

**EXERCISE****MCQ**

1. Which of the following statements is true?  
(a) C Library functions provide I/O facilities      (b) C inherent I/O facilities  
(c) C does not have I/O facilities      (d) Both (a) and (c)  
**Ans.** (a)
2. Header files in C contain  
(a) Compiler commands      (b) Library functions  
(c) Header information of C programs      (d) Operators for files  
**Ans.** (b)
3. Which pair of functions below are used for single character I/O.  
(a) Getchar() and putchar()      (b) Scanf() and printf()  
(c) Input() and output()      (d) None of these  
**Ans.** (a)
4. The printf() function returns which value when an error occurs?  
(a) Positive value      (b) Zero      (c) Negative value      (d) None of these  
**Ans.** (c)
5. Which of the following is character oriented console I/O function?  
(a) getchar() and putchar()      (b) gets() and puts()  
(c) scanf() and printf()      (d) fgets() and fputs()  
**Ans.** (a)
6. Which header file is essential for using strcmp() function?  
(a) string.h      (b) strings.h      (c) text.h      (d) strcmp.h  
**Ans.** (a)

**Questions**

1. What is function? Explain function prototype giving example.
2. Explain advantages of using the function.
3. Give difference between library functions and user defined functions.
4. Explain different categories of functions.
5. What are formal parameters? What is actual argument?
6. Explain call by value and call by reference with example.
7. What is recursion? Explain with example.
8. Write a function called menu which prints the text string "Menu choices". The function does not pass any data back, and does not accept any data as parameters. Write a function prototype for the above function.

9. Write a function called print which prints a text string passed to it as a parameter (i.e., a character based array). Write a function prototype for the above function print.
10. Write a function called total, which totals the sum of an integer array passed to it (as the first parameter) and returns the total of all the elements as an integer. Let the second parameter to the function be an integer which contains the number of elements of the array. Write a function prototype for the above function.

### ■ Programming Exercises

1. Write a program in C which incorporates a function using parameter passing and performs the addition of three numbers. The main section of the program is to print the result.
2. Write a C program incorporating a function to add all elements of a two dimensional array. The numbers of rows are to be passed to the function, and it passes back the total sum of all elements (Use at least a  $4 \times 4$  array).
3. Write a function that converts its input parameter (a double) from a temperature in Fahrenheit to a temperature in Celsius.
4. Write a function that converts its input parameter (a double) from a temperature in Celsius to a temperature in Fahrenheit.
5. Write a function that takes in a string, s, and an integer, n, and prints that string exactly n times, once per line.
6. Write a function that takes in a string, s and a character c, and returns the number of times the character occurs in the string.
7. Write a function that takes in a single positive integer n and returns n! (n factorial). Note that  $n! = 1 \times 2 \times 3 \times 4 \dots \times n$ .
8. Write a program to find out the addition of n numbers using command line argument.

### GTU Exam Paper Solution

#### Questions 2 Marks

1. List out categories of the user-defined function.

**Summer 2020**

**Ans.** Please refer 5.6

2. Justify the statement: main() is user defined function.

**Summer 2020(New)**

**Ans.** We don't need to include any header file to use main function. We must give definition for our code in main function.

3. Give difference between AutoAnd Extern.

**Summer 2021**

**Ans.** Please refer 5.10

4. Enlist elements of User Define Function and explain anyone.

**Summer 2021**

Please refer 5.5

**5. Functions**

5. Write C example of strcat() function.

Summer 2021

**Ans.**

```
#include<stdio.h>
#include<string.h>
int main()
{
    char text1[10],text2[5];
    int length;
    printf("Enter string1:");
    gets(text1);
    printf("\nEnter string2:");
    gets(text2);
    strcat(text1,text2);
    printf("\nString1:%s",text1);
    printf("\nString2:%s",text2);
    return 0;
}
```

6. Explain how you can define constant variable in program without using const keyword?

Summer 2021

**Ans.** Using #define directive . For ex

#define PI 3.14

7. Explain floor() with example.

Summer 2021

**Ans.** **floor ( ) :** This function returns the nearest integer which is less than or equal to the argument passed to this function.

```
#include<stdio.h>
#include <math.h>
int main(){
    printf("\n%f",floor(3.6));
    printf("\n%f",floor(3.2));
    return 0;
}
```

**Output:**

3.000000  
3.000000

**Questions 3 Marks**

1. Explain any three functions of math.h with example.

**Ans.** **floor ( ) :** This function returns the nearest integer which is less than or equal to the argument passed to this function.**ceil ( ) :** This function returns nearest integer value which is greater than or equal to the argument passed to this function.**sqrt ( ) :** This function is used to find square root of the argument passed to this function.

