

1. Write a C program that takes a string as input and reverses each of the word without using built in function.

## Solution of 01:

```
1. String reverse wordByWord.c > ...
1  #include<stdio.h>
2  void stringReverseWordByWord(char str[], int start, int end)
3  {
4      char temp;
5      while(start<end)
6      {
7          temp=str[start];
8          str[start]=str[end];
9          str[end]=temp;
10         start++;
11         end--;
12     }
13 }
14 int main()
15 {
16     char str[1000];
17     int start = 0, end = 0;
18     printf("Enter a string to reverse word by word: ");
19     fgets(str, sizeof(str), stdin);
20     //printf("%d",str[end] );
21     while(str[end]){
22         for (end = start;str[end]&&str[end]!=' '; end++);
23         stringReverseWordByWord(str, start, end-1);
24         start=end+1;
25     }
26     puts(str);
27     return 0;
28 }
29
30
```

## Output of question 01:

```
(base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ gcc 1.\ String\ reverse\ wordByWord.c -o "1. String Reverse"
(base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./'1. String Reverse'
Enter a string to reverse word by word: abc xyz
cba
zyx
(base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ abc 123
Command 'abc' not found, but there are 17 similar ones.
(base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./'1. String Reverse'
Enter a string to reverse word by word: abc 123
cba
321
(base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./'1. String Reverse'
Enter a string to reverse word by word: abx @#1
xba
1#@
(base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ -
```

2. Write a C program that takes a string as input and finds the length of that string using recursive function.

## Solution of Question 2:

```
2. strlen using recursive.c > ...
1 // String Length using Recursion
2 // Created by: Jahidul Islam
3 // Date: 2022-09-09
4 #include<stdio.h>
5 #define SIZE 10000
6
7 int stringLenth(char str[], int start)
8 {
9     if(str[start]=='\0')
10         return start;
11     else
12         return stringLenth(str, start+1);
13 }
14 int main(){
15     char str[SIZE];
16     int start = 1;
17     printf("Enter a string to find the length: ");
18     fgets(str, sizeof(str), stdin);
19     //printf("The length of the string is: %d\n", stringLenth(str, start));
20     int size = stringLenth(str, start);
21     printf("The length of the string is: %d\n", size-1); // -1 for last index which is '\0'
22     return 0;
23 }
24
```

## Output of question 2:

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

● (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ gcc 2.\ srlten\ using\ recursive.c -o '2. strlen recursive'
● (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./2.\ strlen\ recursive
Enter a string to find the length: abc xyz
The length of the string is: 7
● (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./2.\ strlen\ recursive
Enter a string to find the length: abc 12
The length of the string is: 6
● (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./2.\ strlen\ recursive
Enter a string to find the length: abx@#
The length of the string is: 5
○ (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$
```

3. Suppose you have some words and you want to right justify them, that is, align them to the right. Create a program that reads some words and print it all right justified, in the same order as they appear in the input.

## Solution of question 03:

```
c 3. right alignment.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  int main(){
5      printf("Enter total number of Input ( 1 to 50): ");
6      int n;
7      scanf("%d", &n);
8      if(n<1 || n>50){
9          printf("Invalid Input");
10         exit(0);
11     }
12     printf("Enter %d String: ", n);
13     //declare 50 char arrays
14     char str[50][100];
15     for (int i = 0 ; i < n; i++)
16     {
17         scanf("%s", str[i]);
18     }
19     //find the longest string
20     int max = 0;
21     for (int i = 0; i < n; i++)
22     {
23         if(strlen(str[i])>max)
24             max = strlen(str[i]);
25     }
26     //printf("%d", max);
27     printf("The output file is: \n\n");
28     for (int i = 0; i < n; i++)
29     {
30         printf("%*s", max, str[i]);
31         printf("\n");
32     }
33     return 0;
34 }
```

## Output of question 03:

```
j4hidu1z4id@linux: ~/Desktop/C lan/C99/CSE103
-o '3. rightAlignment'
(base) j4hidu1z4id@linux:~/Desktop/C lan/C99/CSE103$ ls
'1. String Reverse'          '3. right alignment.c'
'1. String reverse wordByWord.c'  a.out
'2. strlen using recursive.c'    'CT3_Assignment_CSE 103_221_Summer 2022.pdf'
'2. strlen recursive'          CT3.odt
'3. rightAlignment'            tem.c
(base) j4hidu1z4id@linux:~/Desktop/C lan/C99/CSE103$ ./3.\ rightAlignment
Enter total number of Input ( 1 to 50): 3
Enter 3 String: Bob
Tommy
Jim
The output file is:

  Bob
Tommy
  Jim
(base) j4hidu1z4id@linux:~/Desktop/C lan/C99/CSE103$ ./3.\ rightAlignment
Enter total number of Input ( 1 to 50): 4
Enter 4 String: LONGEST
A
LONGER
SHORT
The output file is:

