

Question-1:

Problem Statement:

Write a C-program to input your Name (up to 20 characters), Roll number (exactly 8-digit natural number), Height (up to 5 digits including decimal numbers), Address (up to 50 characters) and output the same.

Sample Input:

Enter name: Rahul
Enter roll number: 140101001
Enter height: 160.5
Enter address: MG Road, New Delhi

Sample Output:

Student information:
Name: Rahul
Roll Number: 140101001
Height: 160.5
Address: MG Road, New Delhi

Question-2:

Problem Statement:

The given program (shown below) is used to print the value of “phi” as follows
“3.14285707473754882812”.

Modify the below code to print the result that has exactly 20 digits after the decimal point.

```
void main ()
{
    int f
    f = 22 / 7
    printf ('value of f : %d \n', f)
}
```

Expected Output:

Value of f: 3.14285707473754882812

Question-3:

Problem Statement:

Write a C-program to convert 24 hour time format (0 to 23) to 12 hour time format (0 to 11). In addition to this, calculate the percentage of time that has passed for the day. Print the error message “Invalid input time”, if input is not within the range.

Sample Input & Output – 1:

Enter the time: 13

1 pm, 54.16667% of time passed for the day

Sample Input & Output – 2:

Enter the time: 3

3 am, 12.5% of time passed for the day.

Question-4:

Problem Statement:

Given C-program is used to print the ASCII value of a character entered by user.

Identify and correct the (syntactical as well as logical) errors in the program to meet its objective.

```
main()
{
    char c
    printf('Enter a character: ')
    scanf('%d', c)    /* Takes a character from user */
    printf("\n ASCII value of %d = %c \n", c, c)
}
```

Sample Input & Output:

Enter a character: A

ASCII value of A = 65

Question-5:

Problem Statement:

Write a C-program to print the following patterns manually using printf() based on user choice.

If user enters 1, print this pattern:

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

If user enters 2, print this pattern:

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

For other user inputs, print “Invalid Choice”.

Sample Input & Output:

Enter the choice: 2

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

Question-6:**Problem Statement:**

Given program is used to check whether the input alphabet is a vowel or not?.Both upper-case and lower-case alphabets have to be checked.

Identify and correct the (syntactical as well as logical) errors in the program to meet its objective.

```
int main()
{
    int ch

    printf(Enter a character\n)

    if (ch = "a" || ch = "A" || ch = "e" || ch = "E" || ch = "i" || ch = "I" || ch = "o" || ch = "O" || ch = "u" ||
ch = "U")
        printf("%d is a vowel.\n", ch)
    elseif
        printf("%d is not a vowel.\n", ch)

}
```

Sample Input & Output - 1:

```
Enter a character
u
u is a vowel.
```

Sample Input & Output - 2:

```
Enter a character
w
w is not a vowel.
```

Question-7:**Problem Statement:**

Write a C-program to search a given number in a sequence of numbers. Receive exactly ten integer numbers (upto 4 digit length) from user. Print the information about how many times the given number is present in the input. If user enters less than or greater than ten numbers, print the error message "Enter exactly ten numbers". You can use array or any number of variables as you wish.

Sample Input & Output - 1:

```
Enter ten numbers: 100 -700 20 30 10 -10 40 80 90 100
Enter the number to search: 100
100 is present in the given numbers for two times.
```

Sample Input & Output - 2:

```
Enter ten numbers: 100 -700 20 30 10 -10 40 80 90 100
Enter the number to search: 99
99 is not present in the given numbers.
```

Question-8:**Problem Statement:**

Given program is used to print the following output:

5 abc 3.140000 s

Identify and correct the (syntactical as well as logical) errors in the program to meet its objective.

```
main()
{
    int dec = 5
    char st[ ] = "abc"
    char ch = "s"
    int pi = 3.14

    scanf("%d %c %d %s \n, dec, str, pi, ch);
}
```

Question-9:

Problem Statement:

Write a C-program to find the greatest among four given distinct input values. The input contains alphabet (lower case and upper case letter). If user enters other than above mentioned possibilities, print the error message "Invalid Input". If any two inputs are same, then print the output "Some inputs are same".

Sample Input & Output - 1:

Enter four alphabets: A a Z D
Greatest among the input values: a

Sample Input & Output - 2:

Enter four alphabets: d d B C
Some inputs are same

Hint:

ASCII value of A is 65. ASCII value of Z is 90.
ASCII value of a is 97. ASCII value of z is 122.

Question-10:

Problem Statement:

Given program is used to print the following output:
The color: blue
First number: 12345
Second number: 0025
Third number: 1234
Float number: 3.14
Hexadecimal: ff
Octal: 377
Unsigned value: 4294967146
Signed value: -150
Just print the percentage sign %

Identify and correct the (syntactical as well as logical) errors in the program to meet its objective.
Fix only the errors and *warnings you may omit*.

```

main()
{
    scanf("The color: %m\n", "blue");
    scanf("First number: %c\n", 12345);
    scanf("Second number: %04d\n", 25);
    scanf("Third number: %i\n", 1234);
    scanf("Float number: %f\n", 3.14159);
    scanf("Hexadecimal: %d\n", 255);
    scanf("Octal: %x\n", 255);
    scanf("Unsigned value: %s\n", -150);
    scanf("Signed value: %u\n", -150);
    scanf("Just print the percentage sign %%\n", 10);
}

```

Question-11:

Problem Statement:

Write a C-program to prompt the user to enter any natural number in the range between 0 to 9,999. If the entered input number 'x' is out of this range, then print the following message: "x is out of range 0 to 9,999". Otherwise print the digits of the input number 'x' in word in the order they appear in the number.

Sample Input & Output – 1:

Enter the input number: 233
Two Three Three

Sample Input & Output – 2:

Enter the input number: 10000
10000 is out of range 0 to 9,999

Question-12:

Problem Statement:

The given program segment (shown below) is used to check whether the input integer number is even or odd. Identify and correct the (syntactical as well as logical) errors in the program to meet its objective.

```

main()
{
    printf ("Enter a number you want to check: ");
    scanf ("%d", num);

    if ((num / 2) = 0)
        printf ("%d is even.\n", num);
    else
        printf ("%d is odd.\n", num);
}

```

Followings are the expected sample outputs your program after corrections are made.

Sample Input & Output – 1:

Enter a number you want to check: 4
4 is even

Sample Input & Output – 2:

Enter a number you want to check: 5
5 is odd

Question-13:**Problem Statement:**

Write a C-program to convert date in the format of **DDMMYYYY** to **DD / Month / YYYY** and output the same. If the input is not adhering to the above format, print the error message “Invalid input”. The input date should be stored in an unsigned integer variable.

Sample Input & Output – 1:

Enter the date: 01012015
01 / January / 2015

Sample Input & Output – 2:

Enter the date: 0101201
Invalid input

Question-14:**Problem Statement:**

Given program is used to check whether the user entered year (upto 4 digits) is leap or not. Identify and correct the (syntactical as well as logical) errors in the program to meet its objective.

```
int main()
{
    printf ("Enter a year to check if it is a leap year: ");
    scanf ("%f", year);

    if ( year/400 = 0)
        printf("\n %d is a leap year.\n", year);
    else ( year/100 = 0)
        printf("/f is not a leap year.\n", year);
    else ( year/4 = 0 )
        printf("%f is a leap year.\n", year);
    else
        printf("%d is not a leap year.\n", year);
}
```

Sample Input & Output:

Enter a year to check if it is a leap year: 2000
2000 is a leap year.

