice except for one. Find that single one.
100. Print and count all the numbers which are less than a given key element from a given array
.

101. Given an array of non-negative integers, you are initially positioned at the first index of the array.
Each element in the array represents your maximum jump length at that position.
Determine if you are able to reach the last index. For Example:
A= [2,3,1,1,4], return *true*
A= [3,2,1,0,4], return *false*

102. Input a int array. Square the elements in even position and cube the elements in odd position and add them as result. input: {1,2,3,4} output: 1^3+2^2+3^3+4^2

103. Find the sum of maximum and minimum number from a given input array Input:{19,17,12} Output:31

104. You are given a list of n-1 integers and these integers are in the range of 1 to n. There are no duplicates in list. One of the integers is missing in the list. Write an efficient code to find the missing integer.

```
    Example:
    I/P    [1, 2, 4, 6, 3, 7, 8]
   O/P    5
```

105. Given an unsorted array of size n. Array elements are in range from 1 to n. One number from set {1, 2 …n} is missing and one number occurs twice in array. Find these two numbers.

```
Examples:

arr[] = {3, 1, 3}
Output: 2, 3   // 2 is missing and 3 occurs twice

arr[] = {4, 3, 6, 2, 1, 1}
Output: 1, 5  // 5 is missing and 1 occurs twice
```

106. Given an array and a number k where k is smaller than size of array, we need to find the k'th smallest element in the given array. It is given that all array elements are distinct.

```
Examples:

Input: arr[] = {7, 10, 4, 3, 20, 15}
       k = 3
Output: 7

Input: arr[] = {7, 10, 4, 3, 20, 15}
       k = 4
Output: 10
```

107. Given two sorted arrays and a number x, find the pair whose sum is closest to x and the pair has an element from each array.

We are given two arrays ar1[0…m-1] and ar2[0..n-1] and a number x, we need to find the pair ar1[i] + ar2[j] such that absolute value of (ar1[i] + ar2[j] – x) is minimum.

```
Example:

Input:  ar1[] = {1, 4, 5, 7};
          ar2[] = {10, 20, 30, 40};
       x = 32
Output:  1 and 30

Input:  ar1[] = {1, 4, 5, 7};
          ar2[] = {10, 20, 30, 40};
       x = 50
Output:  7 and 40
```

108. Given three arrays sorted in non-decreasing order, print all common elements in these arrays.

```
Examples:

ar1[] = {1, 5, 10, 20, 40, 80}
ar2[] = {6, 7, 20, 80, 100}
ar3[] = {3, 4, 15, 20, 30, 70, 80, 120}
Output: 20, 80

ar1[] = {1, 5, 5}
ar2[] = {3, 4, 5, 5, 10}
ar3[] = {5, 5, 10, 20}
Output: 5, 5
```

109. Given an array which is sorted, but after sorting some elements are moved to either of the adjacent positions, i.e., arr[i] may be present at arr[i+1] or arr[i-1]. Write an efficient function to search an element in this array. Basically the element arr[i] can only be swapped with either arr[i+1] or arr[i-1].

For example consider the array {2, 3, 10, 4, 40}, 4 is moved to next position and 10 is moved to previous position.

```
Example:

Input: arr[] =  {10, 3, 40, 20, 50, 80, 70}, key = 40
Output: 2
Output is index of 40 in given array
```

```
Input: arr[] =  {10, 3, 40, 20, 50, 80, 70}, key = 90
Output: -1
```

-1 is returned to indicate element is not present

110. Given an array of 1s and 0s which has all 1s first followed by all 0s. Find the number of 0s. Count the number of zeroes in the given array.

```
Examples:

Input: arr[] = {1, 1, 1, 1, 0, 0}
Output: 2

Input: arr[] = {1, 0, 0, 0, 0}
Output: 4

Input: arr[] = {0, 0, 0}
Output: 3

Input: arr[] = {1, 1, 1, 1}
Output: 0
```

111. Given an array of integers, replace every element with the next greatest element (greatest element on the right side) in the array. Since there is no element next to the last element, replace it with -1. For example, if the array is {16, 17, 4, 3, 5, 2}, then it should be modified to {17, 5, 5, 5, 2, -1}.

