	b)	List and explain the uses of Dynamic Memory Allocation Functions. Write a C program to allocate a block of memory using malloc().	L4	CO4	5M
		UNIT-V			
10	a)	Develop a C program to read a text file, convert all the lower case characters into upper case characters and print the file data.	L3	CO4	5M
	b)	Develop a C program to find the factorial of a number using recursion.	L3	CO4	5M
		OR			
11	a)	Explain briefly the scope and lifetime of variables.	L2	CO1	4M
	b)	Explain File accessing functions with an example.	L4	CO4	6M

PVP 23

Code: 23ES1102

I B.Tech - I Semester – Regular/Supplementary Examinations
DECEMBER 2024

INTRODUCTION TO PROGRAMMING
(Common for ALL BRANCHES)

Duration: 3 hours Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level CO – Course Outcome

PART – A

		BL	CO
1.a)	List the differences between Compiler and Interpreter.	L2	CO1
b)	Define Keyword. List out any 5 keywords in C.	L2	CO1
c)	List out any 5 string manipulation functions.	L2	CO1
d)	List the differences between Break and Continue statements.	L2	CO1
e)	List two differences between Structure and Union.	L2	CO1
f)	Compare user defined and library functions in C.	L2	CO1
g)	Expalin the use of the fseek () function.	L2	CO1
h)	Explain the usage of if-else statement with an example.	L2	CO1
i)	Define a pointer-to-pointer.	L2	CO1
j)	Write the logic to transpose a 3x3 matrix.	L2	CO1

PART – B

			BL	CO	Max. Marks
		UNIT-I			
2	a)	Explain Basic Input and Output operations of C language.	L2	CO1	4M
	b)	Develop a C program to find the max of 3 numbers using conditional operators.	L3	CO2	6M
		OR			
3	a)	List different types of operators supported by C and explain the usage of Bitwise operators with an example.	L2	CO1	6M
	b)	Draw a flowchart for calculating simple interest.	L2	CO1	4M
		UNIT-II			
4	a)	Write a C program to find the type of triangle formed by the given sides.	L3	CO2	6M
	b)	Develop a C program to determine whether the given number is zero, positive or negative.	L3	CO2	4M
		OR			
5	a)	An electricity board charges the following rates for the use of electricity: for the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: above 300 units Rs 1 per unit. All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges.	L3	CO2	5M
	b)	Identify the differences between entry and exit controlled loop statements.	L3	CO2	5M
		UNIT-III			
6	a)	Develop a C program to count and print the number of duplicate entries in an integer array.	L3	CO3	6M
	b)	Explain pre-defined string functions strcpy and strrev with an example.	L2	CO3	4M
		OR			
7		Define array. How single dimension and two-dimension arrays are declared and initialized?	L2	CO3	10M
		UNIT-IV			
8	a)	Define pointer? Explain how the pointer variable is declared and initialized.	L2	CO3	4M
	b)	Develop a C program to find the sum and mean of all elements in an array using pointers.	L3	CO3	6M
		OR			
9	a)	Develop a C program to declare structure book having data member as book_name, book_id, book_price. Accept this data for 3 books and display it.	L3	CO3	5M

Scheme of evaluation

PVP 23

Code: 23ES1102

I B.Tech - I Semester – Regular/Supplementary Examinations
DECEMBER 2024

INTRODUCTION TO PROGRAMMING
(Common for ALL BRANCHES)

Duration: 3 hours Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level CO – Course Outcome

Part A

1.a) List the differences between Compiler and Interpreter. (2M)
Any two differences - 2M

1.b) Define Keyword. List out any 5 keywords in C. (2M)
Definition - 1M
Any 5 keywords - 1M

1.c) List out any 5 string manipulation functions. (2M)
Any 5 string manipulation functions – 2M

1.d) List the differences between Break and Continue statements. (2M)
Any two differences - 2M

1.e) List two differences between Structure and Union. (2M)
Any two differences - 2M

1.f) Compare user defined and library functions in C. (2M)
Any two differences - 2M

1.g) Explain the use of the fseek () function. (2M)
Use of fseek() – 1M
Parameters explanation – 1M

1.h) Explain the usage of if-else statement with an example. (2M)
Syntax or Flow chart – 1M
Any example – 1M

1.i) Define a pointer-to-pointer. (2M)
Definition – 2M

1.j) Write the logic to transpose a 3x3 matrix. (2M)
Logic for transpose – 2M

.....

Part B

Unit -1

2. a) Explain Basic Input and Output operations of C language. (4M)

Any one input function with example – 2M

Any one output function with example – 2M

2.b) Develop a C program to find the max of 3 numbers using conditional operators. (6M)

Program – 6M

OR

3. a) List different types of operators supported by C and explain the usage of Bitwise operators with an example. (6M)

Any 4 types of operators – 2M

Bitwise operators, any four - 4M

3. b) Draw a flowchart for calculating simple interest. (4M)

Flowchart – 4 M

.....