Question-15:**Problem Statement:**

Write a C-program to find the sum of all digits in a 3-digit input number (111 to 999) and print the reverse of the same input number. If the input is not within the range (111 to 999), print the error message "Input is out of range 111 to 999".

Sample Input & Output – 1:

Enter the time: 123

Sum: 6

Reverse: 321

Sample Input & Output – 2:

Enter the time: 1234

Input is out of range 111 to 999

Question-16:**Problem Statement:**

Given program is used to print the output "M.S. Dhoni" when "a" is in between 10 and 20. Print "M.E.K Hussey", when "a" is in between 21 to 30. Print "A.B. de villiers" when "a" is above 31. Otherwise, print "Wrong choice". Note that boundary values are inclusive.

Identify and correct the (syntactical as well as logical) errors in the program to meet its objective.

```
void main()
{
    printf ("\n Enter the number a: ")
    scanf ("%f", a)

    if(a >= 10 && a >= 20)
    {
        printf('M.S. Dhoni \n')

    else if(a >= 20 && a<=30)
    {
        printf('M.E.K Hussey \n')

    else if(a => 30)
    {
        printf('A.B. de villiers \n')
    else
        printf('Wrong choice\n')
    }
    }
}
```

Sample Input & Output-1:

Enter the number a: 10

M.S. Dhoni

Sample Input & Output-2:

Enter the number a: 30
M.E.K Hussey

Question-17:**Problem Statement:**

Write a C-program to order the given numbers. Receive exactly three distinct integer numbers (upto 4 digit length) from user. Print the numbers in both increasing order as well as decreasing order. If user enters less than or greater than three numbers, print the error message "Enter exactly three distinct numbers". In addition to this, when any of the two numbers are equal, print "Invalid input".

Sample Input & Output - 1:

Enter five distinct numbers: 100 -700 20
Increasing order: -700 20 100
Decreasing order: 100 20 -700

Sample Input & Output - 2:

Enter five distinct numbers: 100 -700 100
Invalid input

Question-18:**Problem Statement:**

Given program is used to check whether the value stored in a variable 'a' is equal to 5 or not. Identify and correct the (syntactical as well as logical) errors in the program to meet its objective.

```
void main
{
    int a = 2 ;

    if ((a = 6) = 5)
        scanf (a is equal to 5 \n)
    else
        scanf (a is not equal to 5 \n)
    return 0
}
```

Expected Output:

a is equal to 5

Sample Input & Output - 1:

Enter five distinct numbers: 100 -700 20 30 10
Enter your choice:
(1). Largest number
(2). Smallest number
(3). Sum of all numbers
(4). Search for a given number
1
Largest number: 1000

Sample Input & Output - 2:

Enter five distinct numbers: 100 -700 20 30 10 -10 40 80 90 1000
Enter exactly five numbers.

Sample Input & Output - 3:

Enter ten distinct numbers: 100 -700 20 30 10
Enter your choice:
(1). Largest number
(2). Smallest number
(3). Sum of all numbers
(4). Search for a given number
5
Wrong Choice

Question-19:**Problem Statement:**

Given program is used to print the following output:
Case 1: 9876
Case 2:9876
Case 3:987.65
Case 4:988
Case 5:9.876543e+02

Identify and correct the (syntactical as well as logical) errors in the program to meet its objective.

```
main()
{
    /* Prints the number right justified within 6 columns */
    printf('Case 1:%6d\n',9876)

    /* Prints the number to be right justified to 3 columns but, there are 4 digits so number is not
    right justified */
    printf('Case 2:%3d\n',9876)

    /* Prints the number rounded to two decimal places */
    printf('Case 3:%d\n',987.6543)

    /* Prints the number rounded to 0 decimal place, i.e, rounded to integer */
    printf('Case 4:%d\n',987.6543)

    /* Prints the number in exponential notation(scientific notation) */
    printf('Case 5:%e\n',987.6543)
}
```

Question – 20:

The correct output of the following C-program is 32. However, some of the operators in the computation to calculate 'res' in the program are incorrect . Hence, correct the computation with appropriate operators so that 'res' stores the correct value 32.
Also, the program is incomplete. You have to complete the program so that the correct output is

printed.

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

```
int main()
```

```
{
```

```
    unsigned int res;
```

```
    res = (64 <<(2+1-2)) && (~(1<<2));
```

```
    //print the value stored in res variable.
```

```
    return 0;
```

```
}
```

You are allowed to use only “**Basic Input/Output operations**” and “**Control Statements**” of C in your C-program for Question-2. You should not use any looping constructs like, **for**, **while**, **do .. while** in your C-program.

Question - 21:

There is an elevator in a building. If the lift-man presses 1 in the elevator, it goes up by one floor, while if he presses 2, the elevator goes down by one floor. If the lift-man presses 3, the elevator goes up by two floors.

Write a C program to control the elevator of a four storey building. The program takes two inputs, (i) A four digit number each of whose digits come from the set {1, 2, 3}, and (ii) The floor at which the elevator initially rests.

Calculate the final position of the elevator and also the path traversed by it from its initial position.

Remember that the elevator cannot move above the top floor or below the ground floor.

Sample Input :

Enter the current position of elevator: 3

Enter the number: 1233

Sample Output :

Elevator is in floor 4

Elevator is in floor 3

Elevator is in floor 4

Elevator is in floor 4

Question – 22:

Modify the following program to test whether a user given input number has '1' or '0' (in binary representation) in the user specified position. The given program is incomplete. You need to complete the program by inserting appropriate statements inside the boxes.

Sample Input/Output -1:

Enter the number to be tested: 8

Enter the bit position to be tested: 3

bit-3 is equal to 1

Sample Input/Output -2:

Enter the number to be tested: 8
Enter the bit position to be tested: 0
bit-0 is equal to 0

```
int main()
{
    int testValue; //user input value
    int bit_pos ; // bit position
    printf ("Enter the number to be tested: ") ;
    
    
    //check whether the bit_pos is valid. i.e., bit_pos should be in the range 0 to 31.
    //For bit_pos out of this range, print "Invalid bit position".
    
    if (testValue && (1 >> bit_pos))
        
        // bit_pos is equal to 1
    else
        
        // bit_pos is equal to 0
    return 0;
}
```

You are allowed to use only “**Basic Input/Output operations**” and “**Control Statements**” of C in your C-program for Question-2. You should not use any looping constructs like, **for**, **while**, **do .. while** in your C-program.

Question – 23:

Write a C program to take two single digit numbers as input. Convert them into binary. If the first number is greater than the second number and both are even, XOR them. If the first number is greater than the second number and at least one of them is odd, XNOR (XOR followed by complement) them. If the second number is greater than or equal to the first number and both are even, give a left shift to both of the numbers and then bit-wise-AND them. If the second number is greater than or equal to the first and at least one of them is odd, perform bit-wise-OR operation.

Sample Input :

Enter two single digit number: 4 2

Sample Output:

The binary values are: 0100 and 0010
XOR of 0100 and 0010 is 0110
Output : 0110

Question – 24:

Following C-program is used to illustrate the difference between the arithmetic and logical

"RIGHT" shift operations. Modify the program such that it prints (-4), (124). The given program is incomplete. You need to complete the program by inserting or modifying appropriate statements inside the boxes.

```
#include <stdio.h>
int main()
{
    //Output of arithmetic right of (-8) by one bit is (-4).
    //Hint: Sign bit at MSB will not be shifted.
    char a = -8;
    char b = a << 1;
    printf("Arithmetic right shift of (-8) by 1-bit position is %d\n",b);

    //Output of logical right of (-8) by one bit is (124).
    char c = -8 ;
    char d = c >> 2;
    printf("Logical right shift of (-8) by 1-bit position is %d\n",b);

    return 0 ;
}
```

You are allowed to use only “**Basic Input/Output operations**” and “**Control Statements**” of C in your C-program for Question-2. You should not use any looping constructs like, **for**, **while**, **do .. while** in your C-program.

