SCHEME OF VALUATION
(Scoring Indicators)

Revision : 2015		Course code : 4044		
Course Title: Programming in C				
Qst.No	Scoring Indicator	Split up score	Sub Total	Total
	PART A			
I (1)	The variable which are declared before the begining of a function is called global variable.	2		10

I (2)	do
	{
	body of loop;
	}
	while(expression);
		2		

I (3)	An array with more than one subscript is generally called multidimensional array	2		
I (4)	Strlen():Find out the length of a string strrev():It reverses a string	1 1	2	
I (5)	Parameter received by the function is called formal parameter	2		
	PART B			

II (1)	**Increment /decrement operators.**
	The increment and decrement operators are ++ and --.
	The operator ++ adds one to its operand where as the operator — subtracts one from the operand.
	M=5;
	M --; //decrement operator
	The value of m would be 4.
	M=5;
	M++; //increment operator
	Then ,the value of m would be 6.
	Increment/Decrement operators are of two types:
	Prefix increment/decrement operator.
	Postfix increment/decrement operator.
		2
		2
		2
			6	

II (2)	1 **Interger(int)** Integer are whole number.It occupy 2 byte of storage. Eg:int a; **Floating point(float)** Floating point are numbers with decimal point.It occupy 4byte.Eg.float a; **charactor(char)** A single charactor can be defined as a caractor type data.It occupy 1 byte. Eg:char a; **void** void is an empty datatype .It is usually used to specify the type of function.Eg:void main()	2 2 2	6	

II (3)	#include<stdio.h>
	void main()
	{ int n,rev=0,rem;
	printf("Enter an Integer:");
	scant("%d",&n);
	do
	{
	rem=n%10;
	rev=rev*10+rem;
	n=n/10;
	}
	while(n>0);
	printf("Reversed Number =%d",rev);
	}
		2
		
		2
		
		2
			6	30

II (4)	#include <stdio.h>
	void main()
	{char string[50];
	int i, length = 0;
	printf("Enter a string \n");
	gets(string);
	for (i = 0; string[i] != '\0'; i++)
	{ length++;}
	printf("The length of a string is = %d\n", length); }
		2
		
		2
		
		2
			6	

II (5)	Ponter is a variable that store address of another variable.
	Pointer declaration
	syntax:datatype *variable name;
	variable name= &variable;
	Eg:int a=5;
	int *p;
	p=&a;
		2
		
		2
		
		2
			6	

II (6)	#include<stdio.h>
	int add(int num1, int num2);
	int main()
	{ int a, b, sum;
	printf("\nEnter the two numbers : ");
	scanf("%d%d",&a,&b);
	sum =add(a,b);
	printf("Addition of two number is : %d\n",sum);
	return (0);}
	int add(int a,int b)
	{ int add;
	add =a + b;
	return (add);}
		2
		
		2
		
		2
			6	

II (7)	Call by value ➢ Value of actual arguments are passing tothe formal parameter ➢ Operation is done on formal parameter	3		

	➢ Changes in the formal parameter will not reflect in the actualparameter call by reference ➢ Address are passed ➢ Function operation on address than value ➢ Formal arguments are points to the actual argument.	3	6	
	PART C			
III (a)	**Structure of c programming** Documentation section Link section Definition section Global declaration part main()function section {Declaration part; Executable part;} subfunction section function1 .. function n Give Any one example	5 4	9	15

III (b)	void main()
	{int a,b,c;
	printf("enter the values of a,b and c");
	scanf("%d%d%d",&a,&b,&c);
	if(a>b && a>c)
	printf("%d is greatest of %d %d %d", a,a,b,c);
	else if(b>c)
	printf("%d is greatest of %d %d %d",b,a,b,c);
	else
	printf("%d is gratest of %d %d %d",c,a,b,c);
	}
		2
		2
		2
			6	

IV (a)	Nested if
	It is a multiple alternative decision statement.Nesting means one condition contains another condition.
	Syntax:
	if(condition1)
	{ if(condition2)
	{ Statement1;}
	else
	{ Statement2;}
	}
	else
	{if(condition3)
	{ Statement3 }
	else
	{ Statement4;}
		2
		3
			9	

