SRM VALLIAMMAI ENGINEERING COLLEGE

(AN AUTONOMOUS INSTITUTION)

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF GENERAL ENGINEERING

LAB MANUAL

I SEMESTER

1901010 - ‘C’ PROGRAMMING LABORATORY

Regulation – 2019

Academic Year 2022 – 2023 (ODD SEMESTER)

Prepared by

Dr. S.K.Saravanan, AP(Sel.G) / CSE	**Dr. S.Parthasarathy, AP(Sel.G) / CSE**

SYLLABUS

1901010 C PROGRAMMING LABORATORY L T P C
004 2

OBJECTIVES:

- To develop programs in C using basic constructs.
- To develop applications in C using arrays and functions.
- To develop applications in C using Strings and Structures.

List of Programs

1. Programs using I/O statements and expressions.
2. Programs using decision-making constructs.
3. Write a program to find whether the given year is leap year or Not? (Hint: not everycenturion year is a leap. For example 1700, 1800 and 1900 is not a leap year)
4. Write a program to perform the Calculator operations, namely, addition, subtraction, multiplication, division and square of a number.
5. Check whether a given number is Armstrong number or not?
6. Check whether a given number is odd or even?
7. Write a program to perform factorial of a number.
8. Write a C program to find out the average of 4 integers.
9. Show how to display array elements using two dimensional array.
10. Write a C program to perform swapping using function.
11. Display all prime numbers between two intervals using functions.
12. Reverse a sentence using recursion.
13. Write a program in C to get the largest element of an array using the function.
14. Write a C program to concatenate two string.
15. Write a C program to find the length of String.
16. Find the frequency of a character in a string.
17. Write a C program to Store Student Information in Structure and Display it.
18. The annual examination is conducted for 10 students for five subjects. Write a program to read the data and determine the following:
    (a) Total marks obtained by each student.
    (b) The highest marks in each subject and the marks of the student who secured it.
    (c) The student who obtained the highest total marks.

TOTAL: 60 PERIODS

COURSE OUTCOMES:

Upon completion of the course, the students will be able to

- Develop C programs for simple applications making use of basic constructs, arrays and strings.
- Develop C programs involving functions, recursion, pointers, and structures.
- Design applications using sequential and random access file processing.

INDEX

Ex.No. : 1 Program using I/O Statements and Expressionsfor sum of odd and even numbers.

Date :

Aim

To write a C Program to perform I/O statements and expressionsfor sum of odd and even numbers..

ALGORITHM

1. Start
2. Declare variables and initializations
3. Read the Input variable.
4. Using I/O statements and expressions for computational processing.
5. Display the output of the calculations.
6. Stop

PROGRAM

```
/*
 * Sum the odd and even numbers, respectively, from 1 to a given upperbound.
 * Also compute the absolute difference.
 * (SumOddEven.c)
 */
#include <stdio.h> // Needed to use IO functions
int main()
{
        int sumOdd  = 0; // For accumulating odd numbers, init to 0

        int sumEven = 0; // For accumulating even numbers, init to 0

        int upperbound;  // Sum from 1 to this upperbound

        int absDiff;     // The absolute difference between the two sums

        int number = 1;

         // Prompt user for an upperbound
        printf("Enter the upper bound: ");

        scanf("%d", &upperbound);   // Use %d to read an int

         // Use a while-loop to repeatedly add 1, 2, 3,..., to the upperbound
        while (number <= upperbound)
        {

                if (number % 2 == 0)

                        sumEven += number;   // Add number into sumEven

                else

                        sumOdd += number;    // Add number into sumOdd

                ++number; // increment number by 1

        }
         // Compute the absolute difference between the two sums
        if (sumOdd>sumEven)

                absDiff = sumOdd - sumEven;

        else

                absDiff = sumEven - sumOdd;

         // Print the results

        printf("The sum of odd numbers is %d.\n", sumOdd);

        printf("The sum of even numbers is %d.\n", sumEven);

        printf("The absolute difference is %d.\n", absDiff);

        return 0;

}
```

OUTPUT

```
Enter the upper bound: 1000

The sum of odd numbers is 250000.

The sum of even numbers is 250500.

The absolute difference is 500.
```

RESULT:

Thus a C Program using i/o statements and expressions was executed and the output was obtained.

Ex.No: 2A Program using Decision-Making Constructs- Pay Calculation.

DATE :

AIM

To write a C Program to perform decision-making constructs- Pay Calculation.

ALGORITHM

```
1. Start
2. Declare variables and initializations
3. Read the Input variable.
4. Codes are given to different categories and da is calculated as follows:
   For code 1,10% of basic salary.
   For code 2, 15% of basic salary.
   For code 3, 20% of basic salary.
   For code >3 da is not given.
5. Display the output of the calculations .
6. Stop
```

PROGRAM

```
#include <stdio.h>
#include<conio.h>
void main ()
{
        float basic , da , salary ;
        int code ;
        char name[25];
        da=0.0;
        printf("Enter employee name\n");
        scanf("%[^\n]",name);
        printf("Enter basic salary\n");
        scanf("%f",&basic);
        printf("Enter code of the Employee\n");
        scanf("%d",&code);
        switch (code)
        {
                case 1:
                        da = basic * 0.10;
                        break;
                case 2:
                    da = basic * 0.15;
                    break;
                case 3:
                        da = basic * 0.20;
                        break;
                default:
                    da = 0;
        }
        salary = basic + da;
        printf("Employee name is\n");
        printf("%s\n",name);
        printf ("DA is %f and Total salary is =%f\n",da, salary);
        getch();
}
```

OUTPUT

```
        Enter employee name
        sriram
        Enter basic salary
        5000
        Enter code of the Employee
        1
        Employee name is
        sriram
        DA is 500.000000 and Total salary is =5500.000000
```

RESULT

Thus a C Program using decision-making constructs was executed and the output was obtained.

