https://shikshamentor.com/programming-in-c-for-msbte-k-scheme/

312303 - Programming In 'C' (Sem II)

As per MSBTE's K Scheme

CO / CM / IF / AI / AN / DS

Unit IV — Functions — Marks - 14

S. N.	MSBTE Board Asked Questions	Exam Year	Marks
1	**List the categories of user defined function.** ANS: There are four types of user-defined functions: 1. Function with no arguments and no return type. 2. Function with no arguments and a return type. 3. Function with arguments and no return type. 4. Function with arguments and with return type.	W-23	2M

2	**Develop a program to find the factorial of a number using recursion.**
	ANS:
	`#include<stdio.h> #include <conio.h>`
	`int factorial(int no)`
	`{`
	`if(no==1) return(1); else`
	`return(no*factorial(no-1));`
	`}`
	`void main()`
	`{`
	`int fact,no;`
	`clrscr();`
	`printf("\n Enter number: "); scanf("%d",&no);`
		W-23,
		S-22
			4M

	fact=factorial(no);
	printf("\n Factorial number=%d",fact);
	 getch();
	}
	<u>Output</u>
	Enter number :5
	Factorial of number is :120		

3	**List any two string handling function.** ANS: a. strcpy(str1, str2) b. strlen(str) c. strcat(str1, str2) d. strcmp(str1, str2)	S-23	2M
4	**Explain use of any two math function with example** ANS: Syntax: a)double ceil (double b) This function returns the smallest integer value that is greater or equal to b and rounds the value upwards. For a negative value, it moves towards the left. Example 3.4 returns -3 has the output. Example: This program explains by taking input in the float argument and returns the ceil value. #include <stdio.h> #include <math.h> int main() { float n, ceilVal; printf(" Enter any	S-23	4M

```
Numeric element
: ");
scanf("%f", &n);
ceilVal = ceil(n);
printf("\n
The Value of %.2f =
%.4f ", n, ceilVal);
return 0;
}
```

b) sqrt ()

This function returns the square root of a specified number.

Syntax:

sqrt(arg)

Example:

The below code explains the most known mathematical function sqrt() by taking ‘n’ values to compute the square root for the different ‘n’ values.

```
#include <stdio.h>
#include <math.h>
int main()
{
double n,output;
printf("Enter a
number\n");
scanf("%lf", &n);
output = sqrt(n);
printf("Square root
of %.2lf = %f",
n,output);
return 0;
}
```

5	**Explain call by value with example.**	**S-23**	**4M**

ANS:

The call by value method of passing arguments to a function copies the actual value of an argument into the formal parameter of the function. In this case, changes made to the parameter inside the function have no effect on the argument.

By default, C programming uses call by value to pass arguments. In general, it means the code within a function cannot alter the arguments used to call the function. Consider the function swap() definition as follows.

```
/* function definition to swap the values */
void swap(int x, int y)
 {
int temp;


temp = x; /* save the value of x */ x = y;  /* put y into x */
y = temp; /* put temp into y */


return;
}
```

Now, let us call the function swap() by passing actual values as in the following example –

```
  Program:
   #include <stdio.h>
  /* function declaration */ void swap(int x, int y);


int main ()
```

```
 {


   /* local variable definition */ int a = 100;
int b = 200;


   printf("Before swap, value of a : %d\n", a ); printf("Before s
   value of b : %d\n", b );


   /* calling a function to swap the values */ swap(a, b);


   printf("After swap, value of a : %d\n", a ); printf("After swap, v
   of b : %d\n", b );


   return 0;
   }
   void swap(int x, int y) { int temp;
temp = x; /* save the value of x */ x = y; /* put y into x
*/
y = temp; /* put temp into y */


return;
}
```

Let us put the above code in a single C file, compile and execute it, it will produce the following result Before swap, value of a : 100 Before swap, value of b : 200 After swap, value of a : 100 After swap, value of b : 200

6	**Write a C program to generate Fibonacci series for given number using recursion.**	**S-23**	**6M**

ANS:

```
#include <stdio.h>

#include <conio.h>

int fibonacci(int n)

{

  if(n == 0)

    return 0;

  else if(n == 1)

    return 1;

  else

 return (fibonacci(n-1) + fibonacci(n-2));

}

int main()

{

  int n;

  printf("Enter the number of terms\n");

  scanf("%d", &n);

  printf("Fibonacci Series: ");

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

   printf("%d ", fibonacci(i));

  }

  return 0; }
```

7	**Write a C program to find Factorial of number using recursion.**
	ANS:
	