112. Given a 2D array, print it in spiral form. See the following examples.

```
Input:
        1    2    3    4
        5    6    7    8
        9   10   11   12
       13   14   15   16
Output:
1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10



Input:
      1   2   3   4  5   6
      7   8   9  10  11  12
      13  14  15 16  17  18
Output:
1 2 3 4 5 6 12 18 17 16 15 14 13 7 8 9 10 11
```

113. Given an array A[] consisting 0s, 1s and 2s, write a function that sorts A[]. The functions should put all 0s first, then all 1s and all 2s in last.

```
Example
Input = {0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1};
Output = {0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2}
```

114. Given an array A[], write a function that segregates even and odd numbers. The functions should put all even numbers first, and then odd numbers.

```
Example
Input = {12, 34, 45, 9, 8, 90, 3}
Output = {12, 34, 8, 90, 45, 9, 3}
```

In the output, order of numbers can be changed, i.e., in the above example 34 can come before 12 and 3 can come before 9

115. Given an array arr[] of n integers, construct a Product Array prod[] (of same size) such that prod[i] is equal to the product of all the elements of arr[] except arr[i]. Solve it without division operator and in O(n).

```
Example:
arr[] = {10, 3, 5, 6, 2}
prod[] = {180, 600, 360, 300, 900}
```

116. You are given an array of 0s and 1s in random order. Segregate 0s on left side and 1s on right side of the array. Traverse array only once.

```
Input array   =  [0, 1, 0, 1, 0, 0, 1, 1, 1, 0]
Output array =  [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]
```

117. Write a ‘C’ program to remove all the duplicate entries and pile up all the unique entries to the top of the array, and replace the blank entries with -1

118. Write a ‘C’ program to swap all odd elements and all even elements of an integer array of size ‘n’

119. Given a sorted array and a number and element K. find K nearest elements to the number in sorted array.

120. Given an array of unsorted elements, find the minimum difference between any 2 elements in the array.

121. Given an input string, write a function that returns the compressed string for the input string in INPLACE. (no extra memory) (length of compressed string < = length of input string).

For example, if the input string is “aaabcdeeee”, then the function should return “a3b1c1d1e4?.

122. Given an array that has positive numbers and negative numbers and zero in it. You need to separate the negative numbers and positive numbers in such a way that negative numbers lies to left of zero and positive numbers to the right 5and the original order of elements should be maintained

123. Given an array of integers, {1,0,2,0,3,0,0,4,5,6,7,0,0,0}, you have to create a new array which will be like (1,2,3,4,5,6,7,0,0,0,0,0,0,0}, without using any other temporary array.

124. Given an array *arr[]* of size *n* where every element is in range from *0* to *n-1*. Rearrange the given array so that *arr[i]* becomes *arr[arr[i]]*. This should be done with *O(1)* extra space.

Examples:

Input: arr[] = {3, 2, 0, 1}
Output: arr[] = {1, 0, 3, 2}

Input: arr[] = {4, 0, 2, 1, 3}
Output: arr[] = {3, 4, 2, 0, 1}

Input: arr[] = {0, 1, 2, 3}
Output: arr[] = {0, 1, 2, 3}

If the extra space condition is removed, the question becomes very easy. The main part of the question is to do it without extra space.

125. Write a program to print all the LEADERS in the array. An element is leader if it is greater than all the elements to its right side. And the rightmost element is always a leader. For example int the array {16, 17, 4, 3, 5, 2}, leaders are 17, 5 and 2.

126. Majority Element**:** A majority element in an array A[] of size n is an element that appears more than n/2 times (and hence there is at most one such element).

Write a function which takes an array and emits the majority element (if it exists), otherwise prints NONE as follows:

```
     I/P : 3 3 4 2 4 4 2 4 4
     O/P : 4

    I/P : 3 3 4 2 4 4 2 4
     O/P : NONE
```

127. Given an array of integers , replace each element with the product of the remaining elements.

```
 Eg : Input - 1 2 3 4
    Output : 24 12 8 6
```

128. Replace each element of an array with its greatest next integer from the array in O(n).

129. Given array of ints. find ar[i],ar[j] such that j>i and ar[j]-ar[i] is maximum. Famous problem.

130. Given an array of integers which is initially increasing and then decreasing, find the maximum value in the array.

131. Given a sorted array which has been rotated, we have to find the point of rotation. I did it in O(n). Then he asked me to write a more optimized code.

132. You are given an array whose each element represents the height of the tower. The width of every tower is 1. It starts raining. How much water is collected between the towers?
Eg. [1,5,3,7,2] – then answer is 2 units between towers 5 and 7.
Looks easy, but if you don’t observe well, then you might end up with the wrong logic like I did at first. Also there are lots of possible corner cases. Luckily I could identify them all.

133. Given an array of integers, find an index such that if you split the array into two parts the absolute value of the difference between the sum of elements in both parts had to be minimum.