2B. Program to find if a number is Negative, Positive or Zero.

Aim:

To write a C program to find if a number is negative, positive or zerousingif ... else if ... else statement.

Algorithm:

1. Start the program
2. Get the number
3. Check the number if it is negative, positive or equal to using if statement.
4. If the number is < 0print number is negative, else if the number is >0 print it is positive else the number =0.
5. Display the result
6. Stop the program.

Program:

```
#include<stdio.h>
void  main()
{
        int n;
        printf("Enter a number:");
        scanf("%d",&n);
        if(n<0)
                printf("Number is negative");
        else if(n>0)
                printf("Number is positive");
        else
                printf("Number is equal to zero");
}
```

Output

```
Enter a number:109
Number is positive
Enter a number:-56
Number is negative
Enter a number:0
Number is equal to zero
```

RESULT

Thus a C Program using decision-making constructs was executed and the output was obtained.

2C Program to Check if entered alphabet is vowel or a consonant.

Aim:

To write a C program to check if entered alphabet is vowel or a consonant using switch case.

Algorithm:

1. Start the program
2. Get the character for choice
3.Give the multiple choices using case statement whether one of the choices are in theconsonants and print the same.
4. In the default case print that it is a consonant.
5. Display the result
6. Stop the program.

Program:

```
#include <stdio.h>
int main()
{
char alphabet;
printf("Enter an alphabet:");
scanf("%c",&alphabet);
switch(alphabet)
   {
          case 'a':
                        printf("Alphabet a is a vowel.\n");
                        break;
          case 'e':
                        printf("Alphabet e is a vowel.\n");
                        break;
          case 'i':
                        printf("Alphabet i is a vowel.\n");
                        break;
```

```
        case 'o':
                        printf("Alphabet o is a vowel.\n");
                        break;

        case 'u':
                        printf("Alphabet u is a vowel.\n");
                        break;
        default:
                        printf("You entered a consonant.\n");
   }
return 0;
}
```

Output

```
Enter an alphabet:i
Alphabet i is a vowel.

Enter an alphabet: o
Alphabet o is a vowel.

Enter an alphabet: u
Alphabet u is a vowel.

Enter an alphabet: c
You entered a consonant.
```

Result:

Thus the C program using decision-making construct has been verified and executed successfully.

Ex.No: 3. Write a program to find whether the given year is leap year or Not.

(Hint: not every centurion year is a leap. For example 1700, 1800 and 1900 is not a leap year)

Aim:

To write a C program to find whether the given year is leap year or not using switch case.

Algorithm:

1. Start the program
2. Get the year to find whether it’s a leap year
3. Check if the year % 4 == 0 and year % 100 == 0 and year %400 == 0 for the leap year, else it’s not a leap year
4. Display the result
5. Stop the program.

Program:

```
#include <stdio.h>
int main()
{
        int year;
        printf("Enter a year: ");
        scanf("%d",&year);
        if(year%4 == 0)
        {
                if( year%100 == 0)
                {
                        // year is divisible by 400, hence the year is a leap year
                        if ( year%400 == 0)
                                printf("%d is a leap year.", year);
                        else
                                printf("%d is not a leap year.", year);
                }
                else
                printf("%d is a leap year.", year );
        }
        else
            printf("%d is not a leap year.", year);
        return 0;
}
```

Output

```
Enter a year: 1900
1900 is not a leap year.

Enter a year: 1905
1905 is not a leap year.
```

Result:

Thus the C program using to find whether the given year is leap or not year has been successfully verified and executed.

Ex.No: 4. Write a program to perform the Calculator operations, namely, addition, subtraction, multiplication, division and square of a number.

Aim:

To write a C program to perform calculator operation using the switch case.

Algorithm:

1. Start the program
2. Get the first and second number
3. Choose the choice for addition, subtraction, multiplication, division and square of a number using switch case.
4. If the choice is addition add operation is performed and other operations are performed respectively till square of the number.

5.If the choice is default it shows “invalid operation”.

5. Display the result
6. Stop the program.

Program:

```
#include <stdio.h>
int main()
{
        int num1,num2;
        float result= 0.0 ;
        char ch;    //to store operator choice
        printf("Enter first number: ");
        scanf("%d",&num1);
        printf("Enter second number: ");
        scanf("%d",&num2);
        printf("Choose operation to perform (+,-,*,/,%): ");
        scanf(" %s",&ch);
        result=0;
```

```
    switch(ch)
    {
        case '+':
            result=num1+num2;
            break;
        case '-':
            result=num1-num2;
            break;

        case '*':
            result=num1*num2;
            break;

        case '/':
            result=(float)num1/(float)num2;
            break;

        case '%':
            result=num1%num2;
            break;

        default:
            printf("Invalid operation.\n");
    }

    printf("Result: %d %c %d = %f\n",num1,ch,num2,result);
    return 0;
}
```

Output

```
   First run:
   Enter first number: 10
   Enter second number: 20
   Choose operation to perform (+,-,*,/,%): +
   Result: 10 + 20 = 30.000000

   Second run:
   Enter first number: 10
   Enter second number: 3
   Choose operation to perform (+,-,*,/,%): /
   Result: 10 / 3 = 3.333333

   Third run:
   Enter first number: 10
   Enter second number: 3
   Choose operation to perform (+,-,*,/,%): >
Invalid operation.
   Result: 10 > 3 = 0.000000
```

Result:

Thus the C program using to perform calculator operation has been successfully verified and executed.