Question – 25:

Write a C program which takes a three digit number. Each digit of the number is coefficient of a quadratic equation. The right most digit is the coefficient of x^0 and left most is coefficient of x^2 . Your C program will find the roots of the polynomial. If both the roots are positive it will add them up. If both the roots are negative it will multiply them. If the roots are imaginary it will print “try again”.

Input : 132

Polynomial is: $x^2 + 3x + 2$

Output : 2

Question – 26:

In a **XYZ** plane, you will be given two cubes. For each cube the user will provide it's bottom left co-ordinate (x, y, z) along with it's length. Write a C program to check if the cubes are intersecting or not. Note that cubes are crossing if they share a common point inside both of them.

Input : Given input:

1 2 3 4 where (1, 2, 3) is the bottom left co-ordinate of the first cube whose side is of length 4

5 6 7 8 where (5, 6, 7) is the bottom left co-ordinate of the second cube whose side is of length 8.

Output : Yes or No

Question – 27:

The given C-program is incomplete. You need to complete the program by inserting appropriate statements in place of the dashes so that the output given below is obtained.

64 4096
16 2048
8 1024
128 512

```
#include <stdio.h>
int main()
{
    printf("%d %d\n", 4096....., 32.....);
    printf("%d %d\n", 32....., 128.....);
    printf("%d %d\n", 1024....., 32.....);
    printf("%d %d\n", 32....., 4096.....);
    return 0;
}
```

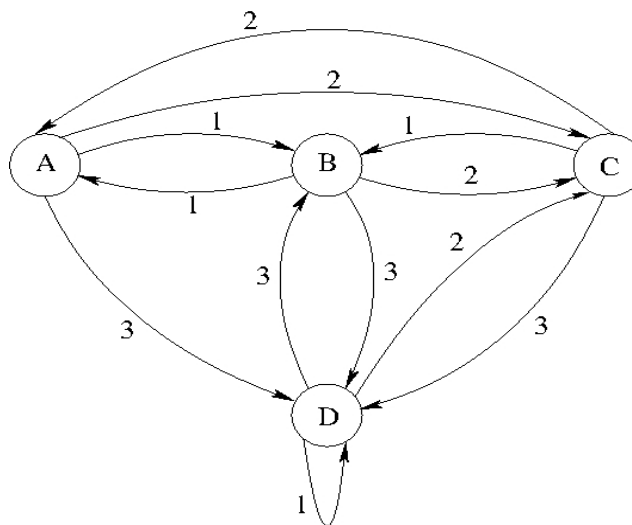
Hint: Use right/left shift operators to obtain the following output.

You are allowed to use only “**Basic Input/Output operations**” and “**Control Statements**” of C in your C-program for Question-2. You should not use any looping constructs like, **for**, **while**, **do .. while** in your C-program.

Question – 28:

The figure given below represents a labeled map connecting four cities A, B, C and D. A salesman has been given a four digit number whose digits have been taken from the set 1, 2 and 3. At a given city, the salesman reads the next digit of the number and proceeds to another city based on the label present on the transition-arc. For example, if the next number is 2 at city B, the salesman travels to city C.

Assuming that the salesman can initially be in any of the four cities, write a C-program to find the city where the salesman finally travels to after reading the four digit number from right to left.



Sample Input:

Enter the initial city: B
Enter the number:
2331

Sample Output:

Final city: A

Question – 29:

The given program is incomplete. You need to complete the program by inserting the appropriate statements inside the boxes. Using binary/unary bit-wise operators, modify the following C-program to print the output exactly same as below:

Line 1 - Value of c is 12
Line 2 - Value of c is 61
Line 3 - Value of c is 49
Line 4 - Value of c is -61
Line 5 - Value of c is 240
Line 6 - Value of c is 15

```
main()
{
    unsigned int a = 60;      /* 60 = 0011 1100 */
    unsigned int b = 13;     /* 13 = 0000 1101 */
    int c = 0;
```

```

c = a b;
printf("Line 1 - Value of c is %d\n", c );
c = a b;
printf("Line 2 - Value of c is %d\n", c );
c = a b;
printf("Line 3 - Value of c is %d\n", c );
c = a ;
printf("Line 4 - Value of c is %d\n", c );
c = a ;
printf("Line 5 - Value of c is %d\n", c );
c = a ;
printf("Line 6 - Value of c is %d\n", c );
}

```

You are allowed to use only “Basic Input/Output operations” and “Control Statements” of C in your C-program for Question-2. You should not use any looping constructs like, for, while, do .. while in your C-program.

Question – 30:

Write a C program to take a single digit number m as input. Convert it

into binary. Then give a left shift on m to get a new binary number n . If m is even, Bit-wise-OR it with n . Else, Bit-wise-OR n with itself. Check if the result after Bit-wise-OR-ing is odd or even.

Sample Input :

Enter a single digit number (m): 4

Sample Output:

The binary value of m is 100

n is: 1000

Output : 1100 & even.

Question – 31:

The given program is incomplete. You need to complete the program by inserting/correcting appropriate statements inside the boxes. Modify the following program and print the output as North=3, South=7, West=11 and East=13

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

```
_____ enum {north, south, west, east} Direction;
```

```
main()
```

```
{
```

```
Direction initialstep;
```

```
// read the user input value for initialstep
```

```
If (initialstep!=1)
```

```

        printf("current direction is North, value=%d\n",north);
    else if(initialstep-=2)
        printf("current direction is South, value=%d\n",south);
    else if(initialstep!=3)
        printf("current direction is West, value=%d\n",west);
    else if(initialstep+=4)
        printf("current direction is East, value=%d\n",east);
    else
        printf("Wrong Number\n");
    // Print the user input value multiplied by 4 using bitwise operator
}

```

You are allowed to use only “**Basic Input/Output operations**” and “**Control Statements**” of C in your C-program for Question-2. You should not use any looping constructs like, **for**, **while**, **do .. while** in your C-program.

Question – 32:

A person is depositing his capital to his saving scheme. The saving scheme has two policies.

1. If the person is depositing for three years, he will get 7% interest annually in case the capital is more than one lakh. Otherwise he will get 6.5% interest annually.
2. If the person goes for a four years scheme, he will get 6% interest annually when amount is less than one lakh. Otherwise he will get 6.75% interest annually. In this scheme he will also get Rs 2500 as maturity bonus at the end of the term.

Write a C program to take investment amount (principal) as input. The program should show all possible maturity amounts with respect to each scheme number.

Note that all interests are compound interest. That means, interest amount will be added to the principal amount at every term. The formula for estimating compounded amount is

$$A = p[(1 + r/n)^n]$$

P= principal investment

R= interest rate

n= number of times that interest is compounded per year (**in our, n is 1**).

t= number of years the money is invested.

Expected input and output

Input: 5000

Output: 8144.47

Question – 33:

The given program is incomplete. You need to complete the program by inserting the appropriate statements inside the boxes. Modify the below C-program to print the following output:

[0.000000]

[0.625000]

[20]

[2]

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

```
int main()
```

```
{
```

```
    int a = 5, b = 8;
```

```
    float c = 0.0, d = 0.0;
```

```
        c = a / b ;
```

```
        //print the value stored in c.
```

```
        d = a / b ;
```

```
        //print the value stored in d.
```

```
        //Multiply value stored in a by 4 using bit-wise operator.
```

```
        //divide value stored in b by 4 using bit-wise operator.
```

```
}
```

You are allowed to use only “**Basic Input/Output operations**” and “**Control Statements**” of C in your C-program for Question-2. You should not use any looping constructs like, **for**, **while**, **do .. while** in your C-program.