Unit -2

4.a) Write a C program to find the type of triangle formed by the given sides. (6M)

Program – 6M

4.b) Develop a C program to determine whether the given number is zero, or positive or negative. (4M)

Program – 4M

OR

5.a) An electricity board charges the following rates for the use of electricity: For the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: above 300 units Rs 1 per unit. All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges. (5M)

Program – 5M

5. b) Identify the differences between entry and exit controlled loop statements. (5M)

Any five differences - 5M

.....

Unit -3

6.a) Develop a C program to count and print the number of duplicate entries in an integer array. (6M)

Program – 6M

6.b) Explain pre-defined string functions strcpy and strrev with an example. (4M)

strcpy explanation and example – 2M

strrev explanation and example – 2M

OR

7) Define array. How single dimension and two-dimension arrays are declared and initialized? (10M)

Array definition – 2M

1D array declaration and initialization – 4M

2D array declaration and initialization – 4M

.....

Unit -4

8.a) Define pointer? Explain how the pointer variable is declared and initialized. (4M)

Pointer definition-1M

Declaration and initialization of pointer variable- 3M

8.b) Develop a C program to find the sum and mean of all elements in an array using pointers. (6M)

Using pointers with arrays – 2M

Logic to find the sum and mean – 4M

OR

9.a) Develop a C program to declare structure book having data member as book_name, Book_id, book_price. Accept this data for 3 books and display it. (5M)

Program – 5M

9.b) List and explain the uses of Dynamic Memory Allocation Functions. Write a C program to allocate a block of memory using malloc(). (5M)

Uses of Dynamic Memory Allocation Functions – 3M

Program using malloc() – 2M

.....

Unit -5

10.a) Develop a C program to read a text file, convert all the lower case characters into upper case characters and print the file data. (5M)

Program – 5M

10.b) Develop a C program to find the factorial of a number using recursion. (5M)

Program – 5 M

OR

11.a) Explain briefly the scope and lifetime of variables. (4M)

scope of variables – 2M

lifetime of variables – 2M

11.b) Explain File accessing functions with an example. (6M)

Any 3 file functions- 2+2+2=6M

.....

Key

PVP 23

Code: 23ES1102

I B.Tech - I Semester – Regular/Supplementary Examinations
DECEMBER 2024

INTRODUCTION TO PROGRAMMING
(Common for ALL BRANCHES)

Duration: 3 hours Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level CO – Course Outcome

Part A

1.a) List the differences between Compiler and Interpreter. (2M)

Any two differences - 2M

No	Compiler	Interpreter
1	Compiler Takes **Entire** program as input	Interpreter Takes **Single** instruction as input.
2	Intermediate Object Code is **Generated**	**No** Intermediate Object Code is **Generated**
3	Conditional Control Statements are Executes **faster**	Conditional Control Statements are Executes **slower**
4	**Memory Requirement : More** (Since Object Code is Generated)	**Memory Requirement** is Less
5	Program need not be **compiled** every time	Every time higher level program is converted into lower level program
6	**Errors** are displayed after **entire program** is checked	**Errors** are displayed for **every instruction** interpreted (if any)

1.b) Define Keyword. List out any 5 keywords in C. (2M)

Definition - 1M
Any 5 keywords - 1M

Definition: Keywords are reserved words in C that have predefined meanings and cannot be used as identifiers.
Examples: int, float, return, if, else, while, break, continue

1.c) List out any 5 string manipulation functions. (2M)

Any 5 string manipulation functions – 2M

strlen(), strcpy(), strcat(), strcmp(), strrev(), strstr(), strncpy(), strncat(), strncmp(),

1.d) List the differences between Break and Continue statements. (2M)

Any two differences - 2M

Aspect	`break` Statement	`continue` Statement
Purpose	Exits from the current loop (or switch) completely.	Skips the current iteration of the loop and continues with the next iteration.
Usage in Loops	Used to terminate a loop (like `for`, `while`, `do-while`).	Used to skip the remaining part of the loop and move to the next iteration.
Usage in Switch	Used to exit a `switch` statement and prevent fall-through.	Not applicable in `switch` statements.
Flow Control	Terminates the loop or `switch` statement entirely.	Resumes control at the next iteration of the loop.

1.e) List two differences between Structure and Union. (2M)

Any two differences - 2M

Aspect	Structure	Union
Memory Allocation	Each member has its own memory location. The total size is the sum of the sizes of all members.	All members share the same memory location. The total size is the size of the largest member.
Access to Members	All members can be accessed simultaneously.	Only one member can be accessed at a time.
Memory Usage	More memory is used because every member gets its own space.	Less memory is used since all members share the same memory space.
Size	The size of a structure is the sum of the sizes of its members.	The size of a union is the size of its largest member.
Initialization	All members can be initialized separately.	Only the first member is initialized, as all members share the same space.

