Scheme of Evaluation
(Scoring Indicator)

Revision :(15) PROGRAMMING IN C Course Code :4044

Course Title :Diploma In Electronics Engineering

(NB. Marks may be given to programs with suitable logic. One particular logic has been specified in this scheme)
(Distribution of marks for programs : Declaration- 1 , Input- 2 or 1 , Computation -3, Output- 1)

Qn. No	Scoring Indicator	Split up Score	Sub Total	Total
	PART-A			
1	printf(),puts(),putchar(),putch()	1+1	2	2
2	character or String	2	2	2
3	Break the execution of a loop or exit from a loop	2	2	2
4	strcat,strlen,strcmp,strcat (any two)	1+1	2	2
5	Call by value and call by reference	1+1	2	2
II.	**PART - B**			

1	Structure of a C Programme
	Documentation Section
	Definition Section
	Global Declaration Section
	Main()
	{
	Local Declarion
	Body of the the C Program
	}
	Function 1()
	{
	Body
	} (Explain briefly the Structure)
		6	6	6

2	Exit Control Loop : The Condition of the loop is specified after the body of the loop, First time the loop is executed, Entry Control Loop : The Condition is check before the exxection of the loop, if the condition is false then the loop cannot be executed , the counter varieble initilized before starting the loop (Expain with clow chart also)	6	6	6

3	Main()
	int f;
	float c;
	printf("Enter a Faren heat Value ");
	scanf("%d",&f);
	c= 5/9 *(f-32);
	printf("%d Faren heat is %f Centigrade ",f,c);
	}
		6	6	6

4	main()
	int n,r,x;
	printf("Enter a number ");
	scanf("%d",&n);
	while (n !=0)
	{
	x=n%10;
	r=r*10+x;
	n=n/10;
	}
	printf("The Reverse of the number is %d",r);
	}
		6	6	6

5	# include <string.h)
	main()
	{
	char str1[10],str2[10];
	printf("Enter the first String");
	scanf("%s",str1);
	printf("Enter the second String");
	scanf("%s".str2);
	strcat(str1,str2);
	printf("%s %s",str1,str2);
	}
		6	6	6

6	The user defined functions in C is used to create separate programming blocks or fuctions for repetative tasks. The funcions will improve the modularity of the programm. By using the functions to avoid the length of the program.By using the function we can easly debug the program,test the program, and maintaine the program Declarion of a Function retun type Function name (List of argumens) Briefly Explain)	6	6	6

7	Call by Value :In this method, the value of each variable in calling function is copied into corresponding dummy variables of the called function. In this method, the address of actual variables in the calling function are copied into the dummy variables of the called function. The changes made to the dummy variables in the called function have no effect on the values of actual variables in the calling function. With this method, using addresses we would have an access to the actual variables and hence we would be able to manipulate them. In this method, the address of actual variables in the calling function are copied into the dummy variables of the called function. Call by refernce With this method, using addresses we would have an access to the actual variables and hence we would be able to manipulate them. In this method, the address of actual variables in the calling function are copied into the dummy variables of the called function	3+3	6	6
III.a	Arithmatic Operator : + - * / ^ % , Expalin Each Relational Opearors : > < >= <= != == Expalin Each with example	4+4	8	8

b.	#include <stdio.h>
	#include <math.h>
	
	main()
	{
	
	float radius;
	float surface_area, volume;
	
	printf("Enter radius of the sphere : \n");
	scanf("%f", &radius);
	volume = (4.0/3) * (22/7) * radius * radius * radius;
	printf("\n Volume of sphere is : %.3f", volume);
	
	}
		7	7	7

IV.a	The major Data types in C are :- Fundamental Data types, Derived Data types, User Defined data Types The Fundamental Data types or Prinamry Data types :- Intiger, Character, Float, Void Derived Data Types : Arrays, Structure User defined : Enum (Expalin briefly Each)	**8**	**8**	**8**

b.	`#include <stdio.h>`
	`main()`
	`{`
	`int n1, n2, n3;`
	`printf("Enter three numbers: ");`
	`scanf("%lf %lf %lf", &n1, &n2, &n3);`
	`if (n1>=n2)`
	`{`
	`if(n1>=n3)`
	`printf("%.2lf is the largest number.", n1);`
	`else`
	`printf("%.2lf is the largest number.", n3);`
	`}`
	`else`
	`{`
	`if(n2>=n3)`
	`printf("%.2lf is the largest number.", n2);`
	`else`
	`printf("%.2lf is the largest number.",n3);`
	`}`
	`}`
		7	**7**	**7**

V.a	The major Looping Statements are : Do loop, While Loop, For Loop (Expalin Any two with exampleand flow chart)	**4+4**	**8**	**8**

b.	/* Fibonacci series program in C language */
	#include <stdio.h>
	
	int main()
	{
	int n, first = 0, second = 1, next, c;
	
	printf("Enter the number of terms\n");
	scanf("%d", &n);
	
	n=20;
	
	for (c = 0; c <= n; c++)
	{
	if (c <= 1)
	next = c;
	else
	{
	next = first + second;
	first = second;
	second = next;
	}
	printf("%d\n", next);
	}
		7	7	7