Ex.No: 5. Check whether a given number is Armstrong number or not.

Aim:

To write a C program to check whether a given number is Armstrong number or not using switch case.

Algorithm:

```
1. Start the program
2. read number
3. set result=0 and originalnumber=number
4. reminder=originalnumber%10
5. result+=(reminder*reminder*reminder)
6. number=originalnumber/10
7. repeat steps 4 to 6 until number > 0
8. if result = number
9. display number is armstrong
10. else
11. display number is not armstrong
12. stop the program
```

Program:

```
#include <stdio.h>
int main()
{
        int number, originalNumber, remainder, result = 0;
        printf("Enter a three digit integer: ");
        scanf("%d", &number);
        originalNumber = number;
        while (originalNumber != 0)
        {
                remainder = originalNumber%10;
                result += remainder*remainder*remainder;
                originalNumber /= 10;
        }
        if(result == number)
                printf("%d is an Armstrong number.",number);
        else
                printf("%d is not an Armstrong number.",number);
        return 0;
}
```

Output

```
Enter a three digit integer: 371
371 is an Armstrong number.

Enter a three digit integer: 369
369 is an not an Armstrong number.
```

Result:

Thus the C program to check whether a given number is armstrongor not has been successfully verified and executed.

Ex.No: 6. Write a C program to check whether a given number is odd or even.

Aim:

To write a C program to find an odd or even number.

Algorithm:

1. Start the program
2. Get the number
3. Check the number if it is odd or even using if statement.
4. If the number is even check the condition as n%2 ==0 else it is even.
5. Display the result
6. Stop the program.

Program:

```
#include <stdio.h>
int main()
{
        int number;
        printf("Enter an integer: ");
        scanf("%d", &number);
        // True if the number is perfectly divisible by 2
        if(number % 2 == 0)
                printf("%d is even.", number);
        else
                printf("%d is odd.", number);
        return 0;
}
```

Output

```
Enter an integer: -7
-7 is odd.
```

Result:

Thus the C program to find an odd or even number has been successfully verified and executed.

Ex.No:7. Write a program to perform factorial of a number.

Aim:

To write a C program to find a factorial of a given number.

Algorithm:

```
1. Start the program
2. Get the number
3. If the number < 0 print “Error for finding a factorial”
4. Else Initialize variables
      factorial←1
      i←1
5: Read value of n
6: Repeat the steps until i=n
        factorial←factorial*i
        i←i+1
7: Display factorial
8 Stop the program.
```

Program:

```
#include <stdio.h>
int main()
{
        int n, i;
        long factorial = 1;
        printf("Enter an integer: ");
        scanf("%d",&n);
        // show error if the user enters a negative integer
        if (n < 0)
                printf("Error! Factorial of a negative number doesn't exist.");
        else
        {
                for(i=1; i<=n; ++i)
                        factorial *= i;         // factorial = factorial*i;
                printf("Factorial of %d = %lu", n, factorial);
        }
        return 0;
}
```

Output

```
Enter an integer: 10
Factorial of 10 = 3628800
```

Result:

Thus the C program to find the factorial of a given number has been successfully executed and verified.

Ex.No: 8. Write a C program to find out the average height of persons.

Aim:

To write a C program to find average height of persons,

Algorithm:

```
1.Start
2.Declare variables
3.Read the total number of persons and their height.
4.Calculate avg=sum/n and find number of persons their h>avg.
5.Display the output of the calculations .
6.Stop
```

Program

```
//Get a Height of Different Persons and find how many of themare above average

#include <stdio.h>
#include <conio.h>
void main()
{
        int i,n,sum=0,count=0,height[100];
        float avg;
        clrscr();
         //Read Number of persons
        printf("Enter the Number of Persons : ");
        scanf("%d",&n);
         //Read the height of n persons
        printf("\nEnter the Height of each person in centimeter\n");
        for(i=0; i<n; i++)
        {
                scanf("%d",&height[i]);
                sum  = sum + height[i];
         }
        avg = (float)sum/n;
```

```
     //Counting
    for(i=0;i<n;i++)
    {
        if(height[i]>avg)
            count++;
    }
    //display
    printf("\nAverage Height of %d persons is : %.2f\n",n,avg);
    printf("\nThe number of persons above average : %d ",count);
    getch();
}
```

Output

```
Enter the Number of Persons : 5

Enter the Height of each person in centimeter
150
155
162
158
154

Average Height of  5 persons is : 155.8

The number of persons above average : 2
```

Result

Thus a C Program average height of persons was executed and the output was obtained.

Ex.No: 9 Compute the Body Mass Index of the individuals using 2D array.

Aim:

To write a C Program to Populate a two dimensional array with height and weight of persons and compute the Body Mass Index of the individuals..