LONGEST
  A
  LONGER
  SHORT
(base) j4hidu1z4id@linux:~/Desktop/C lan/C99/CSE103$
```

4. Cyber Security is a key issue to protect our daily documents and applications stored and submitted in various platforms. Having a robust encryption system to our generated password is very essential in this perspective. Your task is to create a nice and smooth encrypted password generator. Follow the instructions carefully to build the password generator. [5 Marks]

## Solution of question 04:

```
C 5. passGen.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  int digitCount(int n)
5  {
6      if (n == 0)
7      {
8          return 0;
9      }
10     return 1 + digitCount(n / 10);
11 }
12 int Input_Number()
13 {
14     int n;
15     printf("Enter a 5 digits number: ");
16     scanf("%d", &n);
17     if (digitCount(n) != 5)
18     {
19         printf("Wrong Input\n");
20         exit(0);
21     }
22     return n;
23 }
24 // it has one more page
25
```

Green University of Bangladesh  
Department of Computer Science and Engineering  
CT3 (Assignment), Summer 2022

Course Code: CSE 103

Course Title: Structured Programming

```
C 5. passGen.c > ...
29 //page 2
30 void password_generator(int a)
31 {
32     switch (a)
33     {
34         case 1:
35             printf("#");
36             break;
37         case 2:
38             printf("a");
39             break;
40         case 3:
41             printf("t");
42             break;
43         case 4:
44             printf("j");
45             break;
46         case 5:
47             printf("9");
48             break;
49         case 6:
50             printf("E");
51             break;
52         case 7:
53             printf("@");
54             break;
55         case 8:
56             printf("2");
57             break;
58         case 9:
59             printf("F");
60             break;
61         case 0:
62             printf("?");
63             break;
64         default:
65             printf("\n");
66             break;
67     }
68 }
69 // it has one more page
70 int main()
```



```
68 }  
69 // it has one more page  
70 int main()  
71 {  
72     int recived, temp;  
73     int i, j, k;  
74     int str[5];  
75  
76     recived = Input_Number();  
77  
78     for (i = 0; i <= 5; i++)  
79     {  
80         temp = recived;  
81         recived = temp % 10;  
82         for (j = 5; j > i; j--)  
83         {  
84             str[j] = recived;  
85         }  
86         temp = temp / 10;  
87         recived = temp;  
88     }  
89     // reverse the array and pass it to the password generator  
90     for (k = 5; k >= 0; k--)  
91     {  
92         password_generator(str[k]);  
93     }  
94     printf("\n\n");  
95 }  
96  
97 //end of the code
```

**Output of question 04:**



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

• (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ gcc 5.\ passGen.c -o '5. password Generator'
• (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./5.\ password\ Generator
Enter a 5 digits number: 12345
#atj9

• (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./5.\ password\ Generator
Enter a 5 digits number: 287
Wrong Input
• (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./5.\ password\ Generator
Enter a 5 digits number: 5249629665
Wrong Input
• (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./5.\ password\ Generator
Enter a 5 digits number: 91778
F#@@2

○ (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$
```

5. In programming terms a recursive function can be defined as a routine that calls itself directly or indirectly. Using recursive algorithm, certain problems can be solved quite easily. Write a program using recursive function that can convert a Decimal number to its equivalent Binary number.

## Solution of question 05:

```
C 4. decimal to binary using recursion.c > ...
1 |
2 | #include<stdio.h>
3 | int decToBinary(int n)
4 | {
5 |     if (n == 0)
6 |         return 0;
7 |     else
8 |         return (n % 2 + 10 * decToBinary(n / 2)); //recursive call
9 | }
10 | int main()
11 | {
12 |     int n;
13 |     printf("Enter a decimal number: ");
14 |     scanf("%d", &n);
15 |     // print and calling the recursive function
16 |     printf("%d\n", decToBinary(n));
17 |     return 0;
18 | }
19 |
```

## Output of question 05:

```
● (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ gcc 4.\ decimal\ to\ binary\ using\ recursion.c -o '4. decToBin recF'
● (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./4.\ decToBin\ recF
Enter a decimal number: 3
11
● (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./4.\ decToBin\ recF
Enter a decimal number: 5
101
● (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./4.\ decToBin\ recF
Enter a decimal number: 11
1011
o (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$
```

6. Write a program that will correctly decode a set of characters into a valid message. Your program should read a given file of a simple coded set of

characters and print the exact message that the characters contain. The code key for this simple coding is a one for one character substitution based upon a single arithmetic manipulation of the printable portion of the ASCII character set.

## Solution of question 06:

```
c 6. read file and decode.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      freopen("sampleInput5.txt", "r", stdin);
5      freopen("sampleOutput5.txt", "w", stdout);
6      char ch;
7      int t;
8      while(scanf("%c", &ch) == 1) {
9          if(ch != '\n')
10             printf("%c", ch-7);
11         else printf("\n");
12     }
13     return 0;
14 }
```

## output 06:

1.