VI.a	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 useful to think of an array as a collection of variables of the same type. Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element. To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows – type arrayName [arraySize]; Eg : int a[10] int a[10][10] char name [20][10]	8	8	8

b.	```
	void main()
	{
	
	int i, j, a, n, number[30];
	printf("Enter the value of N \n");
	scanf("%d", &n);
	
	printf("Enter the numbers \n");
	for (i = 0; i < n; ++i)
	scanf("%d", &number[i]);
	
	for (i = 0; i < n; ++i)
	{
	
	for (j = i + 1; j < n; ++j)
	{
	
	if (number[i] > number[j])
	{
	
	a = number[i];
	number[i] = number[j];
	number[j] = a;
	
	}
	
	}
	``
		7	7	7

VII.a	Pointers in C : Pointers are used to store and manage the addresses of dynamically allocated blocks of memory. Such blocks are used to store data objects or arrays of objects. Most structured and object-oriented languages provide an area of memory, called the heap or free store, from which objects are dynamically allocated. A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address. The general form of a pointer variable declaration is – type *var-name; Here, type is the pointer's base type; it must be a valid C data type and var-name is the name of the pointer variable. The asterisk * used to declare a pointer is the same asterisk used for multiplication. However, in this statement the asterisk is being used to designate a variable as a pointer. Take a look at some of the valid pointer declarations int *ip; - pointer to an integer double *dp; - pointer to a double float *fp; - pointer to a float char *ch - pointer to a character	8	8	8

b.	#include <stdio.h>
	main()
	{
	int first, second, third, *p, *q, *r,sum,product;
	float av
	printf("Enter three integers to add\n");
	scanf("%d%d %d", &first, &second,&third);
	
	p = &first;
	q = &second;
	r=&third
	sum = *p + *q+*r;
	product = *p * *q * *r;
	av= sum/3;
	printf("Sum %d \n product %d \n Average %f \n",sum,product,av);
	
	}
		7	7	7

VIII.a	String functions in C: Function Work of Function strlen() Calculates the length of string strcpy() Copies a string to another string strcat() Concatenates(joins) two strings strcmp() Compares two string (Briefly Expalin any 4 with example)	4 * 2	8	8

b.	#include <stdio.h>
	#include <string.h>
	
	int main(){
	  char string1[20];
	  int i, length;
	  int flag = 0;
	
	  printf("Enter a string:");
	  scanf("%s", string1);
	
	  length = strlen(string1);
	
	  for(i=0;i < length ;i++){
	  if(string1[i] != string1[length-i-1]){
	  flag = 1;
	  break;
	  }
	}
	
	  if (flag) {
	  printf("%s is not a palindrome", string1);
	  }
	  else {
	  printf("%s is a palindrome", string1);
	  }
	  return 0;
	}
		7	7	7

IX.a	Just like variables, array can also be passed to a function as an argument . In this guide, we will learn how to pass the array to a function using call by value and call by reference methods.
	The array name itself is the address of first element of that array. For example if array name is arr then you can say that arr is equivalent to the &arr[0]
	#include <stdio.h>
	Eg. void display(int age)
	{
	  printf("%d", age);
	}
	
	int main()
	{
	  int ageArray[] = {2, 3, 4};
	  display(ageArray[2]); //Passing array element ageArray[2]
	  return 0;
	}
	(Briefly explain)
		8	8	8

b.	/* Write a program to find Sum of digits of a Number using Function */
	
	#include<stdio.h>
	#include<conio.h>
	void main()
	{
	int s;
	long n;
	clrscr();
	printf("Enter a number ");
	scanf("%ld",&n);
	s=sumdig(n)
	printf("\n sum of digits of %ld=%d",n,s);
	getch();
	}
	sumdig(long num)
	{
	int d,sum=0;
	while(num>0)
	{
	d=num%10;
	num=num/10;
	sum+=d;
	}
	return(sum);
	}
		7	7	7

X.a	A recursive function is a function that calls itself during its execution. ... The function Count() above uses recursion to count from any number between 1 and 9, to the number 10. Recursive function, in logic and mathematics, a type of function or expression predicating some concept or property of one or more variables, which is specified by a procedure that yields values or instances of that function by repeatedly applying a given relation or routine operation to known values of the function. (Explain with an example)	**8**	**8**	**8**

b.	#include <stdio.h>
	long int multiplyNumbers(int n);
	
	int main()
	{
	int n;
	printf("Enter a positive integer: ");
	scanf("%d", &n);
	printf("Factorial of %d = %ld", n, multiplyNumbers(n));
	return 0;
	}
	long int multiplyNumbers(int n)
	{
	if (n >= 1)
	return n*multiplyNumbers(n-1);
	else
	return 1;
	}
		7	7	7