	`#include<stdio.h>`
	`#include<conio.h>`
	`int main()`
	`{`
	`int i,fact=1,number;`
	`printf("Enter a number: ");`
	`scanf("%d",&number);`
	`for(i=1;i<=number;i++)`
	`{`
	`fact=fact*i;`
	`}`
	`printf("Factorial of %d is: %d",number,fact);`
	`return 0;`
	`}`
		S-23	**4M**

8	**State any two advantages of function** **ANS:** **1) Big code can be difficult to read, so when divided into smaller functions, it increases readability.** **2) Program becomes modular.** **3) It reduces complexity in debugging.** **4) It enhances reusability of the code.**	**W-22**	**2M**

9	**Describe how recursive function is used in calculating factorial of a number.**	**W-22**	**4M**

ANS:

Recursive function :

Recursion is a process of calling a function by itself. a recursive function body contains a function call statement that calls itself repetitively.

Example: for calculating factorial of a number using recursion function call from main() : fact(n); // consider n=5

Function definition:

```
int
fact(int n)
{
if(n==1)
return 1;
else
return(n*fact(n-1));
}
```

In the above recursive function a function call fact (n-1) makes a recursive call to fact function. Each time when a function makes a call to itself, it save its current status in stack and then executes next function call. When fact ( ) function is called from main function, it initializes n with 5. Return statement inside function body executes a recursive function call. In this call, first value of n is stored using push ( ) operation in stack (n=5) and a function is called again with value 4(n-1). In each call, value of n is push into the stack and then it is reduce by 1 to send it as argument to recursive call. When a function is called with n=1, recursive process stops. At the end all values from stack are retrieved one by one using pop ( ) operation to perform multiplication to calculate factorial of number.

10	**Write a Cprogram to find area of circle using function.**
	Note: Any type of function declaration and definition should be considered (with
	return value or no return value or with parameter or no parameter)
	ANS:
	#include<stdio.h>
	#include<conio.h>
	void circle_area(float radius) //user defined function.
	{
	float area;
	area=3.14*radius*radius;
	printf("Area of circle= %f",area);
	}
	void main() //main function
	{
	float r;
	printf("Enter the radius of circle : ");
	scanf("%f", &r);
	circle_area(r);
	getch();
	}
		W-23, W-22	**4M**

11	**Write a C program to find Factorial of number using recursion.** **ANS:** **#include<stdio.h>** **#include<conio.h>** **int main()** **{** **11. int i,fact=1,number;** **12. printf("Enter a number: ");**	**W-22**	**6M**

	scanf("%d",&number);
	for(i=1;i<=number;i++)
	{
	fact=fact*i;
	 }
	printf("Factorial of %d is: %d",number,fact);
	return 0;
	}		

12	**List the categories of functions and explain any one with example.**
	
	Different categories of function:
	. **Function with no arguments and no return value.**
	. **Function with arguments and no return value.**
	. **Function with no arguments and return value.**
	. **Function with arguments and return value.**
	. **Function with no arguments and no return value: This category of function cannot return any value back to the calling program and it does not accept any arguments also. It has to be declared as void.**
	
	For example:
	void add()
	{
	inta,b,c; a=5; b=6;
	c=a+b; printf("%d",c);
	}
	It should be called as add();
		S-22,W-19	**4M**

6. Function with arguments and no return value: This category of function cannot return any value back to the calling program but it takes arguments from calling program. It has to be declared as void. The number of arguments should match in sequence, number and data type.

Function with arguments and return value: This category of function can return a value back to the calling program but it also takes arguments from calling program. It has to be declared with same data type as the data type of return variable.
It should be called as int s = add(4,5); where x will have 4 and y will have 5 as their values and s will store value returned by the function.

13 Give any four differences between call by value and call by reference. S-22,W-19 4M

ANS:

Sr. No.	Call by value	Call by reference
1.	When function is called by passing values then it is call by value	When function is called by passing address of variable then it is called as call by reference.
2.	Copy of actual variable is created when function is called.	No copy is generated for actual variable rather address of actual variable is passed.
3.	In call by value, memory required is more as copy of	In call by reference, memory required is

	variable is created.	less as there is no copy of actual variables.
4.	Example:- Function call - Swap (x,y); Calling swap function by passing values.	Example:- Function call – Swap (&x, &y); Calling swap function by passing address
5.	Original (actual) parameters do not change. Changes take place on the copy of variable.	Actual parameters change as function operates on value stored at the address.