**Example**

```
#include<stdio.h>
#include <math.h>
int main()
{
    printf("\n%lf",ceil(3.6));
    printf("\n%lf",ceil(3.3));
    printf("\n%lf",floor(3.6));
    printf("\n%lf",floor(3.2));
    printf("\n%lf",sqrt(16));
    printf("\n%lf",sqrt(7));
    return 0;
}
```

2. Discuss about function passing argument and no return.

**Ans.** Please refer 5.6.2

**Summer 2021(New)**

3. Explain static variable.

**Summer 2021(New)**

**Ans.** A static int variable remains in memory while the program is running. A normal or auto variable is destroyed when a function call where the variable was declared is over.

```
#include<stdio.h>
intfun()
{
    staticintcount = 0;
    count++;
    returncount;
}

intmain()
{
    printf("%d ", fun());
    printf("%d ", fun());
    return0;
}
```

**Output:**

1 2
-----

4. Explain working of a function.What happens in background when a function is called?

**Summer 2021**

**Ans.** Please refer 5.4

5. Enlist all types of function and explain function with return type with argument.

**Summer 2021**

**Ans.** Please refer 5.6

6. Give advantage and disadvantage of function.

Summer 2021

**Ans.** Here are several advantages of using functions in your code:

- Use of functions enhances the *readability* of a program. Breaking the code in smaller Functions keeps the program organized, easy to understand and makes it reusable.
- The C compiler follows *top-to-down* execution, so the control flow can be *easily managed* in case of functions.
- It reduces the complexity of a program and gives it a *modular structure*.

#### Disadvantages of using functions

- There may not be any speed advantage.
- It's less portable..

7. Enlist all string built in function and explain strrev() and strcpy().

Summer 2021

**Ans.**

Function	Syntax (or) Example	Description
strcpy()	strcpy(string1, string2)	Copies string2 value into string1
strncpy()	strncpy(string1, string2, 5)	Copies first 5 characters string2 into string1
strlen()	strlen(string1)	returns total number of characters in string1
strcat()	strcat(string1, string2)	Appends string2 to string1
strcmp()	strcmp(string1, string2)	Returns 0 if string1 and string2 are the same; less than 0 if string1<string2; greater than 0 if string1>string2
strlwr()	strlwr(string1)	Converts all the characters of string1 to lower case.
strupr()	strupr(string1)	Converts all the characters of string1 to upper case.
strstr()	strstr(string1, string2)	Returns a pointer to the first occurrence of string2 in string1

#### 1. strrev()

The strrev() function reverses the string and original string is overwritten. It takes the form:

**strrev(s1);**

**Example :** If s1="there",

strrev(s1);

Output will be s1="ereht".

#### 2. strcpy()

The strcpy() function copy one string into another string. It takes the form

**strcpy(s1,s2);**

**Example :** If s1="the" and s2="their",

strcpy(s1,s2);

output will be s1="their" and s2="their". Because it will copy string s2 to s1.