Question – 34:

In XY plane you will be given two rectangles. For each rectangle the user will provide it's bottom left co-ordinate (x, y) along with it's length and breadth.

Write a C program to check if the rectangles are intersecting or not. Note that rectangles are crossing if they share a common area.

Input : Given input:

1 2 3 4 where (1, 2) is the bottom left co-ordinate of the first rectangle whose sides are of length 3 and 4

5 6 7 8 where (5, 6) is the bottom left co-ordinate of the second rectangle

whose sides are of length 7 and 8.

Output : Yes or No

Question – 35:

Problem Statement:

Write a C-program to input your Name in the format First Mid Last (up to 20 characters including space), Roll number (exactly 8-digit natural number), Height (up to 5 digits including decimal numbers), Address (up to 50 characters including space and commas). You should use single variable to store the entire name. Likewise for Roll number, Height and Address.

Part I: Print the same information that you have entered.

Sample Input:

Enter name: Rahul Singh Pratap
Enter roll number: 140101001
Enter height: 160.5
Enter address: MG Road, New Delhi

Sample Output:

You have entered
Name: Rahul Singh Pratap
Roll Number: 140101001
Height: 160.5
Address: MG Road, New Delhi

Part II: Check whether the first name and last name have same number of characters or not. If yes, print the Roll Number. Otherwise, print Address

Question – 36:

Problem Statement:

Write a C-program to input your Name in the format First Mid Last (up to 20 characters including space), Roll number (exactly 8-digit natural number), Height (up to 5 digits including decimal numbers), Address (up to 50 characters including space and commas). You should use single variable to store the entire name. Likewise for Roll number, Height and Address.

Part I: Print the same information that you have entered.

Sample Input:

Enter name: Rahul Singh Pratap
Enter roll number: 140101001
Enter height: 160.5
Enter address: MG Road, New Delhi

Sample Output:

You have entered
Name: Rahul Singh Pratap
Roll Number: 140101001
Height: 160.5
Address: MG Road, New Delhi

Part II: Check whether the middle name and last name have same number of characters or not. If yes, print the Address. Otherwise, print Height

Question – 37:**Problem Statement:**

Write a C-program to input time in 24 hrs format into 12 Hrs format. You may store hours, mins, and secs in three different variables.

Part I: Print the input in 24 hrs format into 12hrs format.

Sample Input:

Time in 24 hrs format: 9:30:5

Sample Output:

Time in 12 hrs format: 9:30:5 AM

Sample Input:

Time in 24 hrs format: 13:30:5

Sample Output:

Time in 12 hrs format: 1:30:5 PM

Sample Input:

Time in 24 hrs format: 13:70:5

Sample Output:

Invalid time format.

Part II: Check if the time is AM or PM. If AM, then print the time as the number of milisecond of the day. If PM, calculate the LCM of the three numbers (hours, mins and secs).

Question – 38:**Problem Statement:**

Write a C-program to input time in 24 hrs format into 12 Hrs format. You may store hours, mins, and secs in three different variables.

Part I: Print the input in 24 hrs format into 12hrs format.

Sample Input:

Time in 24 hrs format: 9:30:5

Sample Output:

Time in 12 hrs format: 9:30:5 AM

Sample Input:

Time in 24 hrs format: 13:30:5

Sample Output:

Time in 12 hrs format: 1:30:5 PM

Sample Input:

Time in 24 hrs format: 13:70:5

Sample Output:

Invalid time format.

Part II: Check if the time is AM or PM. If AM, then print the time as the number of milisecond of the day. If PM, calculate the GCD of the three numbers (hours, mins and secs).

Question – 39:**Problem Statement:**

Write a C-program to print the following pattern based on user provided input (integer) number. If user enters the number **n**, then print the output containing 1 star at the 1st row. 2 stars at the second row, . . ., **n** stars at the **n**th row. All stars should be left aligned.

Part-1 : Count the number of stars

Part-2: Print the pattern

Sameple Input and Output – 1:

Enter the number of pattern lines: 3

Number of stars: 6

The pattern is

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

Sameple Input and Output – 2:

Enter the number of pattern lines: 5

Number of stars: 15

The pattern is

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

Question – 40:

Problem Statement:

Write a C-program to print the following pattern based on user provided input (integer) number. If user enters the number **n**, then print the output containing 1 at the 1st row. (2, 1) at the second row, . . . , (**n, n-1, . . . , 2, 1**) at the **n**th row. All numbers should be right aligned.

Part – 1: Count the number of time the integer Ceiling($n/2$) is printed in the pattern.

Part – 2: Print the pattern

Note:

When $n=1$, Ceiling ($n/2$)=1,

When $n=2$, Ceiling ($n/2$)=1.

When $n=3$, Ceiling ($n/2$)=2,

When $n=4$, Ceiling ($n/2$)=2,

When $n=5$, Ceiling ($n/2$)=3,

So on..

Sameple Input and Output – 1:

Enter the number of pattern lines: 3

The number of time the integer Ceiling($n/2$) is printed in the pattern: 2

The pattern is shown below

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

Sameple Input and Output – 2:

Enter the number of pattern lines: 5

The number of time the integer Ceiling($n/2$) is printed in the pattern: 3

The pattern is shown below

```

        1
       2 1
      3 2 1
     4 3 2 1
    5 4 3 2 1

```

Question – 41:

Problem Statement:

Write a C-program to print the following pattern based on user provided input (integer) number. If user enters the number **n**, then print the output containing 1 star at the **n**th row. 2 stars at the (n-1)th row, . . ., **n** stars at the **1st** row. All stars should be left aligned.

Part-1 : Count the number of stars

Part-2: Print the pattern

Sample Input and Output – 1:

Enter the number of pattern lines: 3

Number of stars: 6

Pattern is

```

* * *
* *
*

```

Sample Input and Output – 2:

Enter the number of pattern lines: 5

Number of stars: 15

Pattern is

```

* * * * *
* * * *
* * *
* *
*

```

Question – 42:

Problem Statement:

Write a C-program to print the following pattern based on user provided input (integer) number. If user enters the number **n**, then print the output containing 1 at the nth row. (4, 2) at the (n-1)th row, . . ., **((n*n), n*(n-1), . . . , 2n, n)** at the **nth** row. All numbers should be right aligned.

Part – 1: print the maximum integer

Part – 2: Print the pattern

Sameple Input and Output – 1:

Enter the number of pattern lines: 3

Max integer is 9

Pattern is

```
9  6  3
  4  2
    1
```

Sameple Input and Output – 2:

Enter the number of pattern lines: 5

Max integer is 25

Pattern is

```
25 20 15 10 5
  16 12  8  4
    9  6  3
      4  2
        1
```

Question – 43:

Problem Statement:

Write a C-program to input date in day:month:year format. You may store day, month, and year in three different variables.

Part I: Print the input in ddmmyy

Sample Input: 5:2:16

Sample Output: 050216

Sample Input:

30:2:16

Sample Output:

Invalid date

Part II: Check if the year is leap year or not. If yes, then print the input date in 5th Feb. 2016 format. If no, calculate the LCM of the three numbers (day, month and year).

Note: to simplify your program, you may consider only three months Jan, Feb and Mar.

Question – 45:

Problem Statement:

Write a C-program to input time in 24 hrs format into 12 Hrs format. You may store hours, mins, and secs in three different variables.