1.f) Compare user defined and library functions in C. (2M)

Any two differences - 2M

Aspect	User-Defined Functions	Library Functions
Definition	Created by the programmer.	Predefined in libraries.
Examples	Functions like `sum()`, `factorial()`.	`printf()`, `scanf()`, `sqrt()`.
Function Declaration	Must declare and define the function before using it.	Functions are declared in header files (e.g., `stdio.h`, `math.h`) and can be used directly.

1.g) Explain the use of the fseek () function. (2M)

Use of fseek() – 1M
Parameters explanation – 1M

fseek() is used to move the file pointer associated with a given file to a specific position.

Syntax:

```
int fseek(FILE *pointer, long int offset, int position);
```

Parameters

- **pointer:** It is the pointer to a FILE object that identifies the stream.
- **offset:** It is the number of bytes to offset from the position
- **position:** It is the position from where the offset is added. Position defines the point with respect to which the file pointer needs to be moved. It has three values:
  - **0 or SEEK_END**: It denotes the end of the file.
  - **1 or SEEK_SET**: It denotes starting of the file.
  - **2 or SEEK_CUR**: It denotes the file pointer's current position.

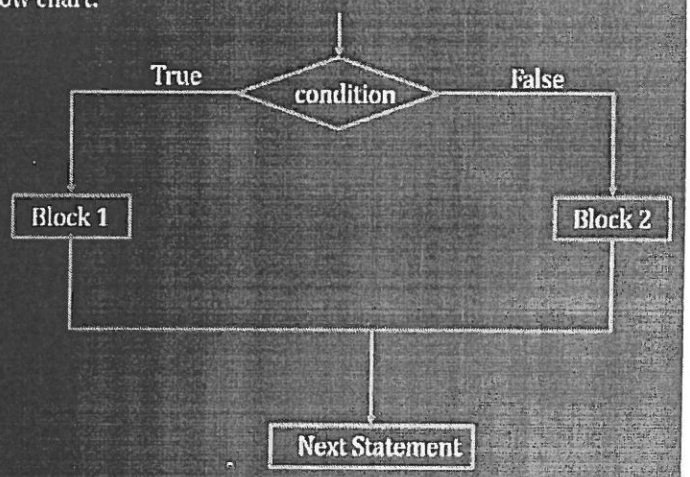
1.h) Explain the usage of if-else statement with an example. (2M)

Syntax or Flow chart – 1M
Any example – 1M

Syntax:

```
if(condition)
{
    statement/s to be executed when
    condition evaluates to true
}
else
{
    statement/s to be executed when
    condition evaluates to false
}
Next statement;
```

Flow chart:



Example:

```
if(a>b)
{
    printf("Largest number is %d",a);
}
else
{
    printf("Largest number is %d",b);
}
printf("Thank you");
```

1.i) Define a pointer-to-pointer. (2M)

Definition – 2M

pointer to pointer is pointer variable which is used to store the address of a pointer

Declaration of pointer to pointer:

Syntax: data-type **ptr_name;

Example: int **p; float **q;

1.j) Write the logic to transpose a 3x3 matrix. (2M)

Logic for transpose – 2M

```
for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        trans[j][i] = mat[i][j];
```

.....

Part B

Unit -1

2. a) Explain Basic Input and Output operations of C language. (4M)

Any one input function with example – 2M
Any one output function with example – 2M

Input statements are used to read the data typed on keyboard into variables. Some of the input statements available in C are scanf(), getchar(), getch(), getche(), gets().

1. scanf(): It is used to read one or more values into given variables from standard input device (keyboard)

Syntax of scanf is

```
scanf("control string", &variable1, &variable2...);
```

Ex:
```
scanf("%d", &x);
scanf("%d%d", &x, &y);
scanf("%d%f%c%s", &a, &b, &c, d);
```

2. getchar(): It is used to read a single character from standard input device into a character variable. After typing the character we have to press the ENTER key

Syntax of getchar is

```
variable-name = getchar();
```

Ex:
```
char x;
x = getchar();
```

3. getch(): It is also used to read a single character from keyboard. But it does not use any buffer, so the entered character is immediately returned without waiting for the ENTER key

Syntax of getch is

```
variable_name = getch();
```

Ex:
```
char x;
x = getch();
```

4. getchel(): Like getch(), this function reads a single character from the keyboard and displays immediately on output screen without waiting for ENTER key

Syntax of getche is

variable-name = getchel();

Ex: char x;
x = getchel();

5. getc(): It is used to read a single character from a given input stream into a character variable.