14	State the syntax of strlen() and strcat() function ANS: a. Syntax: strlen(str): Used to find length of the string b. Syntax: strcat(str1, str2): Used to join two strings	W-19	2M

15	Calculate factorial of a number using recursion
	ANS:
	#include<stdio.h>
	#include<conio.h>
	int factorial(int no)
	{
	if(no==1) return(1);
	else
	return(no*factorial(no-1));
	}
	void main()
	{
	int fact,no;
	clrscr();
	printf("\n Enter number: ");
		W-19	6M

```
scanf("%d",&no);
fact=factorial(no);
printf("\n Factorial number=%d",fact);
getch();
}
```

16	**Distinguish between call by value and call by reference.**	W-23,s-19	4M

ANS:

Sr. No.	Call by value	Call by reference
1.	When function is called by passing values then it is call by value	When function is called b passing address of variable then it is called as call by reference.
2.	Copy of actual variable is created when function is called.	No copy is generated for actual variable rather address of actual variable is passed.
3.	In call by value, memory required is more as copy of variable is created.	In call by reference, memor required is less as there is no copy of actual variables.
4.	Example:- Function call - Swap (x,y); Calling swap function by passing values.	Example:- Function call – Swap (&x, &y); Calling swap function by passing address
5.	Original (actual) parameters do not change. Changes take place on the copy of variable.	Actual parameters change a function operates on value stored a the address.

17	**Write a program to reverse the number 1234 (i.e. 4321) using function.**
	(Note: Any other correct logic shall be considered).
	ANS:
	`#include<stdio.h>`
	`#include<conio.h>`
	`void findReverse();`
	`void main()`
	`{`
	`findReverse();`
	`}`
	`void findReverse()`
	`{`
	`int num, res=0,ans=0;`
	`clrscr();`
	`printf("Enter the number");`
	`scanf("%d", &num);`
	`while(num!=0)`
	`{`
	`res=num%10;`
	`ans=ans*10+res;`
	`num=num/10;`
	`}`
	`printf("Reverse number is %d", ans);`
	`getch();`
	`}`
		S-19	**4M**

18	**Write a program to perform addition, subtraction, multiplication and division of two integer number using function.**
	(Note: Any other correct logic shall be considered).
	ANS:
	`#include<stdio.h>`
	`#include<conio.h>`
	`void add(int x,int y)`
		S-19	**4M**

```
{
printf("\nAddition=%d",x+y);
}
void sub(int x,int y)
{
printf("\nSubtraction=%d",x-y);
}
void mult(int x,int y)
{
printf("\nMultiplication=%d",x*y);
}
void div(int x,int y)
{
printf("\nDivision=%d",x/y);
}
void main()
{
intx,y;
clrscr();
printf("Enter x:");
scanf("%d",&x);
printf("Enter y:");
scanf("%d",&y);
add(x,y);
sub(x,y);
mult(x,y);
div(x,y);
getch();
}
```

19	Explain recursion with suitable example. List any two advantages.
	ANS:
	Recursion means a function calls itself repetitively. A recursive function contains a function call to itself inside its body.
	Example:
	#include<stdio.h>
	#include<conio.h>
	int factorial(int N);
	void main()
	{
	int N,fact;
	clrscr();
	printf("Enter number:");
	scanf("%d",&N);
	fact=factorial(N);
	printf("\n Factorial is:%d",fact);
	getch();
	}
	int factorial(int N)
	{
	if(N==1)
	return(1);
	else
	return(N*factorial(N-1));
	}
	Advantages:
	Reduces length of the program
	Reduces unnecessary calling of a function
	Useful when same solution is to be applied many times.
		S-19	6M

20	**Explain User defined function with example.**	**S-18**	**4M**

ANS:

Functions are basic building blocks in a program. It can be predefined/ library functions or user defined functions. Predefined functions are those which are already available in C library. User defined functions are those which are written by the users to complete a specific task. Execution of a C program starts from main(). User defined functions should be called from main() for it to execute. A user defined function has a return type and a name. it my or may not contain parameters.

The general syntax of a user defined function : Return_type func_name(parameter list)

Example:

```
#include<stdio.h>
 #include<conio.h>
void myFunc(int a)
{
printf("The value is: %d",a);
}
void main()
{
myFunc(10);
getch():
}
```

21	**Write a program to accept marks of four subjects as input from user. Calculate and display total and percentage marks of students using function.**
	ANS:
	`#include<stdio.h>`
	`#include<conio.h>`
	`void main() {`
	`float marks[4];`
	`float total=0.0, perc=0.0;`
	`int i;`
	`clrscr();`
	`for(i=1;i<=4;i++)`
	`{`
	`printf("Enter marks of subject %d",i);`
	`scanf("%f%",&marks[i]);`
	`}`
	`for(i=1;i<=4;i++){`
	`total=total+marks[i];`
	`}`
	`printf("Total is :%f",total);`
	`perc=total/4;`
	`printf("Percentage is %f",perc);`
	`getch();`
	`}`
		S-18	4M