Algorithm

1.Start
2.Declare variables
3.Read the number of persons and their height and weight.
4.Calculate BMI=W/H2for each person
5.Display the output of the BMI for each person.
6.Stop

PROGRAM

```
#include<stdio.h>
#include<math.h>
int main(void)
{
        int n,i,j;
        float massheight[3][2];
        float bmi[3];
        printf("How many people's BMI do you want to calculate?\n");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                for(j=0;j<2;j++)
                {
                        switch(j)
                        {
                        case 0:
                                printf("\nPlease enter the mass of the person %d in kg: ",i+1);
                                scanf("%f",&massheight[i][0]);
                                break;
```

```
                    case 1:
                              printf("\nPlease enter the height of the person %d in meter: ",i+1);
                              scanf("%f",&massheight[i][1]);
                              break;
                    }
          }
     }
     for(i=0;i<n;i++)
     {
          bmi[i]=massheight[i][0]/pow(massheight[i][1],2.0);
          printf("Person %d's BMI is %f\n",i+1,bmi[i]);
     }
     return 0;
}
```

OUTPUT

```
How many people's BMI do you want to calculate?
2
Please enter the mass of the person 1 in kg: 88
Please enter the height of the person 1 in meter: 1.8288

Please enter the mass of the person 2 in kg:58
Please enter the height of the person 2 in meter: 2.2

Person 1's BMI is26.31178
Person 2's BMI is11.98347
```

Result

Thus a C Program Body Mass Index of the individuals was executed and the output was obtained.

Ex.No:10. **Write a C program to perform swapping using function.**

Aim:

To write a C program to perform swapping using function.

Algorithm:

```
1. Start the program
2. Declare and get the two integer variablesa and b.
3. call the swap () function
    3.1 In swap definition use the temporary variable and assign temp =a
    3.2 a=b
    3.3 b=temp
4. Print the a and b value.
5. Display the result
6. Stop the program.
```

Program:

```
#include<stdio.h>
#include<conio.h>
void swap(int,int);
void main()
{
        int a,b,r;
        clrscr();
        printf("enter value for a&b: ");
        scanf("%d%d",&a,&b);
        swap(a,b);
        getch();
}
```

```
void swap(int a,int b)
{
        int temp;
        temp=a;
        a=b;
        b=temp;
        printf("after swapping the value for a & b is : %d %d",a,b);
}
```

Output:

```
enter value for a&b: 4
5
after swapping the value for a &b : 5 4
```

Result:

Thus the C program to perform swapping using function has been successfully executed and verified.

}

Ex.No: 11. Write a C program to display all prime numbers between two intervals using functions.

Aim:

To write a C program to display all prime numbers between two intervals using functions.

Algorithm:

```
1. Start the program
2. Enter the lower and upper limit of prime number list.
3. CallprintPrimes(lowerLimit, upperLimit);
    3.1 Check while (lowerLimt<=upperLimt) do
    3.2 Increase the lowerlimit and check using the while loop  If the number is divisible by 1
    and self then it is prime
    3.3 Repeat the step until the upper limit is reached
4. Print the prime numbers in the interval.
5. Display the result
6. Stop the program.
```

Program:

```
#include <stdio.h>
/* Function declarations */
int isPrime(int num);
void printPrimes(int lowerLimit, int upperLimit);
int main()
{
        int lowerLimit, upperLimit;
        printf("Enter the lower and upper limit to list primes: ");
        scanf("%d%d", &lowerLimit, &upperLimit);
        // Call function to print all primes between the given range.
        printPrimes(lowerLimit, upperLimit);
        return 0;
}
```

```
void printPrimes(int lowerLimit, int upperLimit)
{
        printf("All prime number between %d to %d are: ", lowerLimit, upperLimit);
        while(lowerLimit<= upperLimit)
        {
                // Print if current number is prime.
                if(isPrime(lowerLimit))
                {
                        printf("%d ", lowerLimit);
                }
                lowerLimit++;
        }
}
int isPrime(int num)
{
        int i;
        for(i=2; i<=num/2; i++)
        {
                if(num % i == 0)
                {
                        return 0;
                }
        }
        return 1;
}
```

Output

```
Enter the lower and upper limit to list primes 20 50
Prime numbers between 20 and 50 are: 23 29 31 37 41 43 47
```

Result:

Thus the C program to display all prime numbers between two intervals using functions has been successfully executed and verified.

Ex.No:12. **Write a C program to reverse a sentence using recursion.**

Aim:

To write a C program to reverse a sentence using recursion

Algorithm:

1. Start the program
2. Read the sentence
3. Declare the reverse sentence () function.
4. Read the sentence reversely until newline is reached as (if c!=’\n’)
5. Display the result
6. Stop the program

Program:

```
#include <stdio.h>
voidreverseSentence();
int main()
{
        printf("Enter a sentence: ");
        reverseSentence();
        return 0;
}

voidreverseSentence()
{
        char c;
        scanf("%c", &c);
        if( c != '\n')
        {
                reverseSentence();
                printf("%c",c);
        }
}
```

Output:

```
Enter a sentence: margorpemosewa
awesome program
```

Result:

Thus the C program to reverse a sentence using recursion has been successfully executed and verified.

Ex.No: 13. Write a program in C to get the largest element of an array using function.

Aim:

To write a C program to get the largest element of an array.

Algorithm:

1. Start the program
2. Read the number of array elements as size.
3. Read array elements array i= 0,1,2,3,…n-1
4. Assume first element array[0] to be maximum
5. Compare each element array[i] with maximum
6. If(maximum> array[i]) then maximum = array[i]
7. Display the result.
8. Stop the program.

Program:

```
int main()
{
        int array[100], maximum, size, c, location = 1;

        printf("Enter the number of elements in array\n");
        scanf("%d", &size);


        printf("Enter %d integers\n", size);
        for (c = 0; c < size; c++)
                scanf("%d", &array[c]);


        maximum = array[0];

        for (c = 1; c < size; c++)
        {
                if (array[c] > maximum)
                {
                        maximum  = array[c];
                        location = c+1;
                }
        }
        printf("Maximum element is present at location %d and it's value is %d.\n", location, maximum);
        return 0;
}
```

Output:

```
Enter the number of elements in array
5
Enter 5 integers
4
5
6
8
2
Maximum element present at location 4 and its value is 8
```

Result:

Thus the C program to get the largest element of an array using the function has been successfully executed and verified.

