# Solutions

## Question No. 01

Write a C program that will take a string  as input from you and change the cases of the characters.If the letter is capital, change it into small and if the letter is small change it into capital letter. Look at the test case and explanation for more clarification.

**Sample Input:**

```
hElLoWorld
```

**Sample Output:**

```
HeLlOwORLD
```

**Explanation:**

Here the first letter is small in the given input that's why we converted it into capital and the second letter is capital that's why we converted it into small.

## Answer No. 01

```c
#include <stdio.h>
#include <string.h>

int main(){
  char str[100];
  scanf("%s",&str);

  for(int i=0; i<strlen(str); i++){
    if(str[i] >= 'a' && str[i] <= 'z'){
      printf("%c",str[i]-32);
    }
    if(str[i] >= 'A' && str[i] <= 'Z'){
      printf("%c",str[i]+32);
    }
  }

  return 0;
}
```

## Question No. 02

```
int i=0;
while(i<10)
{
      printf(“I am inside the loop”);
}
```

What is wrong with this block of C code? Mark the errors and fix them.

## Answer No. 02

In this code,
Loop will never stop, because the condition we give **i<10**, but we can not increase the value of **i** , that’s why **i** will always be 0 and the loop will always run.
To fix this error we need to increase the value of i, by this way when **i** will be 10, the loop will stop.

After fixing the error the code is

```
int i=0;
  while(i<10)
  {
    printf("I am inside the loop");
    i++;
  }
```

## Question No. 03

Your faculty was born in the year 1997 and he loves the digits of this number. So he is asking you to write a code where you will be given a string of digits as input and you need to find whether all the digits 1,9, and 7 is present in that string or not. Use function to solve this problem.
[Hint: Write a function that will find a particular character in a string and call that function thrice to find each of 1,9, and 7 in the given string.]

**Sample Input 1:**

```
36892502945417835
```

**Sample Output 1:**

Yes

**Explanation 1:**

All 1, 9, and 7 are present in the number

**Sample Input:**

3689250294547835

**Sample Output:**

No

**Explanation:**

The digit 1 is missing from the input string

## Answer No. 03

```
#include <stdio.h>
#include <string.h>

int checkDigit(char digitstr[], char digit){
  for(int i=0; i<strlen(digitstr); i++){
    if(digitstr[i] == digit){
      return 1;
    }
  }
  return 0;
}

int main(){
  char str[100];
  scanf("%s", &str);

  if(checkDigit(str,'1')==1 && checkDigit(str,'7')==1 &&
checkDigit(str, '9')==1){
    printf("Yes");
  } else{
    printf("No");
  }

  return 0;
}
```

## Question No. 04

You will be given two integers **n** and **k**. First you need to print all the even numbers from 1 to n serially and then followed by those even numbers print all the odd numbers in the same range. Finally, find the k-th number in the sequence that you just printed.

**Sample Input:**

```
13 9
```

**Sample Output:**

```
2 4 6 8 10 12 1 3 5 7 9 11 13
The 9th element in this sequence is 5.
```

**Explanation:**

After printing the even numbers and then all the odd numbers if we look at the sequence the 9th element in the sequence is 5.
The 1st one is 2, the 2nd one is 4 and so on.

## Answer No. 04

```c
#include <stdio.h>

int main(){
  int n, k;
  scanf("%d %d", &n, &k);
  int arr[n], index=0;

  for(int i=1; i<=n; i++){
    if(i%2==0){
      arr[index] = i;
      index++;
    }
  }

  for(int i=1; i<=n; i++){
    if(i%2!=0){
      arr[index] = i;
      index++;
    }
  }

  for(int i=0; i<n; i++){
    printf("%d ",arr[i]);
  }
  printf("\nThe %dth element in this sequence is %d.",k, arr[k-1]);

  return 0;
}
```

## Question No. 05

Assume that you have a function named add_three_nums() which will take three numbers as arguments. The description of the function is also given here. Just copy it and paste it in your code.

```
int add_three_nums(int a, int b, int c)
{
      return a+b+c;
}
```

Now, write a C program where you will take only two inputs and add those two numbers using this function. You can add them directly using the + operator. You must use this function to add them, but the challenge is you are only given two inputs.

## Answer No. 05

```
#include <stdio.h>

int add_three_nums(int a, int b, int c)
{
  return a+b+c;
}

int main(){
  int x, y;
  scanf("%d %d",&x,&y);

  printf("%d",add_three_nums(x,y,0));
```

```c
  return 0;
}
```

## Question No. 06

Write a function that will take an integer as an argument and return the factorial of the integer. Then write another function that will take two integers as arguments and then return the ratio of the factorials of the first argument with the second one. You need to make two calls to the first function inside the second function to find the factorials of the numbers.