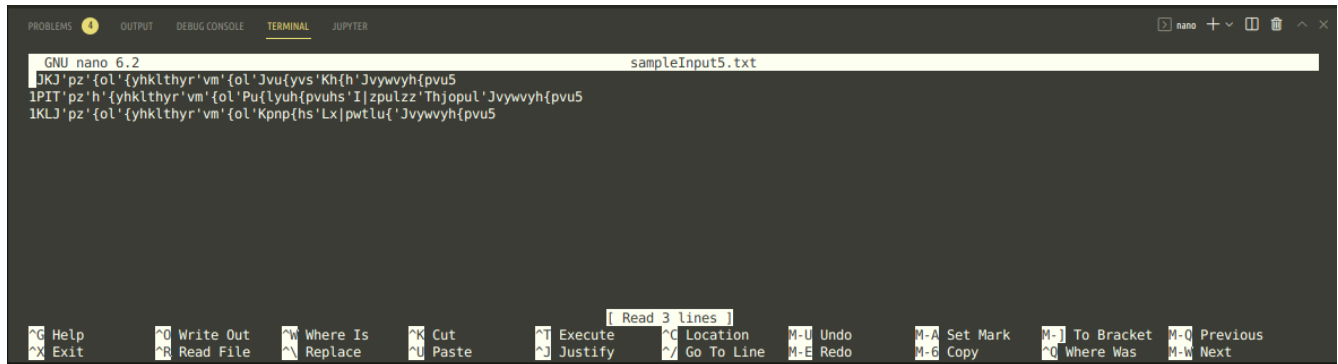
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
(base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ gcc 6.\ read\ file\ and\ decode.c
(base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ./a.out
(base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ nano sampleInput5.txt
o (base) j4hidulz4id@linux:~/Desktop/C lan/C99/CSE103$ ^[[2-
```

Green University of Bangladesh  
Department of Computer Science and Engineering  
CT3 (Assignment), Summer 2022

Course Code: CSE 103

Course Title: Structured Programming

2.

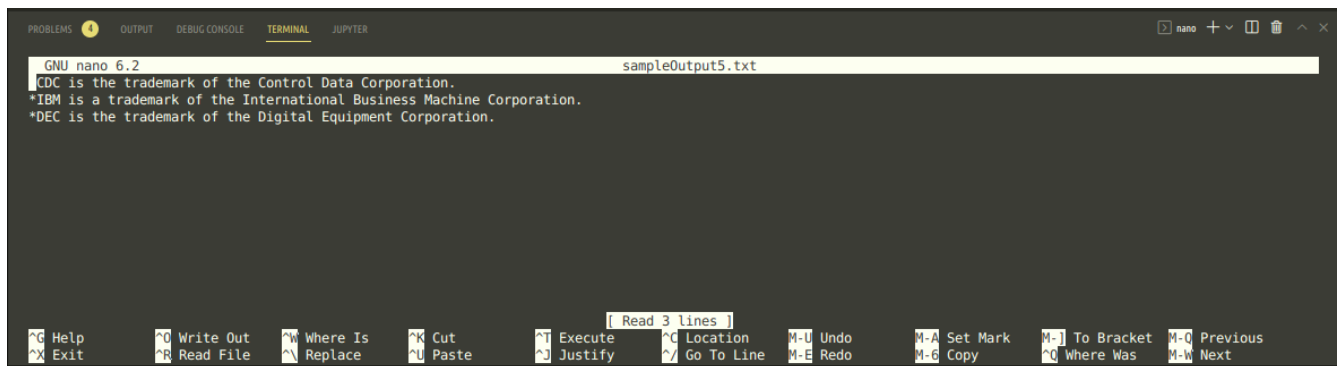


```
GNU nano 6.2 sampleInput5.txt
}KJ'pz'{ol'{yhklthyr'vm'{ol'Jvu{yvs'Kh{h'Jvyvvyh{pvu5
1PIT'pz'h'{yhklthyr'vm'{ol'Pu{lyuh{pvuhs'I}zpuLzz'Thjopul'Jvyvvyh{pvu5
1KLJ'pz'{ol'{yhklthyr'vm'{ol'Kpnp{hs'Lx|pwtlu{'Jvyvvyh{pvu5
```

Read 3 lines

^G Help   ^O Write Out   ^W Where Is   ^K Cut   ^T Execute   ^C Location   M-U Undo   M-A Set Mark   M-] To Bracket   M-Q Previous  
^X Exit   ^R Read File   ^\_ Replace   ^U Paste   ^J Justify   ^\_/ Go To Line   M-E Redo   M-G Copy   ^Q Where Was   M-W Next

3.



```
GNU nano 6.2 sampleOutput5.txt
*CDC is the trademark of the Control Data Corporation.
*IBM is a trademark of the International Business Machine Corporation.
*DEC is the trademark of the Digital Equipment Corporation.
```

Read 3 lines

^G Help   ^O Write Out   ^W Where Is   ^K Cut   ^T Execute   ^C Location   M-U Undo   M-A Set Mark   M-] To Bracket   M-Q Previous  
^X Exit   ^R Read File   ^\_ Replace   ^U Paste   ^J Justify   ^\_/ Go To Line   M-E Redo   M-G Copy   ^Q Where Was   M-W Next