Ex.No:14. Write a C program to concatenate two strings.

Aim:

To write a C program to concatenate two strings.

Algorithm:

1. Start the program
2. Read the strings
3. Use the for loop to get the first sentence and use the second for loop for second sentence and increment the counter i.

4. Now assign the sentence s2[j] to s1[i] until the string reaches the condition s2[j]!=0 and the sentences are concatenated

5.Display the result

6.Stop the program

Program:

```
#include <stdio.h>
int main()
{
        char s1[100], s2[100], i, j;
        printf("Enter first string: ");
        scanf("%s", s1);
        printf("Enter second string: ");
        scanf("%s", s2);
        // calculate the length of string s1and store it in i
        for(i = 0; s1[i] != '\0'; ++i);
        for(j = 0; s2[j] != '\0'; ++j, ++i)
        {
                s1[i] = s2[j];
        }
        s1[i] = '\0';
        printf("After concatenation: %s", s1);
        return 0;
}
```

Output

```
Enter first string: hi
Enter second string:welcome
After concatenation: Hi welcome
```

Result:

Thus the C program to concatenate two strings has been successfully executed and verified.

Ex.No: 15. Write a C program to find the length of String.

Aim:

To write a C program to find the length of the string.

Algorithm:

1. Start the program
2. Read the sentence
3. Use the for loop and test the condition that s[i] != '\0'
4. Use the i value to display the length
5. Stop the program

Program:

```
 #include <stdio.h>
int main()
{
        char s[1000];
        inti;
        printf("Enter a string: ");
        scanf("%s", s);
        for(i = 0; s[i] != '\0'; ++i);
        printf("Length of string: %d", i);
        return 0;
}
```

Output:

```
Enter a string: Program
Length of string: 7
```

Result:

Thus the C program to find the length of String has been successfully executed and verified.

Ex.No: 16. Find the frequency of a character in a string.

Aim:

To write a C program to find the frequency of a character in the string.

Algorithm:

1. Start the program.
2. Read the sentence
3. Calculate the frequency of the character repeated in the string.
4. Read each character of string to find its frequency
5. Display the result
6. Stop the program

Program:

```
#include <stdio.h>
int main()
{
        charstr[1000], ch;
        inti, frequency = 0;
        printf("Enter a string: ");
        gets(str);
        printf("Enter a character to find the frequency: ");
        scanf("%c",&ch);
        for(i = 0; str[i] != '\0'; ++i)
        {
                if(ch == str[i])
                        ++frequency;
        }
        printf("Frequency of %c = %d", ch, frequency);
        return 0;
   }
```

Output

```
Enter a string: This website is awesome.
Enter a character to find the frequency: e
Frequency of e = 4
```

Result:

Thus the C program to find the frequency of a character in a string has been successfully executed and verified.

Ex.No: 17. Write a C program to Store Student Information in Structure and Display it.

Aim:

To write a C program to store the student information using structure.

Algorithm:

1. Start the program
2. Declare the variables in the structure data type.
3. Read the values of the variables in the structure.
4. Display the result
5. Stop the program

Program:

```
#include <stdio.h>
struct Student
{
        char name[50];
        int roll;
        float marks;
} s[10];
int main()
{
        inti;
        printf("Enter information of students:\n");
        // storing information
        for(i=0; i<10; ++i)
        {
                s[i].roll = i+1;
                printf("\nFor roll number%d,\n",s[i].roll);
                printf("Enter name: ");
                scanf("%s",s[i].name);
                printf("Enter marks: ");
                scanf("%f",&s[i].marks);
                printf("\n");
        }
        printf("Displaying Information:\n\n");
        // displaying information
        for(i=0; i<10; ++i)
        {
                printf("\nRoll number: %d\n",i+1);
                printf("Name: ");
                puts(s[i].name);
                printf("Marks: %.1f",s[i].marks);
                printf("\n");
        }
        return 0;
}
```

Output:

```
Enter information of students:

For roll number1,
Enter name: Tom
Enter marks: 98

For roll number2,
Enter name: Jerry
Enter marks: 89
..
Displaying Information:

Roll number: 1
Name: Tom
Marks: 98
.
For roll number2,
Enter name: Jerry
Enter marks: 89
```

Result:

Thus the C program to store Student Information in Structure has been successfully executed and verified.

Ex.No: 18. The annual examination is conducted for 10 students for five subjects. Write a program to read the data and determine the following:

(a) Total marks obtained by each student.
(b) The highest marks in each subject and the marks of the student who secured it.
(c) The student who obtained the highest total marks.

Aim:

To write a C program to get various details regarding the marks obtained by the students.