134. Given an array, find the longest increasing subsequence of size 3 with max product, all numbers are positive.

135. Given an array arrange it in the form a<b>c<d>e<f>g

136. Given an array of numbers arrange it in the form of a single number such that the concluded number is the maximum.
e.g. given : 99,8,76,45,66,9,7,33,5,42
O/P : 99987766654233

2D ARRAY

137. Write a ‘C’ program to rotate a m*n matrix by 90^0

Given a matrix of characters and a string, find whether the string can be obtained from the matrix. From each character in the matrix, we can move up/down/right/left. for example, if the matrix[3][4] is

```
o f a s

l l q w
```

z o w k

and the string is follow, then the function should return true.

138. Find the kth smallest element in a (MxN) matrix.

139. Given a 2 D array. The rows and columns are sorted.

Find the kth largest element from the 2-d array in most efficient way. Can it be done in-place

140. Given MxN matrix, which contains 1s and 0s only. Redraw the matrix so that, if any one position [i,j] contains 1, mark the entire row and column with 1. But make sure because of newly marked 1s, don’t do the same

```
i/p           o/p
0 0 0 1 0   1 1 1 1 1
0 0 0 0 0   1 1 0 1 1
0 0 0 0 0   1 1 0 1 1
1 0 0 0 1   1 1 1 1 1
0 1 0 1 0   1 1 1 1 1
```

141. Given a 2D array containing only 0/1′s and each row is in sorted order. Find the row which contains maximum number of 1s.

142. Print a matrix in spiral order (Code)

Soln: Solved it using recursion. Each recursive call was suppose to print boundary elements. On every recursive call, shifted the origin point and passed new size of matrix.

143. Given a N x N matrix find the no. of times a word is present in that matrix. constraints you can move in 3 directions from one cell 1. forward , 2. down 3. diagonal . Find all teh occurrence of all the word

forward means right (x+1,y)
down mean (x,y+1)
diagonal means (x+1,y+1)

it can be done with BFS. {search the no. of occurance of a given word example "sachin" in the whole NxN matrix}

w | s | r | t | g | g|

a | a | c | h | i | n |
k | c | h | u | j | j |
o | h | i | n | y | q |

in this sachin can be found out 4 times.

144. Given a 2D matrix, print all elements of the given matrix in diagonal order. For example, consider the following 5 X 4 input matrix.

```
  1    2    3    4
  5    6    7    8
  9   10   11   12
 13   14   15   16
 17   18   19   20
```

Diagonal printing of the above matrix is

```
  1
  5    2
  9    6    3
 13   10    7    4
 17   14   11    8
 18   15   12
 19   16
 20
```

SORTING

145. Selection sort
146. Bubble sort
147. Insertion sort
148. Heap sort
149. Quick sort
150. Merge sort

151. Radix sort

STRING (don’t use any library function)

152. String length
153. Check whether a character is vowel or consonant
154. Checker whether a character is an alphabet or not.
155. String copy
156. String concatenation
157. String reverse
158. String compare
159. Find a character in a given string(strchr)
160. Find the string in a given string(strstr)
161. Break the sentence into token(strtok)
162. Counting the vowels
163. Removing the vowels
164. Word reversal
165. Find and replace
166. Remove the repeated characters
167. Lapindrome
    Input: xyzyxz Output: Yes
    Input: acbkcba Output: Yes
168. Compress and decompress the string
169. Balanced string (string contain { } parenthesis)
170. Find the non-repeating character in a string
171. Encryption and decryption using Caesars algorithm
172. Segregate the vowels and constants - inplace
173. Palindrome
174. Convert the string into upper case and lower case
175. Capitalize each word
176. Count the characters, numbers, alphanumerical in a given string
177. Write a program that will exit When I press Q in keyboard
178. Count the number of characters in given string
179. Count the number of occurrences of substring in a string

180. Determine which word has the greatest number of repeated letters in a given string

181. Enter your name and return a string "print the title first and then comma and then first letter of initial name

182. Find the extension of a given string file. input: "hello.jpeg" output: "jpeg"

183. Find the number of words are of given length
Input1:{“aa”,”b”,”cc”,”ddd”} Input2:2 Output1:2

184. Given two numbers represented as strings, return multiplication of the numbers as a string.

Note: The numbers can be arbitrarily large and are non-negative.

185. Given a string containing only digits, restore it by returning all possible valid IP address combinations.

For Example:

Given'25525511135'

return ['255.255.11.135', '255.255.111.35'].

186. Isomorphic Strings.
For example
Given 'egg', 'add', return true.
Given 'foo', 'bar', return false.
Given 'paper', 'title', return true

187. Swap the even and odd characters in a given string.

188. Check the given day is a holiday or working day. Return true if it’s a working day.

Working day('Monday') → true

Working day ('Sunday') → false

Working day ('Tuesday') → true

189. Rotate an array of n elements to the right by k steps
For example, with n = 7 and k = 3, the array [1,2,3,4,5,6,7] is rotated to [5,6,7,1,2,3,4].