Part I: Print the input in 24 hrs format into 12hrs format.

Sample Input:

Time in 24 hrs format: 9:30:5

Sample Output:

Time in 12 hrs format: 9:30:5 AM

Sample Input:

Time in 24 hrs format: 13:30:5

Sample Output:

Time in 12 hrs format: 1:30:5 PM

Sample Input:

Time in 24 hrs format: 13:70:5

Sample Output:

Invalid time format.

Part II: Check if the time is AM or PM. If AM, then print the time as the number of milisecond of the day. If PM, calculate factioal of the hour.

Question – 46:

Problem Statement:

Write a C-program to input your name in the format First Mid Last (up to 40 characters including space), Roll number (exactly 8-digit natural number), Height (up to 5 digits including decimal

numbers), Address (up to 50 characters including space and commas) and the last name of your best friend. You should use single variable to store the entire name. Likewise for Roll number, Height, Address and last name.

Part I: Print the same information that you have entered.

Sample Input:

Enter name: Rahul Singh Pratap
Enter roll number: 140101001
Enter height: 160.5
Enter address: MG Road, New Delhi
Last name of my friend : Deb

Sample Output:

You have entered
Name: Rahul Singh Pratap
Roll Number: 140101001
Height: 160.5
Address: MG Road, New Delhi
Last name of my friend : Deb

Part II: Check whether your last name and your friend last name are same or not. If no, then exchange the first name and last name in the same same variable. Otherwise, print Address

Your output should be: Pratap Singh Rahul

Note: If the variable that you have used to store your name is A. You should perform the exchange operation on the same variable A. After your exchange operation, variable A should contain “Pratap Singh Rahul”.

Question – 47:

Problem Statement:

Write a C-program to input your name in the format First Mid Last (up to 40 characters including space), Roll number (exactly 8-digit natural number), Height (up to 5 digits including decimal numbers), Address (up to 50 characters including space and commas) and the last name of your best friend. You should use single variable to store the entire name. Likewise for Roll number, Height, Address and last name.

Part I: Print the same information that you have entered.

Sample Input:

Enter name: Rahul Singh Pratap
Enter roll number: 140101001
Enter height: 160.5
Enter address: MG Road, New Delhi

Last name of my friend : Deb

Sample Output:

You have entered
Name: Rahul Singh Pratap
Roll Number: 140101001
Height: 160.5
Address: MG Road, New Delhi
Last name of my friend : Deb

Part II: Check whether your last name and your friend last name are same or not. If no, then replace your last name by your friend's last name. Otherwise, print Address

Your output should be: Deb Singh Prarap

Note: If the variable that you have used to store your name is A. You should perform the repacenment operation on the same variable A. After the replacement operation, variable A should contain "Pratap. Singh Rahul".

Question – 48:

Problem Statement:

Write a C-program to input your Height (up to 5 digits including decimal numbers) in cm.

Part I: Print the same information that you have entered.

Sample Input:

Enter height: 160.5

Sample Output:

My Height: 160.5 cm

Part II: Check if your height is divisible by 2 or not. If no, print the digits one after another. If yes, find the GCD of the first two digits

Output: 1, 6, 0, and 5.

If the input height is 160.4, then your output will be GCD of 1 and 6 i.e., 1.

Question – 49:

Problem Statement:

Write a C-program to input your Height (up to 5 digits including decimal numbers) in cm.

Part I: Print the same information that you have entered.

Sample Input:

Enter height: 160.5

Sample Output:

My Height: 160.5 cm

Part II: Check if your height is divisible by 2 or not. If no, print the digits in reverse order. Otherwise, print the digits one after another.

Output: 5.061

If the input height is 160.0, then your output will be 1, 6, and 0.

Question – 50:

Problem Statement:

Write a C-program to print the following pattern based on user provided input (integer) number. If user enters the number **n**, then print the output containing 1 star at the 1st row. 2 stars at the second row, . . . , **n** stars at the **n**th row. All stars should be center-aligned.

Part – 1: Count the number of starts

Part – 2: Print the pattern

Sample Input and Output – 1:

Enter the number of pattern lines: 3

Number of starts is : 6

The pattern is printed below

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

Sample Input and Output – 2:

Enter the number of pattern lines: 5

Number of starts is : 15

The pattern is printed below

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

Question – 51:

Problem Statement:

Write a C-program to print the following pattern based on user provided input (integer) number. If user enters the number **n**, then print the output containing 1 at the 1st row. (1, 2) at the second row, . . ., (**1, 2, ..., n-1, n**) at the **n**th row. All numbers should be center aligned.

Part – 1: Count the number of time the integer Ceiling($n/2$) is printed in the pattern.

Part – 2: Print the pattern

Note:

When $n=1$, Ceiling ($n/2$)=1,

When $n=2$, Ceiling ($n/2$)=1.

When $n=3$, Ceiling ($n/2$)=2,

When $n=4$, Ceiling ($n/2$)=2,

When $n=5$, Ceiling ($n/2$)=3,

So on..

Sameple Input and Output – 1:

Enter the number of pattern lines: 3

The number of time the integer Ceiling($n/2$) is printed in the pattern: 2

The pattern is shown below

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

Sameple Input and Output – 2:

Enter the number of pattern lines: 5

The number of time the integer Ceiling($n/2$) is printed in the pattern: 3

The pattern is shown below

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Question – 52:

Problem Statement:

Write a C-program to print the following pattern based on user provided input (integer) number.

If user enters the number **n**, then print the output containing 1 star at the **n**th and the (**n**+1)th row. 2 stars at the (**n**-1)th and the (**n**+2)th row, . . ., **n** stars at the **1st** and **2nd** row. All stars should be center aligned.

Part – 1: Count the number of starts

Part – 2: Print the pattern

Sample Input and Output – 1:

Enter the number: 3

Number of starts is : 12

The pattern is printed below

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

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

Question – 53:

Problem Statement:

Write a C-program to print the following pattern based on user provided input (integer) number. If user enters the number **n**, then print the output containing 1 at the 1st row. (2, 3) at the second row, (4, 5, 6) at the third row, and so on. All numbers should be right aligned.

Part – 1: Print the maximum integer..

Part – 2: Print the pattern

Sample Input and Output – 1:

Enter the number of pattern lines: 3

Max integer is 6

Pattern is

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

Sample Input and Output – 2:

Enter the number of pattern lines: 5

Max integer is 15

Pattern is

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

Question – 54:

Problem Statement:

Write a C-program to input your Height in mm.

Part I: Print the same information that you have entered.

Sample Input:

Enter height: 1600

Sample Output:

My Height: 1600 mm

Part II: Check if your height is divisible by 2 or not. If no, print the digits one after another. If yes, find its prime factors.

Question – 55:**Problem Statement:**

Write a C-program to input your Height (up to 5 digits including decimal numbers) in ft.

Part I: Print the same information that you have entered.

Sample Input:

Enter height: 5.3

Sample Output:

My Height: 5.3 ft

Part II: Check if your height is divisible by 2 or not. If no, find the factorial of the integer part of your height. Otherwise, print the digits one after another.

Question – 56:**Problem Statement:**

Write a C function which takes a natural number n as argument and calculates the sum of the digits raised to the power of the number of digits in n . Check whether the number is an Armstrong number or not. An Armstrong numbers are the numbers which equal the sum of their own digits raised to the power of the number of digits.

Example: $153 = 1^3 + 5^3 + 3^3$

153 is an Armstrong number

$145 \neq 1^3 + 4^3 + 5^3$

145 is not an Armstrong number

Part – 1: Part I: Given a natural number, print its digits

Input: 145

Output: 1, 4, and 5

Part-2: Check if the natural number is an Armstrong number or not.

