V2V EDTECH LLP

Online Coaching at an Affordable Price.

OUR SERVICES:

- Diploma in All Branches, All Subjects
- Degree in All Branches, All Subjects
- BSCIT / CS
- Professional Courses

+91 93260 50669

v2vedtech.com

V2V EdTech LLP

v2vedtech

MODEL ANSWER

SUMMER– 19 EXAMINATION

Subject Title: C' Programming Language **Subject Code: 22218**

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for anyequivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgment on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q.N.	Answer	Marking Scheme
Q.1		**Attempt any Five :**	**10 M**
	a)	**List any four relational operators with their use.**	**2M**
	Ans:	**Relational operators are use in conditional statements.** **Operator** | **Use** == | equal to != | Not equal to	**Each operator with its use ½ M**

		<	less than
		>	Greater than
		<=	Less than equal to
		>=	Greater than equal to

(Any 4)

b)	**Give syntax of switch-case statement.**	**2M**

Ans:	**Syntax:**
	switch(variable)
	{
	case value1:
	statements
	break;
	case value2:
	statements;
	break;
	.
	.
	.
	default:
	statements;
	break;
	}
		Correct syntax 2M

c)	**Give syntax of for loop.**	**2M**

Ans:	**Syntax:**
	for(initialization; condition; increment/decrement)
	{
	Statements;
	}
		Correct syntax 2M

d)	**State any two differences between call by value and call by reference.**	**2M**

Ans:

Sr. No.	Call by value	Call by reference
1	A copy of actual arguments (value) is passed to respective formal arguments.	Address of actual arguments is passed to formal arguments.
2	Actual arguments will remain safe, they cannot be modified	Alteration to actual arguments is possible within from called

Any two differences 1M each

			accidentally.	function; therefore the code must handle arguments carefully else you get unexpected results.
		3	Address of the actual and formal arguments are different	Address of the actual and formal arguments are the same
		4	Changes made inside the function are not reflected in other functions	Changes made in the function are reflected outside also.
	e)	**Define pointer and state any two uses of pointer.**		**2M**
	Ans:	**Definition**: A pointer is a variable that stores memory address of another variable which is of similar data type. **Uses of pointer:-** 1. Pointers are used for dynamic memory management. 2. Pointers permit references to functions and thereby facilitating passing of functions as arguments to other functions. 3. They can be used to return multiple values from a function via function arguments.		**1M** **any two correct uses 1/2M each)**
	f)	**State the use of ✱, *, & symbols used in pointers.**		**2M**
	Ans:	*** operator:-** It is used to declare a pointer variable. **Example:** int *ptr; The above statement declares ptr as an integer pointer variable. **OR** It is also used as value at operator i.e. it reads the value from the address stored in pointer variable. **Example:** printf("%d", *ptr); The above statement displays value present at the address stored in ptr variable. **& operator:-** It is used to retrieve address of a variable from memory. **Example:** int *ptr,a; ptr=&a;		**1M** **1M**

		The above statement stores the address of variable a in the pointer variable ptr.	
	g)	**Define structure.**	**2M**
	Ans:	**Definition:** A structure is a collection of one or more variables of same or different data types grouped together under a single name.	**2M**
Q 2		**Attempt any Three :**	**12M**
	a)	**Explain any four data types used in C with example.**	**4M**
	Ans:	**Data type in C are:** short int/signed short int/**int**/signed int/unsigned int/**long int**/signed long int/unsigned long int/**char**/**float**/**double**/long double/**void** **Integer data type:** • **short int/signed short int:** It is used to declare integer type variable. It occupy 8 bits(1 byte) memory size to store data such as 10,20,etc. Example: short int number; • **int/signed int/unsigned int**:It is used to declare integer type variable. It occupy 16 bits (2 bytes) memory size to store data such as 100,200,etc. Example:int rollno; • **long int/signed long int/unsigned long int:** It is used to declare integer type variable. It occupy 32 bits (4 bytes) memory size to store data such as mobile number. Example: long int contactno; **Character data type:** • **char/signed char/unsigned char:** It is used to declare character type variable. It occupy 8 bits(1 byte) memory size to store data such as 'a','b','%',etc. Example: char ch; **Floating point data type:** • **float:** It is used to declare floating point type variable. It occupy 32 bits (4 bytes) memory size to store data such as 1.1,2.2,etc. Example: float percentage; • **double:** It is used to declare floating point type variable. It occupy 64 bits (8 bytes) memory size to store data such as 10.1,22.2,etc. Example: double percentage;	**Explanation four data type with example 1M each**