22	**Write a program to swap two numbers using call by value.**
	ANS:
	`#include<stdio.h>`
	` #include<conio.h>`
	`void swap(int a, int b)`
	`{`
	`int temp;`
		S-18	4M

	temp=a;
	 a=b;
	b=temp;
	printf("Numbers after swapping no1=%d and no2=%d",a,b);
	}
	void main()
	 {
	 int no1, no2;
	clrscr();
	printf("Enter the 2
	numbers");
	 scanf("%d%d",&no1,&no2)
	;
	printf("Numbers before swapping no1=%d and no2=
	%d",no1, no2);
	swap(no1,no2);
	getch();
	}		

23	**Write a program to print values of variables and their addresses.**
	ANS:
	
	#include<stdio.h>
	#include<conio.h>
	int main()
	{
	   int num;
	   printf("Enter any number to store in \"num\" variable: ");
	   scanf("%d", &num);
	   printf("\nValue of num = %d", num);
	   printf("\nAddress of num = %u", &num);
	   getch();
	   return 0;
	}
		W-23	6M

24	**If the value of a number (N) is entered through keyboard. Write a program using recursion to calculate and display factorial of number(N).**
	ANS:
	#include<stdio.h>
	#include<conio.h>
	int factorial(int
	num)
	{
	if(num==1)
	{
	return 1;
	}
	else
	{
	return(num*factorial(num-1));
	}
	}
	void main()
	{
	int num;
	int result;
	clrscr();
	printf("Enter a
	number");
	scanf("%d",&num);
	result=factorial(num);
	printf("Factorial of %d is
	%d",num,result);
	getch();
	}
		W-18	**6M**

25	**State any four math functions with its use.** ***(Note: Any other relevant math function shall be considered)*** **ANS:** **Math Functions:** **sqrt() - square root of an integer abs() - absolute value of an integer** **sin() - compute the sine value of an input value cos()- compute the cosine value of an input value pow()- compute the power of a input value** **floor()- round down the input value** **ceil()- round up the input value**	**S-18**	**2M**
26	**plain following functions: getchar( )** **putchar( )** **getch( )** **putch( ) with suitable examples.** **ANS:** **getchar( ) -** **It is function from stdio.h header file. This function is used to input a single character.** **The enter key is pressed which is followed by the character that is typed. The character that is entered is echoed.** ***Syntax:*** **ch=getchar();** ***Example:*** **void main()** **{** **char ch;** **ch = getchar();** **printf("Input Char Is :%c",ch);** **}** **During the program execution, a single character gets or read through the getchar(). The given value is displayed on the**		

	screen and the compiler waits for another character to be typed. If you press the enter key/any other characters and then only the given character is printed through the printf function.
	
	putchar( ) -
	It is used from standard input (stdio.h) header file.
	This function is the other side of getchar. A single character is displayed on the screen.
	Syntax:
	putchar(ch);
	void main()
	{
	    void main()
	{
	char ch='a';
	putch(ch)
	    }
	 getch():
	 getch() method pauses the Output Console until a key is pressed.
	 It does not use any buffer to store the input character.
	putch(): is used to display a single character on the console without buffering.		

27	Develop a program to find factorial of a number using recursion. *(Note: Any other relevant logic shall be considered)* ANS: #include<stdio.h> #include<conio.h> int factorial(int num) { if(num==1)	S-18	4M

	{
	return 1;
	}
	else
	{
	return(num*factorial(num-1));
	}
	}
	void main()
	{
	int num;
	int result;
	 clrscr();
	printf("Enter a number");
	 scanf("%d",&num);
	 result=factorial(num);
	printf("Factorial of %d is %d",num,result);
	getch();
	}		

28	**Develop a program to find diameter, circumference and area of circle using function.**
	(Note: Any other relevant logic shall be considered)
	ANS:
	#include<stdio.h>
	#include<conio.h>
	void circle(float r)
	{
	float diameter,circumference,area;
	diameter=2*r;
	printf("\n Diameter=%f",diameter);
	circumference=2*3.14*r;
	printf("\n Circumference=%f",circumference);
	area=3.14*r*r;
	printf("\n Area=%f",area);
		S-18	6M

	}
	void main()
	{
	float radius;
	clrscr();
	printf("\n Enter radius:");
	scanf("%f",&radius);
	circle(radius);
	getch();
	}		

Thank You

https://shikshamentor.com/programming-in-c-for-msbte-k-scheme/

Visit

https://shikshamentor.com/

SHIKSHA MENTOR
THE PERFECT MENTOR FOR YOUR BRIGHT CAREER