Output: 145 is not an Armstrong number.

Note: Some of the Armstrong numbers are 0,1, 370, 371 and 407

Question – 57:

Problem Statement:

Write a C function which takes a natural number as argument and calculates the product of the sum of its digits and the product of its digits. A Sum-product number is a natural number that equals the product of the sum of its digits and the product of its digits.

Example: $144 = (1+4+4) \times (1 \times 4 \times 4) = 9 \times 16 = 144$

144 is a sum-product number.

Part I: Given a natural number, print its digits

Input: 144

Output: 1, 4, and 4

Part II: Check if the natural number is a sum-product number.

Output: 144 is a sum product number.

Other sum product numbers are 0, 1 and 135

Question – 58:

Problem statement

Write a C function to determine if a number is prime. Given a number k print all the prime factors of k using CheckPrime(). If the number itself is prime print “IT IS A PRIME NUMBER”.

Sample Input:

Enter a number: 15

Sample Output:

The prime factors of 15 are: 3, 5

Sample Input:

Enter a number: 17

Sample Output:

IT IS A PRIME NUMBER

Question – 59:

Problem Statement:

Write a C function to calculate factorial of a number. Given two numbers n and p, check if factorial of P is divisible by n. You can not use modulo operation for checking divisibility.

Sample Input-output:

Part -1:

Enter the numbers n and p: 2 and 3

Factorial of 2 is 2

Factorial of 3 is 6

Part-2:

3! Is divisible by 2.

Question – 60:

Write a C program to develop a calculator to perform addition and subtraction operations of two integer number. The addition and subtraction operations should be implemented using functions i.e., one add function and one subtraction function. Depending on the choice of operation, your calculator should perform either addition or subtraction.

Note: you should take + and – symbols as the operator symbols

Part – 1: Perform addition

Input 1:

Enter the numbers : 10, 13

Operation: +

Output: 23

Input 2

Enter the numbers : -10, 13

Operation: +

Output: 3

Part -2: Perform subtraction

Input 1:

Enter the numbers : 10, 13

Operation: -

Output: -3

Input 2

Enter the numbers : -10, 13

Operation: -

Output: -23

Question – 61:

Problem Statement:

Write a C program to develop a calculator to perform multiplication and division operations of two integer number. The multiplication and division operations should be implemented using functions i.e., one multiplication function and one division function. Depending on the choice of operation, your calculator should perform either multiplication or division.

Note: you should take * and % symbols as the operator symbols

Part – 1: Perform addition

Input 1:

Enter the numbers : 10, 13

Operation: *

Output: 130

Input 2

Enter the numbers : -10, 13

Operation:*

Output: -130

Part -2: Perform subtraction

Input 1:

Enter the numbers : 10, 13

Operation: %

Output: 0.769

Input 2

Enter the numbers : -10, 13

Operation: -

Output: -0.769

...

Question – 62:

Write a C function to estimate product of two polynomials of degree 2. Your program should take two polynomials as input and use the function to perform polynomial sum. You can use as many variables as you need.

Input:

Enter your first polynomial.

Degree of first polynomial : 2

Co-eff of degree 2 :2

Co-eff of degree 1 :3

Co-eff of degree 0 :-1

Enter your first polynomial.

Degree of 2nd polynomial : 2

Co-eff of degree 2 :4

Co-eff of degree 1 :2

Co-eff of degree 0 :1

Part – 1: Print the two polynomials

Output:

Polynomial 1: 2^2+3^1-1

Polynomial 2: 4^2+2^1+1

Part -2: print the product of the two polynomials

Output:

$8^4+16^3-2^2-1^1$

Question – 63:

Problem Statement:

Write a C function to estimate division of two polynomial of degree 2 by another polynomial of degree 1. Your program should take to polynomials as inputs and use the function to perform polynomial division. You can use as many variables as you need.

Input:

Enter your first polynomial.

Degree of dividend polynomial : 2

Co-eff of degree 2 :3

Co-eff of degree 1 :5

Co-eff of degree 0 :2

Enter your divisor polynomial.

Degree of 2nd polynomial : 1

Co-eff of degree 1 :2

Co-eff of degree 0 :1

Part – 1: Print the two polynomials

Output:

Polynomial 1: 3^2+5^1+2

Polynomial 2: 2^1+1

Part -2: print result of the division

Output:

Quotient: $3/2^1 + 7/4$

Remainder: $1/4$

You can also show the floating number if the co-efficients.

Question – 64:

Write a C program to take a number n as input and then print a triangle containing n rows in the following form.

```
      1
     2 3 2
    3 4 5 4 3
   4 5 6 7 6 5 4
```

Use a C function to calculate the k^{th} term of the i^{th} row, given i and k as arguments.

Part-1: Print the pattern for arbitrary input n

Part-2: Count the number of natural number printed in the pattern

Question – 65:

Problem Statement:

Write a C program to take a number n as input and then print a triangle containing n+1 rows in the following form.

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

Use a C function to print the pattern in i^{th} row, given i as arguments. [Hint: k^{th} term of the i^{th} row is ${}^k C_i$].

Part-1: Print the pattern for arbitrary input n

Part-2: Count the number of natural number printed in the pattern

Question – 66:

Write a C program to develop a calculator which supports addition and subtraction operations. The addition and subtraction operations of two integer should be implemented using functions i.e., one add function and one subtraction function. Your calculator should support sequence of addition and subtraction of arbitrary length.

Ex: $10+3-4+3+6-1=16$

Note: you should take + and – symbols as the operator symbols. You can use any input

pattern/format suitable to you. However, your result should reflect solution to the expression of arbitrary length in run time. The same program should be able to perform **10+3-4+3+6-1=16 as well as 98-20+2=80.**

Part – 1: Print your input as proper expression

One possible Input pattern 1:

Enter the number : 10

Operation: +

Enter the number : 5

Operation: -

Enter the number : 1

Operation: +

Enter the number : 2

Operation: e

******Note: e is terminating symbol**

Output: 10+5-1+2

Part -2: estimate the expression

Output: 16

Question – 67:

Problem Statement:

Write a C program to develop a calculator which supports multiplication and division operations. The multiplication and division operations of two integer should be implemented using functions i.e., one multiplication function and one division function. Your calculator should support sequence of multiplication and division of arbitrary length.

Ex: 10*3/3=10

Note: you should take * and / symbols as the operator symbols. You can use any input pattern/format suitable to you. However, your result should reflect solution to the expression of arbitrary length in run time. The same program should be able to perform **10*3/3=10 as well as 10*3=30. The operators follow left association i.e., left operators are applied before the right operators.**

Part – 1: Print your input as proper expression

One possible Input pattern 1:

Enter the number : 10

Operation: *

Enter the number : 3

Operation: /

Enter the number : 3

Operation: *

Enter the number : 2

Operation: e

Note: e is terminating symbol

Output: $10*3/3*2$

Part -2: estimate the expression

Output: 20

Question – 68:

Problem Statement:

Write a C function to calculate factorial of a number k where k is the argument to the Fuction. Write a C program using the factorial function to estimate factorion of a natural number. Factorian of a natural number is the sum of the factorials of its decimal digits.

Example: $145=1!+4!+5!$

Part I: Given a natural number, print its digits

Input: 145

Output: 1, 4, and 5

Part II: Print the factorion of the natural number

Output: 145

Question – 69:

Problem Statement:

Write a C function to calculate sum of Fibonacci series upto k^{th} term where k is the argument to the Fuction. Write a C program using the fibonacci sum function to estimate sum of the Fibonacci sum of the decimal degits of a natural number n.