		• **long double:** It is used to declare floating point type variable. It occupy 80 bits (10 bytes) memory size to store data such as 11.11,21.2,etc.
		Example: long double percentage;
		void data type:
		void data type has no values. When a function does not return any value then the return type of function is specified with void data type.
		example:
		void add( )
		{
		Statements;
		}	

	b)	**Explain nested if-else statement with syntax and example.**	**4M**

	Ans:	**Definition:**
		if...else statement used inside if statement used in a program is called as nested
		if...else statement. When series of decisions are involved in a program we can use
		nested if...else statement.
		Syntax :
		if(test condition1)
		{
		if(test condition2)
		{
		statement-1;
		}
		else
		{
		statement-2;
		}
		}
		else
		{
		statement-3;
		}
		statement-x;
		If test condition-1 is true, then condition-2 is checked. If condition-2 is true, then statement-1 is evaluated. If condition-2 is false then statement-2 is evaluated and then control is transferred to the statement-x. If condition-1 is false then control passes to statemtn-3 and it is executed .Then control passes to statement-x
			2M
			2M

		example:
		
		int num=75;
		if(num<100) ----------Condition 1
		{
		if(num<50) ----------Condition 2
		{
		printf("Number is less than 50"); ----Statement 1
		}
		else
		{
		printf("Number is greater than 50 but less than 100");--- statement 2
		}
		}
		else
		{
		printf("Number is greater than 100");---statement 3
		}
		In the above example, variable num is initialized to value 75.In condition 1 num is Compared with 100 and the condition evaluates to true. So control passes to condition 2.In condition 2 num is greater than 50 so condition is false. Control passes to statement 2 and output is " Number is greater than 50 but less than 100". Then control comes out of nested if...else statement.
			1M

	c)	**Explain Array. State two advantages of array.**	**4M**
	Ans:	**Explanation:** An array is a sequenced collection of similar type of data. Values in an array are stored in Continuous memory locations. All data values stored in an array share a common name. To access individual value from an array, index variable is used. An index variable starts with 0_{th} position. First value in an array is stored at 0^{th} position and last value is stored in size-1 index position. For example, in an array of 5 elements, first value is stored at 0^{th} position and last value is stored at 4^{th} index position. **Syntax** to declare an array: data type arr_name[size] ; In the above syntax, • **data type** specify type of data that can be stored in an array. • **arr_name** specify name of array. • **size** specify number of values that can be stored inside an array.	**3M**

		Example: int arr[5] = {10, 20, 5, 3, 55}; In the above example, an array variable arr is declared and initialized with integer values with the size 5. arr[0] arr[1] arr[2] arr[3] arr[4] 10 20 5 3 55 **Advantages of an array:** • Array represents multiple data items of similar type using single name. • Adding and removing data element at any index position is possible.	**½ Each** **(Any 2)**
	d)	**List any 2 string functions. Give syntax and use of each function.**	**4M**
	Ans:	**String functions:** **1. strcat ( )** **2. strcmp( )** **3. strcpy ( )** **4. strlen ( )** **5. strlwr( )** **6. strupr( )** **1. strcat ( ):-** This string function is used to join two strings together**.** **Syntax:** strcat (string1, string2); string1 and string2 are character arrays. When the function strcat ( ) is executed, string2 is appended to string1 i.e. contents of string2 are added at the end of string1. **2. strcmp( ):-** This string function is used to compare the contents of two strings. It returns 0 if both string are equal. Otherwise it returns the numerical difference between the ascii values of the first non matching pair of characters. **Syntax:** strcmp(string1,string2); string1 and string2 are character arrays. **3. strcpy( ):-** This string function is used to copy the content of one string to the other string. **Syntax:** strcpy(string1,string2);	**List 1M** **Syntax ½ Each** **Use 1M Each**

