JULY 2021

PPSC MODULE 3 SOLUTIONS

CIE 1 & 2 TYPED NOTES

SATISH | UJJWAL | VISHNU | NISHANT

C Programming QB Module 3 Part A Solutions for CIE-1

@ Ujjwal Acharya

1. Define what an array is and write the syntax to declare an array.

> **Arrays** - a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more recognized as a collection of variables of the same type.
> Syntax: data_type array_name[array_size];
> Example: int a[5];

2. Find the output of the following code.

```
#include <stdio.h>
void main()
{
int a[3][2] = {10, 20, 30, 40, 50, 60};
printf("%d", a[2][2]);
}
//Output: 0
```

3. Find the output of the following code.

```
#include <stdio.h>
void main()
{
int a[3][2] = {10, 20, 30, 40, 50, 60};
printf("%d", a[0][4]);
}
//Output: 50
```

4. Find the output of the following code.

```
#include <stdio.h>
#include <string.h>
void main() {
char s1[] = "hellow";
char s2[ ] ="helow";
int x;
x = strncmp(s1,s2,3);
printf("x = %d", x);
}
//Output: x=0
```

5. Find the output of the following code.

```
#include <stdio.h>
#include <string.h>
void main() {
char s1[] = "NEW DELHI";
char s2[] ="BANGALORE";
strncpy(s1,s2,4);
printf("%s", s1);
}
//Output: BANGDELHI
```

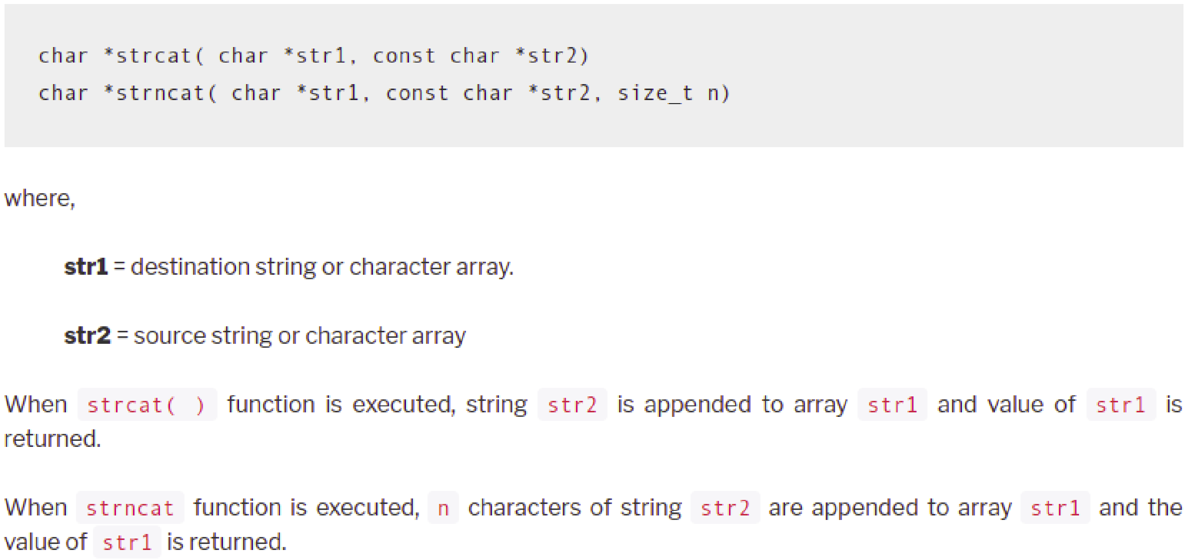
6. Find the output of the following code.

```
#include <stdio.h>
#include <string.h>
void main() {
char s1[] = "NEW DELHI";
char s2[] ="NEW";
printf("%d",strstr(s1,s2));
}
//Output: Error as the print statement is made to print in integers whereas the data type is character
```

7. Find the output of the following code.

```
#include <stdio.h>
void main() {
int a[4][3];
printf("%d",sizeof(a));
}
//Output: 48
```

8. Compare the string handling functions strcat() and strncat().



```
char *strcat( char *str1, const char *str2)
char *strncat( char *str1, const char *str2, size_t n)
```

where,

str1 = destination string or character array.

str2 = source string or character array

When `strcat( )` function is executed, string `str2` is appended to array `str1` and value of `str1` is returned.