## Questions 4 Marks

1. Differentiate : autovariable vs global variable.

Summer 2020

Ans. Please refer 5.10

2. Write a recursive function to multiply two numbers.

Summer 2020

Ans.

```
#include <stdio.h>
#include <stdlib.h>
int product(int,int);
int main()
{
    int num1,num2,result;
    printf("Enter two number to find their product\n");
    scanf("%d %d",&num1,&num2);
    result=product(num1,num2);
    //function call
    printf("Product of %d and %d is %d\n",num1,num2,result);
    return 0;
}
int product(int a, int b)
{
    if(a<b)
    {
        return product(b,a);
    }
    else if(b!=0){
        return (a+product(a,b-1));
    }
    else{
        return 0;
    }
}
```

3. Explain return statement with example.

Summer 2020

Ans. A return statement ends the execution of a function, and returns control to the calling function. Execution resumes in the calling function at the point immediately following the call. A return statement can return a value to the calling function

```
#include <stdio.h>
void Print()
{
    printf("Welcome to C World");
}

int main()
{
    Print();
    return 0;
}
```

4. Write a C program that input a string and reverse the string without string.h

Summer 2020

**Ans.**

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

void main()
{
    int i,n;
    char str[20];
    printf("Enter the String to get reversed\n");
    gets(str);
    n=strlen(str);
    printf("\nReversed string is \n");
    for(i=n-1;i>=0;i--)
    {
        printf("%c",str[i]);
    }
}
```

5. Explain the following string and lingfunctions with example:

Summer 2020(New)

1. strcat()
2. strlen()

**Ans.** Please refer 5.9.1

6. Explain the following stringh and lingfunctions with example:

Summer 2020(New)

1. strcpy()
2. strcmp()

**Ans.** Please refer 5.9.1

7. Write a C program to illustrate call-by-value parameter passing technique.

Summer 2020(New)

**Ans.** Please refer 5.7

8. Write a C program to explain call-by-reference parameter passing technique.

Summer 2020(New)

**Ans.** Please refer 5.7

9. Explain local variable and global variable with suitable example.

Summer 2020(New)

Ans.	Local	Global
	<ol style="list-style-type: none"> <li>1. It is declared inside a function.</li> <li>2. If it is not initialized, a garbage value is stored</li> <li>3. It is created when the function starts execution and lost when the functions terminate.</li> <li>4. Data sharing is not possible as data of the local variable can be accessed by only one function.</li> <li>5. When the value of the local variable is modified in one function, the changes are not visible in another function.</li> </ol>	<ol style="list-style-type: none"> <li>1. It is declared outside the function.</li> <li>2. If it is not initialized zero is stored as default.</li> <li>3. It is created before the program's global execution starts and lost when the program terminates.</li> <li>4. Data sharing is possible as multiple functions can access the same global variable.</li> <li>5. When the value of the global variable is modified in one function changes are visible in the rest of the program.</li> </ol>

10. Give advantage and disadvantage of recursion.

**Ans. Advantages of recursion**

1. The code may be easier to write.
2. To solve such problems which are naturally recursive such as tower of Hanoi.
3. Reduce unnecessary calling of function.
4. Extremely useful when applying the same solution.
5. Recursion reduce the length of code.
6. It is very useful in solving the data structure problem.
7. Stacks evolutions and infix, prefix, postfix evaluations etc.

**Disadvantages of recursion**

1. Recursive functions are generally slower than non-recursive function.
2. It may require a lot of memory space to hold intermediate results on the system stacks.
3. Hard to analyze or understand the code.
4. It is not more efficient in terms of space and time complexity.
5. The computer may run out of memory if the recursive calls are not properly checked.

11. Discuss the differences between actual and formal arguments.

**Ans.**

<b>Actual Parameters</b>	<b>Formal Parameters</b>
<ol style="list-style-type: none"> <li>1. When a function is called, the values that are passed in the function call are called the arguments or actual parameters.</li> <li>2. Actual Parameters are the parameters which are in calling subprogram.</li> <li>3. There is no need to specify datatype in actual parameter.</li> <li>4. The parameters are written in function call are known as actual parameters.</li> <li>5. Actual Parameters can be constant values or variable names.</li> </ol>	<ol style="list-style-type: none"> <li>1. The parameter used in function definition statement which contain data type on its time of declaration is called formal parameter.</li> <li>2. Formal Parameters are the parameters which are in called subprogram.</li> <li>3. The datatype of the receiving value must be defined.</li> <li>4. The parameters are written in function definition are known as formal parameters.</li> <li>5. Formal Parameters can be treated as local variables of a function in which they are used in the function header.</li> </ol>