		string1 and string2 are character arrays. When strcpy( ) function executes the contents of string2 are copied into string1. **4. strlen( ):-** This string function is used to count and return number of characters stored in a string. **Syntax:** variable_name=strlen(string); string is a character array.variable_name is an integer variable that stores the value of the length of the string return by strlen( ) function. **5. strlwr( ):-** This string function is used to convert a given string into lower case letters. **Syntax:** strlwr(string); string is a character array. **6. strupr( ):-** This string function is used to convert a given string into upper case letters. **Syntax:** strupr(string); string is a character array.	
Q.3		**Attempt any Three:**	**12M**
	a)	**Enlist any four bitwise operators used in C and give example of each.**	**4M**
	Ans:	**Bitwise operators:** | – Bitwise OR & – Bitwise AND ~ – One‟s complement ^ – Bitwise XOR << – left shift >> – right shift **Description:** **Bitwise OR – |** It takes 2 bit patterns, and performs OR operations on each pair of corresponding bits. The following example will explain it. 1010 1100 --------	**List 2M** **example of any four bitwise operator**

		OR 1110
		
		Bitwise AND – &
		It takes 2 bit patterns, and perform AND operations with it.
		1010
		1100
		------- 	
		AND 1000
		
		The Bitwise AND will take pair of bits from each position, and if only both the bit is 1, the result on that position will be 1.
		Bitwise AND is used to Turn-Off bits.
		------- 	
		Bitwise NOT:
		One‟s complement operator (Bitwise NOT) is used to convert each “1-bit to 0-bit” and “0-bit to 1-bit”, in the given binary pattern. It is a unary operator i.e. it takes only one operand.
		1001
		NOT 0110
		------- 	
		Bitwise XOR ^
		
		Bitwise XOR ^, takes 2 bit patterns and perform XOR operation with it.
		
		0101
		
		0110
		
		------ 	
		
		XOR 0011
		
		------ 	
		
		Left shift Operator – <<
		The left shift operator will shift the bits towards left for the given number of times.
		int a=2<<1;
		
		Right shift Operator – >>
		The right shift operator will shift the bits towards right for the given number of times.
		int a=8>>1;
			1/2 M each

	b)	**Explain Pointer Arithmetic.**	**4M**
	Ans:	The pointer arithmetic is done as per the data type of the pointer. The basic operations on pointers are **Increment:** It is used to increment the pointer. Each time a pointer is incremented, it points to the next location with respect to memory size . Example, If ptr is an integer pointer stored at address 1000,then ptr++ shows 1002 as incremented location for an int. It increments by two locations as it requires two bytes storage. **Decrement:** It is used to decrement the pointer. Each time a pointer is decremented, it points to the previous location with respect to memory size. Example, If the current position of pointer is 1002, then decrement operation ptr-- results in the pointer pointing to the location 1000 in case of integer pointer as it require two bytes storage. **Addition:** When addition operation is performed on pointer, it gives the location incremented by the added value according to data type. Eg: If ptr is an integer pointer stored at address 1000, Then ptr+2 shows 1000+(2*2) = 1004 as incremented location for an int. **Subtraction:** When subtraction operation is performed on the pointer variable, it gives the location decremented by the subtracted value according to data type. Eg: If ptr is an integer pointer stored at address 1004, Then ptr-2 shows 1004-(2*2) = 1000 as decremented location for an int.	**Arithmetic operation 1M Each**
	c)	**Explain meaning of following statement with reference to Pointer:** **int var = 50;** **int *p1, *p2;**	**4M**

		P1= & var; **P2= p1;**	
	Ans:	**int var = 50;** It is declaration and initialization of integer variable var with value 50. **int *p1, *p2;** It is declaration of integer pointer p1 and integer pointer p2. **P1= & var;** Address of var is assigned to variable P1. **P2= p1;** Value of p1 is assigned to P2.	**Correct meaning of each statement 1M**
	d)	**Explain declaration of structure with example.**	**4M**