## Answer No. 06

```c
long long int factorial(long long int x){
  long long int fact = 1;

  for(long long int i=1; i<=x; i++){
    fact = fact * i;
  }

  return fact;
}

double calculateRatio(long long int  a, long long int b){
  double f1, f2;

  f1 = (double)factorial(a);
  f2 = (double)factorial(b);

  return (f1/f2);
}
```

## Question No. 07

Write a C function that will take an array of integers as an argument and return the median value of the array.
For hints see this

Note - The given array will not be sorted, hence first you need to sort the given array.And calculate the median for both odd and even length arrays.

## Answer No. 07

```
#include<stdio.h>
int main()
{
   int n;
   scanf("%d",&n);
   int arr[1000];
   for(int i=0; i<n; i++)
      scanf("%d",&arr[i]);
   int i,j,flag=0;
   for(i=0; i<n-1; i++)
   {
      flag=0;
      for(j=0; j<n-i-1; j++)
      {
         if(arr[j]>arr[j+1])
         {
            int temp=arr[j];
            arr[j]=arr[j+1];
            arr[j+1]=temp;
            flag=1;
         }
      }
      if(flag==0)
         break;
   }
   if(n%2!=0)
   {
      int median=arr[n/2];
      printf("%d\n",median);
   }
   else
   {
      float median=((arr[n/2]*1.0)+arr[(n/2)-1])/2;
      printf("%.3f\n",median);
   }

   return 0;
}
```

## Question No. 08

Write a C program that will take a string and another number as input. Shift all the characters of the string by that numerical value. The string will consist of only small letters. Look at the explanation for more clarification.

**Sample Input:**

```
smiley
5
```

**Sample Output:**

```
xrnqjd
```

**Explanation:**

Here, the first character was s and it has been shifted by 5 units, that is, s+5 = x, thus we got the character ‘x’ in the output. Similarly all the character by shifting the characters by 5 units. Take a closer look at the last character. The character was ‘y’ but after shifting it by 5 units we got ‘d’. That is because we traversed the letters in a circular fashion. After ‘y’ comes ‘z’ then ‘a’ then ‘b’ then ‘c’ and then finally ‘d’. Thus, ‘y’+5 will be ‘d’ in our program. Be careful about this case.

## Answer No. 08

```
#include<stdio.h>
int main()
{
   char s[100];
   scanf("%s",s);
   int k;
   scanf("%d",&k);
   for(int i=0; i<strlen(s); i++)
   {
      int asciiValue=s[i];
      if((asciiValue+k)<=122)
         printf("%c",(asciiValue+k));
      else
      {
         int rem=(((asciiValue+k)-122)%26);
         if(rem==0)
            printf("z");
         else
            printf("%c",rem+96);
      }
   }
   return 0;
}
```

## Question No. 09

Write a C program that will take a 3x3 matrix as input and then print the transpose of that matrix. For hints see this.

## Answer No. 09

```
#include<stdio.h>
int main()
{
   int A[3][3],transpose[3][3],i,j,row=3,col=3;


   //Scanning A matrix
   for(i=0; i<row; i++)
   {
      for(j=0; j<col; j++)
      {
         printf("A[%d][%d] =",i,j);
         scanf("%d",&A[i][j]);
      }
      printf("\n");
   }


   //Transpose Matrix
   for(i=0; i<row; i++)
```

```
    {
        for(j=0; j<col; j++)
        {
            transpose[j][i]=A[i][j];
        }
    }

    //Printing transpose matrix
    printf("Transpose - \n");
    for(i=0; i<col; i++)
    {
        printf("\t");
        for(j=0; j<row; j++)
        {
            printf("%d ",transpose[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

## Question No. 10

Write a function that will take the mark of a student as an argument and return the grade as a character. Look at the table below to understand the grading policy.

| **Mark Range** | **Grade** |
|---|---|
| 80-100 | A |
| 60-79 | B |
| 40-59 | C |
| 0-39 | F |

## Answer No.  10

```
#include<stdio.h>
char getMyGrade(int marks)
{
    if(marks>=0 && marks<=39)
        return 'F';
    else if(marks>=40 && marks<=59)
        return 'C';
    else if(marks>=60 && marks<=79)
        return 'B';
    else
        return 'A';
}
int main()
{
    int marks_;
    scanf("%d",&marks_);
    printf("Grade = %c \n",getMyGrade(marks_));

    return 0;
}
```