	} Give an example	4		15

IV (b)	#include<stdio.h>
	void main()
	{int a,b,op;
	printf("1.addition\n2.subtraction\n3.multiplication\n4.division\n");
	printf("enter the values of a & b");
	scanf("%d%d",&a,&b);
	printf("enter your choice : ");
	scanf("%d",&op);
	switch(op)
	{case1:printf("sum=%d",a+b);
	break;
	case 2:printf("difference=%d",a-b);
	break;
	case 3:printf("multiplication =%d",a*b);
	break;
	case 4:printf("Division= %d",a/b);
	break;
	default:printf("EnterYourCorrectChoice.");}}
		2
		2
		2
		2
			6	

V (a)	A count-controlled repetition will exit after running a certain number of times. The count is kept in a variable called an index or counter
	Syntax of for loop:
	
	for (initialization; condition test; increment or decrement)
	{
	Body of the expression;
	}
	
	Give any example
		2
		3
		4
			9	15

V (b)	#include <stdio.h>
	void main()
	{int arr[100],i,size,sum=0,avg=0;
	printf("Enter size of the array: ");
	scanf("%d", &size);
	printf("Enter %d elements in array: ",size);
	for(i=0; i<size; i++)
	{scanf("%d", &arr[i]);
	sum=sum+arr[i];}
	
	printf("Sum : %d\n", sum);}
		2
		2
		2
		2
			6	

VI(a)	An array with two subscript is termed as two dimensional array. **Initialization of 2D array** syntax:datatype arrayname[row size][col size]={value1,value2....}; eg:int a[2][2]={2,3,1,4};	2 2 2	6	15

VI(b)	Main function Read two matrices Find Sum Print the result	1 3 3 2	9	
VII(a)	2.strcat():It appends one string at end of another. eg 3.strcpy():copy the string in to another. eg	1 3 1 3	8	

VII(b)	#include <stdio.h
	void main()
	{ int i, j, a, n, n[100];
	printf("Enter the size of the array \n");
	scanf("%d", &n);
	printf("Enter the numbers \n");
	for (i = 0; i < n; ++i)
	scanf("%d", n+i); for (i = 0; i < n; ++i)
	{ for (j = i + 1; j < n; ++j)
	{ if (*(n+i) > *n+j)
	{ a = *(n+i);
	*(n+i) = *(n+j);
	*(n+j) = a;}}}
	printf("The numbers arranged in ascending order are given below \n");
	for (i = 0; i < n; ++i)
	printf("%d\n", *n+i); }
		2
		2
		2
		1
			7	15

VIII(a)	1. Increment the pointer(++) 2.Decrement the ponter(--) 3.Add a value to the pointer(+) 4.Substract a value from the pointer(-)	2 2 2 2	8	

VIII(b)	#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);
	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;}
		2
		2
		3
			7	15

IX(a)	A programmer creates the function according to their programming requirement.These function also called user defined function. The general format of function is return_type function_name(argument list) {function body;}	2 3	9	

	Any One example	4		15
IX(b)	main()function Read the number call the function function definition Find the product of two number print the product	1 1 1 1 1 1	6	

X(a)	#include<stdio.h>
	viod sum(int *a,int m);
	void main()
	{int a[100],i,limit;
	printf("Enter the limit\n");
	scanf("%d",&limit);
	printf("Enter the Elements\n");
	for(i=0;i<limit;i++)
	scanf("%d",a+i);
	sum(a,limit)};
	void sum(int *a,int n)
	{int i,sum=0;
	for(i=0;i<n;i++)
	{sum+=*(a+i);}
	printf("Sum =%d",sum);
	}
		2
		2
		2
		1
		2
			9	15

X(b)	A function call itself .It is called recursion
	eg.#include <stdio.h>
	int fact(int);
	int main()
	{int num;
	printf(Enter Any number");
	scanf("%d",&num);
	factorial=fact(num);
	printf("Factorial of a number %d",factorial);
	return 0;}
	int fact(num)
	{if (num==0)
	return 1;
	else
	return num*fact(num-1);}
		2
		4
			6	