	Ans:	**Structure:** A structure is a collection of one or more variables of same or different data types grouped together under a single name.
		struct structure_name
		{
		Data_type variable 1;
		Data_type variable 2;
		.
		.
		.
		Data_type variable n;
		}variable_name;
		
		• Structure variable is used to access members of structure inside main function with dot operator.
		• Variables of structure can be declared as:
		Variable of structure can be declared at the end of structure declaration before semi colon or inside the main function.
		
		struct book b; //for a single book
		struct book b[5]; //to store data of 5 books
		
		Example:
		struct book
		{
			2M
			
			Any Example 2M

		char tit[20]; char auth[20]; int price; }b1;	
Q.4	**A)**	**Attempt any THREE :**	**12 M**
	a)	**Write a C program to accept two integer numbers from user and print the result of addition and subtraction.**	**4M**

	Ans:	#include<stdio.h>
		#include<conio.h>
		void main()
		{
		int a,b,add,sub;
		clrscr();
		printf("Enter value for a and b:");
		scanf("%d%d",&a,&b);
		add=a + b;
		sub=a - b;
		printf("\nAddition of a and b=%d\n",add);
		printf("\nSubtraction of a and b=%d",sub);
		getch();
		}
			Correct Logic 2M
			
			Correct syntax 2M

	b)	**Write a C program to check whether given number is positive or negative and display message accordingly.**	**4M**

	Ans:	#include <stdio.h>
		#include <conio.h>
		void main()
		{
		int num;
		clrscr();
		printf("\n Enter number: ");
		scanf("%d",&num);
		if(num > = 0)
			Correct Logic 2M
			
			Correct syntax 2M

		printf("\n %d is positive number", num);
		else
		printf("\n %d is negative number", num);
		getch();
		}
		Output:
		Enter number: 4
		4 is positive number
		Enter number: -6
		-6 is negative number	

	c)	**What is an output of following C code:** **#include<stdio.h>** **Void main()** **{** **int a[5]= {10,20,30,40,50};** **printf("output");** **for(i=0; i<3; i++)** **{** **Print("%d", a[i]);** **}** **}**	**4M**
	Ans:	**output** 10 20 30	**Correct output 4M**

	d)	**Declare a structure book having elements as book_number, book_title, book_price and also declare array of structure taking input of 10 books using C programming language.**	**4M**

	Ans:	#include<stdio.h>
		#include<conio.h>
		void main()
		{
		int i;
		struct book {
		char book_title[50];
		int book_number;
		int book_price;
		}b[10];
		clrscr();
		for(i = 0; i < 10;i++)
		{
		printf("Enter book number, title and price of a book:");
		scanf("%d%s%d",&b[i].book_number,b[i].book_title,&b[i].book_price);
		}
		printf("The details of book are:\nBook_Number \tTitle \tPrice\n");
		for(i = 0; i < 10;i++)
		{
		printf("\tbook number=%d \tbook title=%s \tbook price =%d\n", b[i].book_number,
		b[i].book_title,b[i].book_price);
			Correct Logic 2M
			Correct syntax 2M

		}
		getch();
		}	

Q.5		**Attempt any TWO :**	**12 M**
	a)	**Explain if-else statement using syntax and example.**	**6M**

	Ans:	**Syntax of if-else statement :**
		if (test expression)
		{
		True-block statement (s)
		}
		else
		{
		False-block statement (s)
		} Statement-x;
		
		Explanation :
		1. If-else statement is a decision making statement and is used to control the flow of execution of statements.
		2. It allows the computer to evaluate the expression first and then depending on whether the value of the expression is true or false, it transfers the control to the particular statement block.
		3. If the test expression is true, then true block statement(s) are executed, immediately following the if statement are executed otherwise false block statement(s) are executed.
		