Algorithm:

1. Start the program
2. Read the inputs of the student details.
3. Declare the structure and its size along with the variables.
4. Read the values for the variables.
5. Display the result
6. Stop the program

Program:

```
#include<stdio.h>
#define SIZE 50

struct Student
{
        char name[30];
        introllno;
        int sub[3];
};

void main()
{
        inti, j, max, count, total, n, a[SIZE], ni;
        struct Student st[SIZE];
        clrscr();

        printf("Enter how many students: ");
        scanf("%d", &n);
```

```
// for loop to read the names and roll numbers
for (i = 0; i< n; i++)
{
        printf("\nEnter name and roll number for student %d : ", i);
        scanf("%s", &st[i].name);
        scanf("%d", &st[i].rollno);
}

// for loop to read ith student's jth subject
for (i = 0; i< n; i++)
{
        for (j = 0; j <= 2; j++)
        {
                printf("\nEnter marks of student %d for subject %d : ", i, j);
                scanf("%d", &st[i].sub[j]);
        }
}

// (i) for loop to calculate total marks obtained by each student

for (i = 0; i< n; i++)
{
        total = 0;
        for (j = 0; j < 3; j++)
        {
                 total = total + st[i].sub[j];
        }
        printf("\nTotal marks obtained by student %s are %dn", st[i].name,total);
        a[i] = total;
}

/* (ii) for loop to list out the student's roll numbers whohave secured the highest marks in
each subject */

//  roll number who secured the highest marks
```

```
        for (j = 0; j < 3; j++)
        {
                max = 0;
                for (i = 0; i< n; i++)
                {
                        if (max <st[i].sub[j])
                        {
                                max = st[i].sub[j];
                                ni = i;
                        }
                }
        printf("\nStudent %s got maximum marks = %d in Subject : %d",st[ni].name, max, j);
        }

        max = 0;
        for (i = 0; i< n; i++)
        {
                if (max < a[i])
                {
                        max = a[i];
                        ni = i;
                }
        }
        printf("\n%s obtained the total highest marks.", st[ni].name);
        getch();
}
```

Output:

```
Enter how many students: 2
Enter name and roll number for student 0 :Pritesh 1
Enter name and roll number for student 1 :Suraj 2
Enter marks of student 0 for subject 0 : 90
Enter marks of student 0 for subject 1 : 89
Enter marks of student 0 for subject 2 : 78
Enter marks of student 1 for subject 0 : 67
Enter marks of student 1 for subject 1 : 88
Enter marks of student 1 for subject 2 : 99

Total marks obtained by student Pritesh are 257
Total marks obtained by student Suraj are 254

Student Pritesh got maximum marks = 90 in Subject : 0
Student Pritesh got maximum marks = 89 in Subject : 1
Student Suraj got maximum marks = 99 in Subject : 2

Pritesh obtained the total highest marks.
```

Result:

Thus the C program to get various details regarding the marks obtained by the students has been successfully executed and verified.

EX.No. : 19 Railway reservation system

Aim

Create a Railway reservation system in C with the following modules

Booking

Availability checking

Cancellation

Prepare chart

.

Algorithm

1.Start
2.Declare variables
3.Display the menu options
4.Read the option.
5.Develop the code for each option.
6.Display the output of the selected option based on existence .
7.Stop

PROGRAM

```
#include<stdio.h>

#include<conio.h>

int first=5,second=5,third=5;

struct node

{

        int ticketno;

        int phoneno;

        char name[100];

        char address[100];

}s[15];


int i=0;
void booking()

{

        printf("enter your details");
        printf("\nname:");
        scanf("%s",s[i].name);
        printf("\nphonenumber:");
        scanf("%d",&s[i].phoneno);
        printf("\naddress:");
        scanf("%s",s[i].address);
        printf("\nticketnumber only 1-10:");
        scanf("%d",&s[i].ticketno);
        i++;
}

void availability()
{
        int c;
        printf("availability cheking");
        printf("\n1.first class\n2.second class\n3.third class\n");
        printf("enter the option");
        scanf("%d",&c);
        switch(c)
        {
                case 1: if(first>0)
                        {
                                printf("seat available\n");
                                first--;
                        }
                        else
                        {
                                printf("seat not available");
                        }
                        break;

                case 2: if(second>0)
                        {
                                printf("seat available\n");
                                second--;
                        }
                        else
                        {
                                printf("seat not available");
                        }
                        break;

                case 3: if(third>0)
                        {
                                printf("seat available\n");
                                third--;
                        }
                        else
                        {
                                printf("seat not available");
                        }
                        break;
                default:
                        break;
        }
}

void cancel()
{
        int c;
        printf("cancel\n");
        printf("which class you want to cancel");
        printf("\n1.first class\n2.second class\n3.third class\n");
        printf("enter the option");
        scanf("%d",c);
        switch(c)
        {
                case 1:
                        first++;
                        break;
                case 2:
                        second++;
                        break;
                case 3:
                        third++;
                        break;
                default:
                        break;
        }
        printf("ticket is canceled");
}

void chart()
{
        int  c;
        for(c=0;c<I;c++)
        {
                printf(“\n Ticket No\t Name\n”);
                printf(“%d\t%s\n”,s[c].ticketno,s[c].name)
        }
}

void main()
{
        int n;
        clrscr();
        printf("welcome to railway ticket reservation\n");
        while(1)
        {
                printf("1.booking\n2.availability cheking\n3.cancel\n4.Chart \n5. Exit\nenter your
                option:");
                scanf("%d",&n);
                switch(n)
                {
                        case 1: booking();
                            break;
                        case 2: availability();
                            break;
                        case 3: cancel();
                            break;
                        case 4: chart();
                            break;
                        case 5:
                            printf("\n Thank you visit again!");
                            getch();
                            exit(0);
                        default:
                            break;
                }
        }
        getch();
}
```

Output

```
welcome to railway ticket reservation
1.booking
2.availabilitycheking
3.cancel
4.Chart
5. Exit
```

```
enter your option: 2
availabilitycheking
1.first class
2.second class
3.third class

enter the option 1
seat available
1.booking
2.availabilitycheking
3.cancel
4.chart
5. Exit
enter your option: 5
Thank you visit again!
```

Result

Thus a C Program for Railway reservation system was executed and the output was obtained.