Sum upto 4th Fibonacci term = $1+1+2+3$ i.e., $\text{FibonacciSum}(4) = 7$

For a natural number 145, sum of fibonacci sum is $\text{FibonacciSum}(1)+\text{FibonacciSum}(4) + \text{FibonacciSum}(5)=1+7+12=20$

Part I: Given a natural number, print its digits

Input: 145

Output: 1, 4, and 5

Part II: Print the **sum of the fibonacci sum of the natural number**

Output: 20

Question – 70:

Write a function in C to convert a k digit octal number (given as input) to a binary number. Your program should provide the following outputs. Store the octal number either to an int variable or char array.

Part-1: Check the given number is a octal number.

Part-2: If the number of digits is equal to k, print its binary representation. Otherwise print invalid number.

Sample Input-output:

Part -1:

Enter a octal number: 63

Yes it is an octal number

Part-2:

Binary representation of 63 is 101011

Question – 71:

Problem Statement:

Write a function in C to convert a k digit binary number given as input to a decimal number. Your program should provide the following outputs. Store the binary number either to an int variable or char array.

Part-1: Check the number is a binary number.

Sample Input: 110101

Output: Yes it is a binary number

Sample Input: 12001

Output: No, the given number is not a binary number

Part-2: If the binary number has k digits, print its decimal representation.

Sample Input:

Enter a binary number: 111111

Sample Output: 63

Question – 72:

Write a C function to calculate the GCD of two numbers taken as argument. Using this function print all the co-prime (GCD is 1) pairs between 10 and 20.

Part – 1: Print GCD of 10 and 14

Output: GCD of 10 and 14 is 2.

Part -2: Print the list of co-prime pairs between 10 and 20

Output: <10,11>, <10,13> ...

Note: only few of them are given above. Your program should find the rest.

Question – 73:

Problem Statement:

Write a C function to estimate sum of first n natural number where n is taken as argument. Using this function print all the sum of first n natural number between the range 1 to 10.

Part – 1: Print the sum of first 10 natural number.

Output: 55

Part -2: print all the sum of first n natural number between the range 1 to 10

Output:

Sum of first 1 natural number : 1

Sum of first 2 natural number : 3

Sum of first 3 natural number : 6

...

Question – 74:

Write a C function to determine prime factors of a number taken as argument. Using this function, print the prime factors of first n natural numbers. You should write another function to check the given factor is a prime or not.

Input:

Enter a natural number: 10

Part – 1: Check the given natural number is a prime or not.

Output: The given number 10 is a prime.

Part -2: Print the prime factors of the first n natural numbers upto the given number.

Output:

Prime factor of 1: no prime factor exist.

Prime factor of 2: 2

Prime factor of 3: 3

Prime factor of 4: 4

Prime factor of 5: 5
Prime factor of 6: 2, 3
And so on.

Question – 75:

Problem Statement:

Write a C function to estimate sum of two polynomials of degree 2. Your program should take to polynomials as input and use the function to perform polynomial sum. You can use as many variables as you need.

Input:

Enter your first polynomial.
Degree of first polynomial : 2
Co-eff of degree 2 :2
Co-eff of degree 1 :3
Co-eff of degree 0 :-1

Enter your first polynomial.
Degree of 2nd polynomial : 2
Co-eff of degree 2 :4
Co-eff of degree 1 :2
Co-eff of degree 0 :1

Part – 1: Print the two polynomials

Output:

Polynomial 1: 2^2+3^1-1
Polynomial 2: 4^2+2^1+1

Part -2: print sum of the two polynomial

Output:

6^2+4^1

Question – 76:

Problem Statement:

Consider an array with 10 integer numbers. Write a function to find the smallest integer

present in that array. Your program should provide the following output.

Part-1: Print the array that you have entered.

Sample Input: 7 9 1 10 3 19 7 3 8

Sample output: 7 9 1 10 3 19 7 3 8

Part 2: Print the smallest number

Sample Output: 1

Question – 77:

Problem Statement:

Consider an array with 10 integer numbers. Write a function to find the largest integer present in that array. Your program should provide the following output.

Part-1: Print the array that you have entered.

Sample Input: 7 9 1 10 3 19 7 3 8

Sample output: 7 9 1 10 3 19 7 3 8

Part 2: Print the largest number

Sample Output: 19

Question – 78:

Problem Statement:

Write a function which takes a sequence of strings in a single variable as its input and check whether the input string is pallindrome or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and count the number of characters in your name.

Sample Input: Rahul Khanna

Sample output: 11

Part 2: Check if the input is a pallindrom using the function.

Sample Output: Your name Rahul Khanna is not a pallindrome.

Note:

1. Madam is a pallindrom

2. Put it up is a pallindrom
3. Taco cat is a pallindrom

Question – 79:

Problem Statement:

Write a function which takes your full name (first middle last) as its input and check whether the last name and reverse of the first name are same or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and count the number of characters.

Sample Input: Rahul Kr luhar

Sample output: 12

Part 2: Check if the last name and reverse of the first name are same or not using the function.

Sample Output: Last name luhar and reverse of the firsr name rahul are same.

Question – 80:

Problem Statement:

Write a function which takes your full name (first middle last) as its input and check whether the last name and the first name are same or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and count the number of characters in the last name.

Sample Input: Rahul Kr luhar

Sample output: 5

Part 2: Check if the last name and the first name are same or not using the function.

Sample Output: Last name luhar and the firsr name rahul are not same.

Question – 81:

Problem Statement:

Write a function which takes your full name (first middle last) as its input and check whether the last name is a pallindrome or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and count the number of characters in the first name.

Sample Input: Rahul Kr luhar

Sample output: 12

Part 2: Check if the last name is pallindrome or not using the function.

Sample Output: Last name luhar is not a pallindrome.

Note: The words Madam, Racecar are pallindrome.

Question – 82:

Problem Statement:

An array of 20 integers is given. Write a function that will reverse the elements in the array. Your program should print the following.

Part-1: Print the array that you have entered.

Sample Input: 7 9 1 10 3 19 7 3 8

Sample output: 7 9 1 10 3 19 7 3 8

Part 2: Reverse the order of the elements in the array and print.

Sample Output: 8 3 7 19 3 10 1 9 7

Question – 83:

Problem Statement:

Consider an array of integer data type. Write a function to count the number of occurrences of a given number in the array. Your program should give the following output.

Part-1: Print the array that you have entered.

Sample Input: 12 0 11 7 8 0 7 12 8 7 6

Output:

Part-2: Count the number of occurrences of the number 12.

Sample Output:

12 present 2 times

Question – 84:

Problem Statement:

Write a function which takes a sequence of strings in a single variable as its input and check whether the input string is pallindrome or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and print it.

Sample Input: Rahul Khanna

Sample output: Rahul Khanna

Part 2: Check if it is a pallindrom using the function.

Sample Output: Your name Rahul Khanna is not a pallindrome.

Note:

1. Madam is a pallindrom
2. Put it up is a pallindrom
3. Taco cat is a pallindrom

Question – 85:

Problem Statement:

Write a function which takes your full name (first middle last) as its input and check whether the first name and reverse of the last name are same or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and print it.

Sample Input: Rahul Kr luhar

Sample output: Rahul Kr luhar

Part 2: Check if the first name and reverse of the last name are same or not using the function.

Sample Output: Fisir name Rahul and reverse of the last name luhar are same.

Question – 86:

Problem Statement:

Write a function which takes a sequence of strings in a single variable as its input and check whether the input string is pallindrome or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and count the number of characters in your name.