When `strncat` function is executed, `n` characters of string `str2` are appended to array `str1` and the value of `str1` is returned.

9. Find the output of the following code.

```
#include <stdio.h>
void main() {
int i, j, a[][3]= {{1,2,3}, {4,5,6}};
for(i=0; i< 2; i++) {
for(j=0; j < 3;j++)
printf("%5d", a[i][j]);
printf("\n"); }
}
//Output: 1    2    3
//        4    5    6
```

10. Explain the following functions string handling functions. (i) strcmp() (ii). strrev()

> **1. strcmp()**
> — This string function is basically used for the purpose of comparing two strings.
> — This string function compares two strings character by characters.
> — Thus it gives result in three cases:
> Case 1: If the first string > than the second string then, the result will be true.
> Case 2: if first string < than second string then, the result will be false.

> Case 3: If the first string = = to the second string then, the result will be zero.
>
> Example:
>
> ```
> char str1= —Gaurav‖;
> char str2= —Arora‖;
> char str3=strcmp(str1,str2);
> printf(—%s‖,str3);
> ```
>
> **2. strrev()**
>
> — This string function is basically used for the purpose of reversing the string.
>
> Example:
>
> ```
> char str1= —Gaurav;
> char str2[20];
> str2= strrev(str2,str1);
> printf(—%s‖,str2);
> ```

C Programming QB Module 3 Part B Solutions for CIE-1

@ Nishant, Vishnu and Ujjwal

1. Write C program to find the sum of given list of integers.

```
#include <stdio.h>

int main(){

int n,i,sum=0;
printf("Enter the number of integers: ");
scanf("%d",&n);
int arr[n];

for(i = 0; i<n; i++){

scanf("%d",&arr[i]);
sum += arr[i];
}
printf("%The sum of integers is : %d",sum);
return 0;

}
```

2. Write C program to find the largest and smallest number among a list of integers.

```
#include <stdio.h>

int main(){

int n,i;
printf("Enter the number of integers: ");
scanf("%d",&n);
int arr[n];
for(i = 0; i<n; i++)
    scanf("%d",&arr[i]);

int max = arr[0], min = arr[0];

for(i=0;i<n;i++){
    if(arr[i]>max)
        max = arr[i];
    if(arr[i]<min)
        min = arr[i];
}

printf("%The max of integers is : %d and min is: %d",max,min);
return 0;

}
```

3. Write C program to read a list of elements into an array and print the reverse of the list.

```
#include <stdio.h>

int main(){

int n,i;

printf("Enter number of elements: ");
scanf("%d",&n);

int a[n];

for(i=0;i<n;i++)
    scanf("%d",&a[i]);

printf("The reversed array is: ");
for(i=n-1;i>=0;i--)
    printf("%d ",a[i]);

return 0;
}
```

4. Write C program to read two matrices and find the addition and multiplication of two matrices.

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

```c
int matrix_colle(int rows,int columns,int matrix[rows][columns]){
    for (int i = 0; i < rows; i++){
        for (int j = 0; j < columns; j++){
            printf("Enter values one by one:\n");
            scanf("%d", &matrix[i][j]);
        }
    }
    return matrix;
}

int main(){
    int m,n;
    printf("Enter number of rows and columns:\n");
    scanf("%d %d", &m, &n);
    int mat_1[m][n], mat_2[m][n], addi[m][n], mul[m][n];

    printf("Enter values for 1st matrix\n");
    matrix_colle(m, n, mat_1);

    printf("Enter values for 2nd matrix\n");
    matrix_colle(m, n, mat_2);

    for (int i = 0; i < m; i++){
        for (int j = 0; j < n; j++){
            addi[i][j] = mat_1[i][j] + mat_2[i][j];
        }
    }
    printf("The sum of both the matrices is:\n");
    for (int i = 0; i < m; i++){
        for (int j =0; j < n; j++){
            printf("%d   ", addi[i][j]);
        }
        printf("\n");
    }

    //Matrix Multiplication logic
    for (int i = 0; i < m; i++){
        for (int j = 0; j < n; j++){
            mul[i][j] = 0;
            for (int k = 0; k < n; k++){
                mul[i][j] += mat_1[i][k] * mat_2[k][j];
            }
        }
    }
    printf("The multiplication of the matrices is:\n");
    for (int i = 0; i < m; i++){
        for (int j = 0; j < n; j++){
            printf("%d   ", mul[i][j]);
        }
        printf("\n");
    }
}
```