Additional C Programs for exercise

1. To write a C program for temperature conversion from Celsius to Fahrenheit and vice versa.

```
#include<stdio.h>

main()

{

        float cel, fah ,c ,f;

        clrscr();

        printf("\nEnter the fahrenheit value:");

        scanf("%f",&f);

        cel=(5.0/9.0)*(f-32);

        printf("Celsius=%d",cel);

        printf("\nEnter the Celsius value:");

        scanf("%f",&c);

        fah=(9.0/5.0)*c+32;

        printf("Fahrenheit=%d",fah);

        getch();

}
```

2. C Program to print the sine series.

```
#include<stdio.h>
 #include<math.h>
#include<conio.h>
void main()
{
int i,n;
float x,y,z,sum,t;
clrscr();
printf("\t\t PROGRAM TO PRINT SINE SERIES\t\t\n"); printf("\t\t t\t\n");
printf("\n ENTER THE ANGLE:");
scanf("%f", &x);
printf("\n ENTER THE NUMBER OF TERMS:");
scanf("%d", &n);

z=x; x=x*(3.14/180);
sum=x; t=x;
for(i=2; i<=n; i=i+2)
{
t=t*(-x*x)/((2*i-1)*(2*i-2)); sum=sum+t;
}
printf("\n\n THE VALUE OF SIN(%5.2f) IS %5.2f\n", z, sum); getch();
}
```

3. Program to print current system date.

```
#include <stdio.h>

#include <conio.h>

#include <dos.h>

int main()

{

struct date d;

getdate(&d);

printf("Current system date is %d/%d/%d",d.da_day,d.da_mon,d.da_year);

getch();

return 0;

}.
```

4.Program to calculate Standard Deviation.

```
#include <stdio.h>
#include <math.h>
floatstandard_deviation(float data[], int n);
int main()
{
int n, i;
float data[100];
printf("Enter number of datas( should be less than 100): ");
scanf("%d",&n);
printf("Enter elements: ");
for(i=0; i<n; ++i)
scanf("%f",&data[i]);
printf("\n");
printf("Standard Deviation = %.2f", standard_deviation(data,n));
return 0;
}
float standard_deviation(float data[], int n)
{
float mean=0.0, sum_deviation=0.0;
int i;
for(i=0; i<n;++i)
{
mean+=data[i];
}
mean=mean/n;
for(i=0; i<n;++i)
sum_deviation+=(data[i]-mean)*(data[i]-mean);
returnsqrt(sum_deviation/n);
}
```

5. Program to calculate the Power of a Number using Recursion.

```
#include <stdio.h>
int power(int n1,int n2);
int main()
{
int base, exp;
printf("Enter base number: ");
scanf("%d",&base);
printf("Enter power number(positive integer): ");
scanf("%d",&exp);
printf("%d^%d = %d", base, exp, power(base, exp));
return 0;
}
int power(intbase,intexp)
{
if ( exp!=1 )
return (base*power(base,exp-1));
}
```

6. Program to find the ASCII value of a Character.

```
#include <stdio.h>
int main(){
char c;
printf("Enter a character: ");
scanf("%c",&c); /* Takes a character from user */
printf("ASCII value of %c = %d",c,c);
return 0;
}
```

7. Program to find biggest of four no by using ternary numbers.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
inta,b,c,d,big;
clrscr( );
printf(“enter value a”);
scanf(“%d”,&a);
printf(“enter the value of b”);
scanf(“%d”,&b);
printf(“enter the value of c”);
scanf(“%d”,&c);
printf(“enter the value of d”);
scanf(“%d”,&d);
big=(a>b)?(a>c)?(a>d)?a:d:(c>d)?c:d:(b>c)?(b>d)?b:d:(c>d)?c:d;
printf(“Biggest of the given 4 numbers is %d”,big);
getch();
}
```

8. **Matrix Multiplication.**

```
#include <stdio.h>

int main()
{
   int m, n, p, q, c, d, k, sum = 0;
   int first[10][10], second[10][10], multiply[10][10];

   printf("\nEnter the number of rows and columns of first matrix:\n");
   scanf("%d%d", &m, &n);

   /*//Entering elements of first matrix
   printf("\n Enter the elements of first matrix\n");
   for (  c = 0 ; c < m ; c++ )
      for ( d = 0 ; d < n ; d++ )
         scanf("%d", &first[c][d]);*/

   printf("\nEnter the number of rows and columns of second matrix:\n");
   scanf("%d%d", &p, &q);

   //Checking if Matrix Multiplication is possible
   if ( n != p )
   {
      printf("\nMatrices with entered orders can't be multiplied with each other.\n");
      printf("\nThe column of first matrix should be equal to row of second.\n");
   }
   else
   {
      //Entering elements of first matrix
      printf("\nEnter the elements of first matrix:\n");
      for (  c = 0 ; c < m ; c++ )
         for ( d = 0 ; d < n ; d++ )
            scanf("%d", &first[c][d]);

      //Entering elements of second matrix
      printf("\nEnter the elements of second matrix:\n");
      for ( c = 0 ; c < p ; c++ )
         for ( d = 0 ; d < q ; d++ )
            scanf("%d", &second[c][d]);

      //Carrying out matrix multiplication operation
      for ( c = 0 ; c < m ; c++ )
      {
         for ( d = 0 ; d < q ; d++ )
         {
            for ( k = 0 ; k < p ; k++ )
            {
```

```
          sum = sum + first[c][k]*second[k][d];
        }

        multiply[c][d] = sum;
        sum = 0;
      }
    }

    //Printing the final product matrix
    printf("\nThe product of entered matrices is:\n");
    for ( c = 0 ; c < m ; c++ )
    {
      for ( d = 0 ; d < q ; d++ )
        printf("%d\t", multiply[c][d]);

      printf("\n");
    }
  }

  return 0;
}
```

9.C Program to reverse the digits of a number.