Sample Input: Rahul Khanna

Sample output: 11

Part 2: Check if it is a pallindrom using the function.

Sample Output: Your name Rahul Khanna is not a pallindrome.

Note:

4. Madam is a pallindrom
5. Put it up is a pallindrom
6. Taco cat is a pallindrom

Question – 87:

Problem Statement:

Write a function which takes your full name (first middle last) and character **ch** as its inputs and count the number of occurrences of the character **ch** in your name. Your program should take your full name in a SINGLE variable as input and produce the following outputs.

Part-1: Enter your name in a single variable and count the number of characters present in your name.

Sample Input: Rahul Kr luhar

Sample output: 12

Part 2: Print the number of occurrences of a character ch in your name using the function.

Input: r

Sample Output:3.

Question – 88:

Problem Statement:

Write a function which takes your full name (first middle last) as its input and check whether the last name and the first name are same or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and count the number of characters in the last name.

Sample Input: Rahul Kr luhar

Sample output: 5

Part 2: Check if the last name and the first name are same or not using the function.

Sample Output: Last name luhar and the first name rahul are not same.

Question – 89:

Problem Statement:

Write a function which takes your full name (first middle last) as its input and check whether the last name is a pallindrome or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and count the number of characters in the first name.

Sample Input: Rahul Kr luhar

Sample output: 12

Part 2: Check if the last name is pallindrome or not using the function.

Sample Output: Last name luhar is not a pallindrome.

Note: The words Madam, Racecar are pallindrome.

Question – 90:

Problem Statement:

Write a function which takes an array of integers of size n as its input and find the index of the minimum and maximum elements in the array. Your program should read the array from user and produce the following output.

Part-1: Enter the array of integers and print the array

Sample Input:

Size of the array: 5

Enter the elements:

1ST ELEMENT: 4

2nd ELEMENT: 3

3rd ELEMENT: 6

4th ELEMENT: 5

5th ELEMENT: 9

Sample output: 4 3 6 5 9

Part 2: Find the index of the minimum and maximum elements in the array and print them

Sample Output:

Index of the Minimum: 1
Index of the Maximum: 4

Question – 91:

Problem Statement:

Write a function which takes an array of integers of size n as its input and find the minimum element in the first half of the array and maximum element in the second half. Your program should read the array from user and produce the following output.

Part-1: Enter the array of integers and print the array in reverse order

Sample Input:

Size of the array: 5

Enter the elements:

1ST ELEMENT: 4

2nd ELEMENT: 3

3rd ELEMENT: 6

4th ELEMENT: 5

5th ELEMENT: 9

Sample output: 9 5 6 3 4

Part 2: find the minimum element in the first half of the array and maximum element in the second half and print them

Sample Output:

Output:

Min: 5

Max:4

Note: for the arrays:

1 3 5 7 8 9

and

4 2 8 6 5

Respective output should be

Min: 1

Max: 9

and

Min: 2

Max: 6

Question – 92:

Problem Statement:

Write a function which takes an array of integers of size n as its input and find the list of unique elements in the array. Your program should read the array from user and produce the following output.

Part-1: Enter the array of integers and print the array

Sample Input:

Size of the array: 5

Enter the elements:

1ST ELEMENT: 1

2nd ELEMENT: 3

3rd ELEMENT: 1

4th ELEMENT: 2

5th ELEMENT: 1

Sample output: 1 3 1 2 1

Part 2: find the list of unique elements in the array and print them.

Sample Output:

List of unique elements: 1 2 3 (the list may be printed in any order)

Question – 93:**Problem Statement:**

Write a function which takes an array of integers of size n as its input and remove repeated elements. Your program should read the array from user and produce the following output.

Part-1: Enter the array of integers and print the array

Sample Input:

Size of the array: 5

Enter the elements:

1ST ELEMENT: 1

2nd ELEMENT: 3

3rd ELEMENT: 1

4th ELEMENT: 2

5th ELEMENT: 1

Sample output: 1 3 1 2 1

Part 2: remove repeated elements and print the resulted array.

Sample Output:

Array after removing repeated elements: 1 3 2

Question – 94:**Problem Statement:**

Write a function which takes a sequence of strings in a single variable as its input and check whether the input string is pallindrome or not. Your program should take your full name in a SINGLE variable as input and produce the following output.

Part-1: Enter your name in a single variable and count the number of characters in your name.

Sample Input: Rahul Khanna

Sample output: 11

Part 2: Check if it is a pallindrom using the function.

Sample Output: Your name Rahul Khanna is not a pallindrome.

Note:

7. Madam is a pallindrom
8. Put it up is a pallindrom
9. Taco cat is a pallindrom

Question – 95:

Problem Statement:

Write a function which takes your full name (first middle last) and character **ch** as its inputs and count the number of occurrences of the character **ch** in your name. Your program should take your full name in a SINGLE variable as input and produce the following outputs.

Part-1: Enter your name in a single variable and count the number of characters present in your name.

Sample Input: Rahul Kr luhar

Sample output: 12

Part 2: Print the number of occurrences of a character ch in your name using the function.

Input: r

Sample Output:3.

Question – 96:

Problem Statement:

Write a function which takes an array of integers of size n as its input and find the minimum and maximum elements in the array. Your program should read the array from user and produce the following output.

Part-1: Enter the array of integers and print the array

Sample Input:

Size of the array: 5

Enter the elements:

1ST ELEMENT: 4

2nd ELEMENT: 3

3rd ELEMENT: 6

4th ELEMENT: 5

5th ELEMENT: 9

Sample output: 4 3 6 5 9

Part 2: find the minimum and maximum elements in the array and print them

Sample Output:

Minimum: 3

Maximum: 9

Question – 97:

Problem Statement:

Write a function which takes an array of integers of size n as its input and exchange the first half of the elements with that of the second half i.e., exchange the first element of second half with that of the first half, second element of second half with that of the first half and so on. Your program should read the array from user and produce the following output.

Part-1: Enter the array of integers and print the array in reverse order

Sample Input:

Size of the array: 5

Enter the elements:

1ST ELEMENT: 4

2nd ELEMENT: 3

3rd ELEMENT: 6

4th ELEMENT: 5

5th ELEMENT: 9

Sample output: 9 5 6 3 4

Part 2: exchange the first half of the elements with that of the second half in the same order and print the array

Sample Output:

Output: 3 4 6 9 5

Note: for the arrays:

1 3 5 7 8 9

and

4 2 6 8 5

Respective output should be

7 8 9 1 3 5

and

8 5 6 4 2

Question – 98:

Problem Statement:

Write a function which takes an array of integers of size n as its input and find the number of occurrences of each unique element in the array. Your program should read the array from user and produce the following output.

Part-1: Enter the array of integers and print the array

Sample Input:

Size of the array: 5

Enter the elements:

1ST ELEMENT: 1

2nd ELEMENT: 3

3rd ELEMENT: 1

4th ELEMENT: 2

5th ELEMENT: 1

Sample output: 1 3 1 2 1

Part 2: find the list of unique elements in the array and print them.

Sample Output:

1 occurs for 3 times

3 occurs for 1 time

2 occurs for 1 time

Question – 99:**Problem Statement:**

Write a function which takes an array of integers of size n as its input and maximum occurring elements. Your program should read the array from user and produce the following output.

Part-1: Enter the array of integers and print the array

Sample Input:

Size of the array: 5

Enter the elements:

1ST ELEMENT: 1

2nd ELEMENT: 3

3rd ELEMENT: 1

4th ELEMENT: 2

5th ELEMENT: 1

Sample output: 1 3 1 2 1

Part 2:

Sample Output: maximum occurring element and print

Maximum occurring element: 1