5. Write C program to find the transpose of a matrix.

```
#include <stdio.h>
int main() {
  int a[10][10], transpose[10][10], r, c;
  printf("Enter rows and columns: ");
  scanf("%d %d", &r, &c);

  // asssigning elements to the matrix
  printf("\nEnter matrix elements:\n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("Enter element a%d%d: ", i + 1, j + 1);
    scanf("%d", &a[i][j]);
  }

  // printing the matrix a[][]
  printf("\nEntered matrix: \n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("%d  ", a[i][j]);
    if (j == c - 1)
    printf("\n");
  }

  // computing the transpose
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    transpose[j][i] = a[i][j];
  }

  // printing the transpose
  printf("\nTranspose of the matrix:\n");
  for (int i = 0; i < c; ++i)
  for (int j = 0; j < r; ++j) {
    printf("%d  ", transpose[i][j]);
    if (j == r - 1)
    printf("\n");
  }
  return 0;
}
```

6. Write a C program to store numbers into an array and find the frequency of a particular number in array and print it.

```
#include <stdio.h>

int main(){

int n,i;
printf("Enter number of elements: ");

scanf("%d",&n);

int arr[n];

for(i=0;i<n;i++)
    scanf("%d",&arr[i]);

int number, frequency = 0;
printf("Enter a number: ");

scanf("%d",&number);

for(i=0;i<n;i++)
{
    if(number == arr[i]){
        frequency ++;
    }
}

printf("The frequency of %d in array is %d",number,frequency);

return 0;

}
```

7. Write a C program to copy the string str2 into str1 without using strcpy() function.

```
#include <stdio.h>

int main(){
int i;
char str1[100];
char str2[100];
printf("Enter your String: ");
gets(str1);

for(i=0;str1[i]!='\0';i++){
    str2[i] = str1[i];
}
printf("The copied String is: %s",str2);
return 0;

}
```

8. Write a C program to check whether a string is palindrome or not without using string function.

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

int main()
{
    char s[1000];
    int i,n,c=0;

    printf("Enter  the string : ");
    gets(s);
    n=strlen(s);

    for(i=0;i<n/2;i++)
    {
      if(s[i]==s[n-i-1])
      c++;

  }
  if(c==i)
      printf("string is palindrome");
    else
        printf("string is not palindrome");



    return 0;
}
```

9. Write a C program to read your email id and print the number of vowels, consonants and special characters in it.

```
#include <stdio.h>
int main()
{
    char str[100];
    char *p;
    int  vCount=0,cCount=0,specChar=0;

    printf("Enter any string: ");
    fgets(str, 100, stdin);

    //assign base address of char array to pointer
    p=str;

    //'\0' signifies end of the string
    while(*p!='\0')
    {
        if(*p=='A' ||*p=='E' ||*p=='I' ||*p=='O' ||*p=='U'||*p=='a' ||*p=='e' ||*p=
='i' ||*p=='o' ||*p=='u')
            vCount++;
        if(*p=='@' || *p=='.')
            specChar++;
        else
            cCount++;
        //increase the pointer, to point next character
        p++;
    }

    printf("Number of Vowels in String: %d\n",vCount);
    printf("Number of Consonants in String: %d",cCount);
    printf("Number of Special Charactes in String: %d",specChar);
    return 0;
}
```

10. Write a C program to insert a sub-string in to given main string at a given position without using string functions.