```
#include<stdio.h>
 #include<conio.h>
void main()
{
int n, reverse = 0; clrscr();
printf("REVERSAL OF A GIVEN NUMBER\n"); printf("      \n");
printf("ENTER A NUMBER TO REVERSE: \n");
scanf("%d",&n); while (n != 0)
{
reverse = reverse * 10; reverse = reverse + n%10; n = n/10;
}
printf("REVERSE OF ENTERED NUMBER IS: %d\n", reverse);
getch();
}
```

C Language Questions and Answers Viva - Voce

1. What is C language?

C is a programming language developed at AT&T's Bell Laboratories of USA in 1972. The Cprogramming language is a standardized programming language developed in the early 1970s by KenThompson and Dennis Ritchie for use on the UNIX operating system. It has since spread to manyother operating systems, and is one of the most widely used programming languages.

2. What is an array?

Array is a variable that hold multiple elements which has the same data type.

3. What is the purpose of main() function?

The function main() invokes other functions within it. It is the first function to be called whenthe program starts execution.

☐It is the starting function.

☐It returns an int value to the environment that called the program.

☐ Recursive call is allowed for main( ) also.

☐It is a user-defined function.

4. What are the different storage classes in C?

There are four types of storage classes in C language.

☐ Automatic

☐ Extern

☐ Register

☐ Static

5. What is a structure?

Structure constitutes a super data type which represents several different data types in a

single unit. A structure can be initialized if it is static or global.

6. What is the difference between #include<> and #include " "?

#include<> ----> specifically used for built in header files.

#include " " ----> Specifically used for used for user defined/created header file.

7. What are the advantages of the functions?

☐It reduces the Complexity in a program by reducing the code.

☐ Functions are easily understanding and reliability and execution is faster.

☐It also reduces the Time to run a program. In other way, it's directly proportional to

Complexity.

☐It's easy to find-out the errors due to the blocks made as function definition outside the mainfunction.

8. How do declare an array?

We can declare an array by specify its data type, name and the number of elements the arrayholds between square brackets immediately following the array name.

syntax :

data_typearray_name[size];

9. What are the differences between structures and union?

A structure variable contains each of the named members, and its size is large enough to

hold all the members. Structure elements are of same size.

A Union contains one of the named members at a given time and is large enough to hold

thelargest member. Union element can be of different sizes.

10. What is the use of typedef?

The typedef help in easier modification when the programs are ported to another machine. Adescriptive new name given to the existing data type may be easier to understand the code.

11. What is recursion?

A recursion function is one which calls itself either directly or indirectly it must halt at a definite point to avoid infinite recursion.

12. What are the characteristics of arrays in C?

□An array holds elements that have the same data type.

□ Array elements are stored in subsequent memory locations

□ Two-dimensional array elements are stored row by row in subsequent memory locations.

□ Array name represents the address of the starting element

13. Differentiate between for loop and a while loop? What are it uses?

For executing a set of statements fixed number of times we use for loop while when the number of iterations to be performed is not known in advance we use while loop.

14. What are register variables? What are the advantages of using register variables?

If a variable is declared with a register storage class, it is known as register variable. Theregister variable is stored in the CPU register instead of main memory. Frequently used variables aredeclared as register variable as it’s access time is faster.

15. What the advantages of using Unions?

When the C compiler is allocating memory for unions it will always reserve enough room forthe largest member.

16. What is a function?

A large program is subdivided into a number of smaller programs or subprograms. Each subprogram specifies one or more actions to be performed for the larger program. Such subprograms are called functions.

17. What is an argument?

An argument is an entity used to pass data from the calling to a called function.

18. What is the difference between syntax vs logical error?

Syntax Error

☐These involve validation of syntax of language.

☐ Compiler prints diagnostic message.

Logical Error

☐ Logical error are caused by an incorrect algorithm or by a statement mistyped in such a way that it doesn’t violet syntax of language.

☐ Difficult to find.

19. Explain enumerated types.

☐ Enumerated types allow the programmers to use more meaningful words as values to a variable.

☐Each item in the enumerated type variable is actually associated with a numeric code.

20. Differentiate between the expression “++a” and “a++”?

☐With ++a, the increment happens first on variable a, and the resulting value is used. Thisis called as prefix increment.

☐With a++, the current value of the variable will be used in an operation. This is called as postfix increment.

21.What will happen when you access the array more than its dimension?

If the index of the array size is exceeded, the program will crash. But the modern compilers will take care of this kind of errors.

22.What is the use of ‘\0’ and ‘%s’?

‘\0’ is the null character. Every string literal constant is automatically terminated by ‘\0’. The number of bytes required to store a string literal constant is one more than the number of characters present in it. The additional byte is required for storing ‘\0’.

‘%s’ is the format specifier used in scanf function that reads all the characters up to, but not including the white-space character. Thus, scanf function with ‘%s’ specifier can be used to read single word strings but cannot be used to read multi-word strings.

23.Specify any two applications of Array.

•Arrays are used to implement mathematical vectors and matrices, as well as other kinds of rectangular tables.

•Arrays are used to implement other data structures, such as lists, heaps, hash tables, queues and stacks.

24.What is a nested structure?
A structure can be nested within another structure.

25.Is it mandatory that the size of all elements in a union should be same?
No.The standard only guarantee that the size of a union is sufficient for the largest member, i.e, not necessarily the same size.

****