		Example:
		
		#include<stdio.h>
		#include<conio.h>
		void main()
		{
		int num;
		printf("Enter the number");
		scanf("%d",&num)
		if(num>0)
		{
			2M
			
			1M
			
			3M
			
			NOTE: Any other example shall be considered

		printf("Number is positive");
		}
		else
		{
		printf("Number is negative");
		}
		getch();
		}	

	b)	**Write a C program to read string from keyboard and find whether it is palindrome or not.**	**6M**

	Ans:	#include<stdio.h>
		#include<conio.h>
		#include<string.h>
		void main()
		{
		char str1[20],str2[20];
		clrscr();
		printf("Enter String to check if it is palindrome : ");
		scanf("%s",str1);
		strcpy(str2,str1);
		strrev(str2);
		if(strcmp(str1,str2)==0)
		printf("String is a palindrome");
		else
		printf("String is not a paliindrome");
		getch();
		}
		(OR)
		#include <stdio.h>
		#include<conio.h>
		#include <string.h>
		void main(){
		char string1[20];
		int i, length;
		int flag = 0;
		clrscr();
		printf("Enter a string:");
		scanf("%s", string1);
			Correct logic 3M,
			Correct syntax 3M
			NOTE: Any other example shall be considered

		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);
		getch();
		}	

	c)	**Write a program to find length of a string.**	**6M**

	Ans:	#include<stdio.h>
		#include<conio.h>
		#include<string.h>
		void main()
		{
		char str[100];
		int length;
		clrscr();
		printf("Enter a string to calculate it's length\n");
		scanf("%s",str);
		length = strlen(str);
		printf("Length of the string = %d\n", length);
		getch();
		}
		(OR)
		#include <stdio.h>
		#include<conio.h>
		void main()
		{
		char s[30];
		int i;
		clrscr();
			Correct logic 3M
			Correct syntax 3M
			NOTE: Any other example shall be considered

```
        printf("Enter a string: ");
        scanf("%s", s);
        for(i = 0; s[i] != '\0'; ++i);
        printf("Length of string: %d", i);
        getch();
}
```

	Q.6	**Attempt any TWO:**	**12M**
	a)	**Write a program to add two 3 * 3 matrices.**	**6M**
	Ans:		**Correct logic 3M** **Correct syntax 3M** **NOTE: Any other example shall be considered**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[3][3], b[3][3], add[3][3], i, j;
    clrscr();
    printf("Enter values for first matrix: \n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("Enter matrix 1 entry(%d,%d): ",i,j);
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter values for second matrix: \n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("Enter matrix 2 entry(%d,%d): ",i,j);
            scanf("%d",&b[i][j]);
        }
    }
    //Performing addition
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
```

		add[i][j] = a[i][j] + b[i][j];
		}
		}
		printf("Addition matrix is: \n");
		for(i=0;i<3;i++)
		{
		for(j=0;j<3;j++)
		{
		printf("%d\t",add[i][j]);
		}
		printf("\n");
		}
		getch();
		}	

	b)	**Write a program to add two numbers using function.**	**6M**

	Ans:	#include<stdio.h>
		#include<conio.h>
		void add(int, int);
		void main()
		{
		int a, b;
		clrscr();
		printf("Enter two number: ");
		scanf("%d%d",&a,&b);
		add(a,b);
		getch();
		}
		void add(int a, int b)
		{
		printf("Addition of %d and %d is %d",a,b,a+b);
		}
		}
			Correct program 3M
			Correct logic 3M
			NOTE: Any other example shall be considered

	c)	**Write a C program to create structure for student having data members like roll_no. name and marks in 3 subjects and display % of marks as output of program.**	**6M**
	Ans:	#include<stdio.h> #include<conio.h> struct student	**Structure declaration 2M**

		{
		int roll_no;
		char name[20];
		int sm1,sm2,sm3;
		}s;
		void main()
		{
		float percent;
		clrscr();
		printf("Enter student roll no, name, subject1 marks,subject2 marks ,subject3 marks :");
		scanf("%d%s%d%d%d",&s.roll_no,s.name,&s.sm1,&s.sm2,&s.sm3);
		percent=(s.sm1+s.sm2+s.sm3)/3;
		printf("\nRollNo\tName\tPercentage");
		printf("\n%d\t%s\t%.2f",s.roll_no,s.name,percent);
		getch();
		}
			Accept elements 2M
			
			Display answer 2M
			
			NOTE: Any other example shall be considered