```
#include <stdio.h>
int main(){
char str1[100];
char str2[50];
char str_final[200];
int i,j,k=0,pos;

puts("enter the first string");
gets(str1);
puts("Enter Second String:");
gets(str2);
printf("Enter the position to insert");
scanf("%d",&pos);
// Print both strings ///
printf("First string: %s \n",&str1);
printf("Second string : %s \n",&str2);

// Start first loop //

for(i=0;i<pos;i++){
 str_final[i]=str1[i];
printf("i=%d => %c \n",i,str_final[i]);
}
k=i; // Position from where final string will start
printf("\n first loop is over --\n");

for(j=0;str2[j] !='\0';j++){
    str_final[k]=str2[j];
    printf("k=%d => %c \n",k,str_final[k]);
    k=k+1;
}
printf("\n second loop is over --\n");
printf("\n %s  \n",str_final); // Final string at the end of second loop

for(i=i;str1[i] != '\0';i++){
 str_final[k]=str1[i];
 printf("k=%d => %c \n",k,str_final[k]);
 k=k+1;
}
printf("\n third is over --\n");

str_final[k]='\0';
printf("\n Final String is here : \n %s  ",str_final);
return 0;
}
```

C Programming QB Module 3 Part C Solutions for CIE-1

@ Nishant

1. Predict the output of the following code.

```
#include <stdio.h>
int main() {
int arr1[]={97, 98, 99, 100, 101, 102, 104, 105};
int i=0;
while(i++ < 5)
printf("\n %c ", arr1[i++]);
return 0;
}
Output: b
        d
        f
```

2. Find the output of the following code.

```
#include <stdio.h>
void main() {
char a[5] = "IARE";
int i =0;
while(a[i])
printf("%s\n", (a + i++));
}
Output: ARE
        RE
        E
```

3. Find the output of the following code.

```
#include <stdio.h>
void main() {
char s1[10] = “abc”;
char s2[20];
s2 = s1;
printf(“%s”, s2);
}
Output: Error -> You cannot directly assign one array to another.
```

4. Find the output of the following code.

```
#include <stdio.h>
#include <string.h>
void main() {
char s[] = “hello”;
 int i = 0, n = strlen(s);
while(n) {
n--;
s[i] = s[n];
i++; }
printf(“%s”, s);
}
Output: olllo
```

5. Find the output of the following code.

```
#include <stdio.h>
void main() {
int a1[10], a2[10];
int i;
for(i=1; i<=9; i++) {
a1[i] = 'A' + i;
a2[i] = 'a' + i;
printf("%d\n", a2[i] -a1[i]); }
}
```

C PROGRAMMING SOLUTIONS

MODULE 3 CIE 2 PORTION

@N. Satish

PART A

11) Define the function with example in C

ANS) A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function.

SYNTAX:

```
return_type function_name(data_type parameter...){
//code to be executed
}
```

---

12)List the types of functions in C

ANS)There are two types of functions in C programming

i)Standard library functions

ii)User-defined functions

i)Standard library functions:

- These are built-in function in C programming.

- These functions are defined in header file. For example, the printf is to display output on the screen and scanf is used to read the values from the user

ii)User-defined functions:

- You can also create function as per your need. Such functions created by user are known as user-defined function.

13) Differentiate recursive function and non-recursive function.

ANS)Recursive function is a function which calls itself again and again . A recursion function in general has an external high time complexity while a non-recursion once does not . A recursion function generally has smaller code size whereas a non-recursive one is larger.

14) Explain various parameter passing methods in C.

ANS)A Parameter is the symbolic name for data that goes in to a function. There are two ways to pass parameters in c:

I) Call by reference

ii) Call by value

I) **Call by reference**: In this method of parameter passing , original values of variables are passed from calling program to function. Thus, Any change made in the function can be reflected back to the calling program

ii) **Call by value:** In this method of parameter passing, duplicate values of parameters are passed from calling program to function definition. Thus, Any change made in function would not be reflected back to the calling program.

---

15) Explain recursion with example.

ANS) In programming languages, if a program allows you to call a function inside the same function, then it is called a recursive call of the function.

Ex: to find the factorial of a number

SOURCE CODE:

```
#include <stdio.h>
int fact (int);
int main()
{
    int n,f;
    printf("Enter number to find factorial:");
    scanf(" %d",&n);
    f = fact(n);
    printf("factorial = %d",f);
}
int fact(int n)
{
    if (n==0)
        return 0;
    else if ( n == 1)
        return 1;
    else
```

```
        return n*fact(n-1);
}
```

OUTPUT: Enter number to find factorial: 5

factorial = 120

---

16) List the types of storage classes in C

ANS) Every Variable in a program has memory associated with it. Memory Requirement of Variables is different for different types of variables. In C, Memory is allocated & released at different places

Storage class of variable Determines following things :

- •Where the variable is stored
- •Scope of Variable
- •Default initial value
- •Lifetime of variable

Different types of Storage Classes:

Auto Storage Class

Static Storage

Class Extern Storage

Class resister storage class

---

17) Predict the output of the following code

SOURCE CODE:

```
#include <stdio.h>
int function(int,int);
int main()
{
    int a=25,b=24+1,c;
    printf("%d",function(a,b));
    return 0;
}
int function(int x,int y)
{
    return (x-(x==y));
}
```

OUTPUT:24

---

18) Predict the output of the following code

SOURCE CODE:

```
#include <stdio.h>
int main()
{
    function();
    return 0;
}
void function()
{
    printf("Function in C");
}
```

OUTPUT: Function in C

---

19) Predict the output of the following code

SOURCE CODE:

```
#include <stdio.h>
int function();
int main()
{
    int i;
    i=function();
    printf("%d",i);
    return 0;
}
function()
{
    int a;
    a=250;
}
```

OUTPUT: 0

---

20) Predict the output of the following code

SOURCE CODE:

```
#include <stdio.h>
int function();
int main()
{
    int i;
    i=function();
    printf("%d",i);
    return 0;
}
function()
{
    int a;
    a=250;
    return 0;
}
```

OUTPUT:

```
0
```

PART B

11) Write C program that uses both recursive and non-recursive functions to find the sum of n natural numbers

SOURCE CODE:

```
#include <stdio.h>
int addNumbers_recursive(int n);
int addNumbers_nonrecursive(int n);
int main() {
    int num;
```

```c
    printf("Enter a positive integer: ");
    scanf("%d", &num);
    printf("Sum of %d terms using recursion = %d\
n",num,addNumbers_recursive(num));
    printf("Sum of %d terms using nonrecursion= %d",num,
addNumbers_nonrecursive(num));
    return 0;
}

int addNumbers_recursive(int n)
{
    if (n != 0)
        return n + addNumbers_recursive(n - 1);
    else
        return n;
}
int addNumbers_nonrecursive(int n)
{
    int sum = 0,i;
    for(i = 1; i <= n; i++)
    {
        sum += i;
    }
    return sum;
}
```

OUTPUT:

```
Enter a positive integer: 10

Sum of 10 terms using recursion = 55

Sum of 10 terms using nonrecursion= 55
```

12) Write a C program that uses functions to convert decimal number to binary number.

SOURCE CODE:

```
#include <stdio.h>
int decimalToBinary(int num);
int main()
{
    int num;
    printf("Enter a decimal number: ");
    scanf("%d", &num);
    printf("Binary Number of %d is: %d",num,
decimalToBinary(num));
    return 0;
}
int decimalToBinary(int num)
{
    int bi=0,r,temp,i=1;
    temp=num;
    while (temp!=0)
    {
        r= temp%2;
        temp/=2;
        bi+=r*i;
        i*=10;
    }
    return bi;
}
```

OUTPUT:

```
Enter a decimal number: 13

Binary Number of 13 is: 1101
```

---

13 Write C program that uses functions to find the Nth Fibonacci number

SOURCE CODE:

```
#include <stdio.h>
int fibo(int);
int main()
{
    int num,result;
    printf("Enter the nth number in fibonacci series: ");
    scanf("%d", &num);
    if (num <0)
        printf("Fibonacci of negative number is not possible.\
n");
    else
    {
        result = fibo(num);
        printf("The %d number in fibonacci series is %d\n", num,
result);
    }
    return 0;
}
int fibo(int num)
{
    if (num == 1)
```

```
        return 0;
    else if (num == 2)
        return 1;
    else if(num==3)
        return 1;
    else
        return(fibo(num - 1) + fibo(num - 2));
}
```

OUTPUT:

```
Enter the nth number in fibonacci series: 5

The 5 number in fibonacci series is 3
```

14) Explain call by value and call by reference with example

ANS) There are two methods to pass the data into function in c language

i)call by value

ii)call by reference

i)**call by value**: In this method of parameter passing, duplicate values of parameters are passed from calling program to function defination. Thus, Any change made in function would not be reflected back to the calling program

Ex:

```
#include<stdio.h>
void change(int num)
{
    printf("Before adding value inside function num=%d \n",num);
    num=num+100;
    printf("After adding value inside function num=%d \n", num);
}
int main()
{
    int x=100;
    printf("Before function call x=%d \n", x);
    change(x);//passing value in function
    printf("After function call x=%d \n", x);
    return 0;
}
```

OUTPUT:

```
Before function call x=100
Before adding value inside function num=100
After adding value inside function num=200
After function call x=100
```

ii) **call by reference**: In this method of parameter passing , original values of variables are passed from calling program to function. Thus, Any change made in the function can be reflected back to the calling program

Ex:

```
#include<stdio.h>
void change(int *num) {
    printf("Before adding value inside function num=%d \
n",*num);
    (*num) += 100;
    printf("After adding value inside function num=%d \n",
*num);
}
int main() {
    int x=100;
    printf("Before function call x=%d \n", x);
    change(&x);//passing reference in function
    printf("After function call x=%d \n", x);
return 0;
}
```

OUTPUT:

```
Before function call x=100
Before adding value inside function num=100
After adding value inside function num=200
After function call x=200
```

15) Write a user defined function which takes an array of sorted integers and returns the value?

SOURCE CODE:

```
#include <stdio.h>
// function to sort the array in ascending order
void Array_sort(int *array , int n) {
    // declare some local variables
    int i=0 , j=0 , temp=0;
    for(i=0 ; i<n ; i++) {
        for(j=0 ; j<n-1 ; j++) {
            if(array[j]>array[j+1]) {
                temp        = array[j];
                array[j]    = array[j+1];
                array[j+1]  = temp;
            }
        }
    }
}

// function to calculate the median of the array
float Find_median(int array[] , int n) {
    float median=0;

    // if number of elements are even
    if(n%2 == 0)
        median = (array[(n-1)/2] + array[n/2])/2.0;
    // if number of elements are odd
    else
        median = array[n/2];

    return median;
}

int main()
{
    // declare two int arrays
    int array_1[30] = {0};
    // declare some local variables
    int i=0 ,n=0;
    float median=0;
    printf("\nEnter the number of elements for the array : ");
    scanf("%d",&n);
    printf("\nEnter the elements for array_1..\n");
    for(i=0 ; i<n ; i++)    {
        printf("array_1[%d] : ",i);
        scanf("%d",&array_1[i]);
    }

    // Sort the array in ascending order
    Array_sort(array_1 , n);

    // Now pass the sorted array to calculate the median of your
array.
    median = Find_median(array_1 , n);
    printf("\n\nThe median is : %.1f\n",median);
    return 0;
}
```

OUTPUT:

```
Enter the number of elements for the array : 5
Enter the elements for array_1
array_1[0] : 3
array_1[1] : 2
array_1[2] : 1
array_1[3] : 4
array_1[4] : 6
The median is : 3.0
```

---

16) Write C program that uses both recursive and non-recursive functions to find the factorial of a given number

SOURCE CODE:

```
#include <stdio.h>
int recfactorial(int x);
int nonrecfactorial(int x);
void main()
```

```
{
   int n;
   printf("Enter the number:");
   scanf("%d", &n);
   recfactorial(n);
   printf("The factorial of %d using recursive function is %d \
n",n, recfactorial(n));
   nonrecfactorial(n);
   printf("The factorial of %d using nonrecursive function is %d
",n,nonrecfactorial(n));
}
int recfactorial(int x)
{
   int f;
   if(x == 0)
    return(1);
   else
   {
    f = x * recfactorial(x - 1);
    return(f);
   }
}
int nonrecfactorial(int x)
{
   int i, f = 1;
   for(i = 1;i <= x; i++)
      f = f * i;
   return(f);
}
```

OUTPUT:

```
Enter the number:5

The factorial of 5 using recursive function is 120

The factorial of 5 using nonrecursive function is 120
```

---

17) Write a C program that uses functions to convert binary number to decimal number.

SOURCE CODE:

```c
#include<stdio.h>
int BinaryToDecimal(int n);
int main()
{
    int n;
    printf("Enter the Binary Value:");
    scanf("%d",&n);
    printf("Decimal Value of Binary number %d is:
%d",n11,BinaryToDecimal(n));
}
int BinaryToDecimal(int n)
{
    if(n==0)
        return 0;
    else
        return (n% 10 + 2* BinaryToDecimal(n / 10));
}
```

OUTPUT:

```
Enter the Binary Value:11

Decimal Value of Binary number 11 is: 3
```

---

18) Write a C program that uses functions to find 2‟s complement of a binary number.

SOURCE CODE:

```
#include <stdio.h>
#define SIZE 8
int main()
{
    char binary[SIZE + 1], onesComp[SIZE + 1], twosComp[SIZE +
1];
    int j, carry=1;
    printf("Enter %d bit binary value: ", SIZE);

    /* Input 8-bit binary string */
    gets(binary);

    /* Find ones complement of the binary number */
    for(j=0; j<SIZE; j++)
    {
        if(binary[j] == '1') {
            onesComp[j] = '0';
        }
        else if(binary[j] == '0') {
            onesComp[j] = '1';
        }
    }
    onesComp[SIZE] = '\0';

    /* Add 1 to the ones complement     */
    for(j=SIZE-1; j>=0; j--)
    {
        if(onesComp[j] == '1' && carry == 1)  {
            twosComp[j] = '0';
        }
        else if(onesComp[j] == '0' && carry == 1)  {
            twosComp[j] = '1';
            carry = 0;
        }
        else  {
            twosComp[j] = onesComp[j];
        }
    }
    twosComp[SIZE] = '\0';

    printf("Original binary = %s\n", binary);
    printf("Ones complement = %s\n", onesComp);
    printf("Twos complement = %s\n", twosComp);

    return 0;
}
```

OUTPUT: 1

```
Original binary = 1
Ones complement = 0
Twos complement = 1
```

19) Write a program in C to find the square of any number using the function.

Example : Input any number for square : 20

Expected Output : The square of 20 is : 400.00

SOURCE CODE:

```
#include<stdio.h>
int func(int);
int main()
{
    int no;
    float square;
    printf("Enter an no : ");
    scanf("%d",&no);
    square = func(no);
    printf("Square of no is : %.2f ", square);
}
int func(int temp)
{
    return temp*temp;
}
```

OUTPUT:

```
Enter an no : 20

Square of no is : 400.00
```

20) Write a program in C to check whether a number is a prime number or not using the function.

Example : Input a positive number : 5

Expected Output : The number 5 is a prime number.

<u>SOURCE CODE</u>:

```
#include<stdio.h>
int check_prime(int);
int main()
{
    int n, result;
    printf("Enter a number:");
    scanf("%d",&n);
    result = check_prime(n);
    if ( result == 1 )
        printf("%d is prime number", n);
    else
        printf("%d is not prime number", n);
    return 0;
}
int check_prime(int a)
{
    int i;
    for(i=2 ;i<=a-1;i++ )
    {
        if ( a%i== 0 )
        return 0;
    }
    return 1;
}
```

OUTPUT:

```
Enter a number:89

89 is prime number
```

PART C

6) Predict the output of the following code

SOURCE CODE:

```
#include<stdio.h>
int i;
int fun();
int main()
{
    while(i)
    {
        fun();
        main();
    }
    printf("Hello\n");
    return 0;
}
int fun()
{
    printf("Hi");
}
```

OUTPUT:

Hello

7) Predict the output of the following code

SOURCE CODE:

```
#include<stdio.h>
int reverse(int);
int main()
{
    int no=5;
    reverse(no);
    return 0;
}
int reverse(int no)
{
    if(no==0)
    return 0;
    else
    printf("%d,",no);
    reverse(no--);

}
```

OUTPUT: print 5 unlimited times

8) Predict the output of the following code

SOURCE CODE:

```
#include<stdio.h>
int main()
{
    int fun(int);
    int i=fun(10);
    printf("%d\n",--i);
    return 0;
}
int fun(int i)
{
    return (i++);
}
```

OUTPUT: 9

9) Predict the output of the following code

SOURCE CODE:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int i=0;
    i++;
    if(i<=5)
    {
        printf("Infosys");
        exit(1);
        main();
    }
    return 0;
}
```

OUTPUT: Infosys

10) Predict the output of the following code

SOURCE CODE:

```
#include<stdio.h>
int check(int);
int main()
{
    int i=45,c;
    c=check(i);
    printf("%d\n",c);
    return 0;
}
int check(int ch)
{
    if(ch>=45)
    return 100;
    else
    return 10;
}
```

OUTPUT: 100