190. Given a string find the length of longest substring which has none of its character repeated?

for eg:
i/p string: abcabcbb
length of longest substring with no repeating charcters: 3 (abc)

191. Find first non-repeating character in String

ex: geeksforgeeks : f
geeksforgeeksFirst : o

192. Given a string say "I am a human being" the output should reverse all letters of each word but not the whole string as such.

Eg: O/p should be "I ma a namuh gnieb"

193. Given two strings find if they are anagrams or not.
eg. "tom marvolo riddle" and "i am lord voldemort".

(The example added the other constraint that the whitespaces donot matter.)

POINTERS

194. Dynamically allocate memory for given n numbers
195. Reallocate the memory
196. Swapping(call by reference)
197. String length
198. String copy
199. String concatenation
200. String reverse
201. String compare
202. Find a character in a given string(strchr)
203. Find the string in a given string(strstr)
204. Break the sentence into token(strtok)
205. Conversion of amount to words.

Input: 123 output: ONE HUNDRED AND TWENTY THREE

STRUCTURES

206. Date
207. Time

208. Create a structure to store student information.

MINI PROJECTS

209. Calendar program
210. Eight Queen Problems
211. Picture Puzzle – 1
212. Picture Puzzle – 2
213. MinesSweeper

PATTERN PROBLEMS

214. Input : 4

```
****
****
****
****
```

215. Input : 5

```
*****
****
***
**
*
```

216. Input : 5

```
*
**
***
```

```
****
*****
```

217. Input : 5

```
*****
****
 ***
**
*
```

218. Input : 4

```
1 1 1 1
2 2 2
3 3
4
```

219. Input :4

```
1
1 2
1 2 3
1 2 3 4
```

220. Input : 4

```
1   3    5    7
2   4    6    8
9   11  13  15
10  12 14  16
```

221. Input :3

```
1  2  3

4  5  6

7  8  9
```

222. Obtain a output as follows:

```
1

2     3

4     5     6

7     8     9     10

11    12    13    14    15

11    12    13    14    15

7     8     9     10

4     5     6

2     3

1
```

223. Write a C program to print below,when n=3 and s=3

```
3

4     4

5     5     5

4     4

3
```

224. Pattern Question:- For n=5

```
1
3*2
4*5*6
10*9*8*7
11*12*13*14*15
```

225. Pattern Question:-

```
1
```

22
333
4444
55555
4444
333
22
1

226. Pattern Question:-

To print the pattern like for n=3 the program should print

1 1 1 2

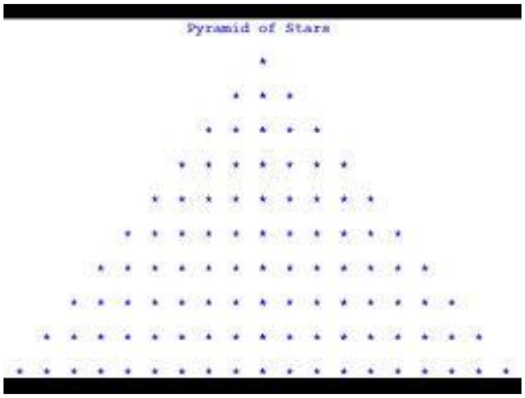
3 2 2 2

3 3 3 4

227. To print the trapezium pattern. for example , we have n=4 the output should be like

```
1*2*3*4*17*18*19*20
- -5*6*7*14*15*16
- - - -8*9*12*13
- - - - - -10*11
```

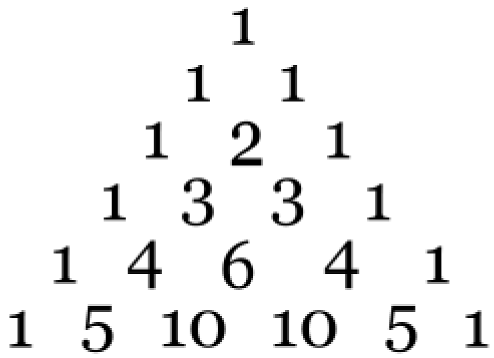
228. C program to print following pyramid pattern of stars



```
              Pyramid of Stars
                  *
                * * *
              * * * * *
            * * * * * * *
          * * * * * * * * *
        * * * * * * * * * * *
      * * * * * * * * * * * * *
    * * * * * * * * * * * * * * *
  * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
```

229.   Write a c program to print Pascal triangle.

In mathematics, Pascal's triangle is a triangular array of the binomial coefficients. In much of the Western world it is named after French mathematician Blaise Pascal, although other mathematicians studied it centuries before him in India, Iran, China, Germany, and Italy.



```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5  10  10   5   1
```