getc(stdin) is equivalent to getchar()

Ex: char x;
x = getc(stdin);

6 gets(): It is used to read the given string from standard input device into variables.

gets() is mainly used to read a line of text from the keyboard

Syntax of gets is

gets(string_name);

Ex: char a[20];
gets(a);

output statements are used to display data on standard output device (monitor). Some of the output statements are printf, putchar(), puts(), putc()

1. printf: It is used to display the data from the given variables or to display a string constant. It is a library function to display output which works only if #include<stdio.h> is included at the begining of the program

Syntax of printf is

printf("control string", variable1, variable2 ---);

Ex: printf("%d", x);
printf("%d%f", total, avg);

2. puts: It is used to print a given string on a standard output device

Syntax of puts is

puts("String") (or) puts(string-variable-name);

Ex: char a[20] = "Sudhakar";
puts(a);

3. putchar: It is used to print a single character on a standard output device

Syntax of putchar is

putchar(character-variable);

Ex: char a='S';
putchar(a);

4. putc(): It is used to print a single character to a given output stream.
putc(c, stdout) is equivalent to putchar(c).

Ex: char a='S';
putc(a, stdout);

.....

2.b) Develop a C program to find the max of 3 numbers using conditional operators. (6M)

Program – 6M

```
#include <stdio.h>
int main()
{
    int a, b, c,max;
    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);
    max = (a > b) ? (a > c ? a : c) : (b > c ? b : c);
    printf("The maximum number is: %d", max);
    return 0;
}
```

.....

OR

3. a) List different types of operators supported by C and explain the usage of Bitwise operators with an example. (6M)

Any 4 types of operators – 2M
Bitwise operators, any four - 4M

An operator is a symbol that tells the computer to perform certain mathematical or logical manipulations. The operators in C are classified into following types

1. Arithmetic operators 2. Relational operators 3. Logical operators
4. Assignment operators 5. Increment/Decrement operators
6. Conditional operator(Ternary operator)
7. Bitwise operators 8. Special operators

bitwise operators in detail

In Arithmetic-logic unit, mathematical operations are done at bit-level. To perform bit-level operations in c programming, bitwise operators are used

1. Bitwise AND(&): the output of bitwise AND is 1 only if the corresponding bits of two operands is 1.

Ex: 12 = 00001100
25 = 00011001

12 & 25 = 00001000
= 8

x	y	x&y
0	0	0
0	1	0
1	0	0
1	1	1

2. Bitwise OR(|): the output of bitwise OR is 0 only if the corresponding bits of two operands is 0

Ex: 12 = 00001100
25 = 00011001

12 | 25 = 00011101
= 29

x	y	x|y
0	0	0
0	1	1
1	0	1
1	1	1

3. Bitwise XOR(^): the output of bitwise XOR is 1 only if the corresponding bits of two operands are opposite

Ex: 12 = 00001100
25 = 00011001

12^25 = 00010101
= 21

x	y	x^y
0	0	0
0	1	1
1	0	1
1	1	0

4. Right shift(>>) Right shift operator shifts all bits towards right by certain number of specified bits

Ex: 12 = 00001100

12 >> 2 = 00000011

= 3

5. Left shift (<<): Left shift operator shift all bits towards left by certain number of specified bits

Ex: 12 = 00001100

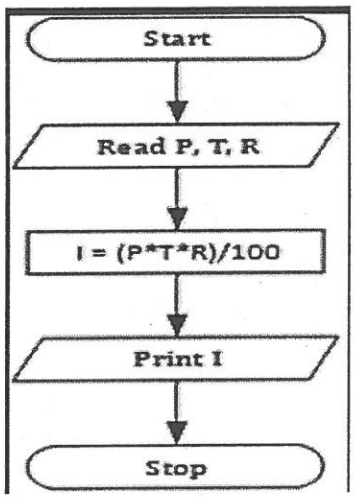
12 << 2 = 00110000

= 48

.....

3. b) Draw a flowchart for calculating simple interest. (4M)

Flowchart – 4 M



.....

Unit -2

4.a) Write a C program to find the type of triangle formed by the given sides. (6M)

Program – 6M

```
#include <stdio.h>
int main() {
    float a, b, c;
    printf("Enter three sides of a triangle: ");
    scanf("%f %f %f", &a, &b, &c);
    if (a == b && b == c)
        printf("Equilateral Triangle");
    else if (a == b || b == c || a == c)
        printf("Isosceles Triangle");
    else
        printf("Scalene Triangle");
    return 0;
}
```

.....

4.b) Develop a C program to determine whether the given number is zero, or positive or negative. (4M)

Program – 4M

```
#include <stdio.h>
int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if (num > 0)
        printf("Positive number");
    else if (num < 0)
        printf("Negative number");
    else
        printf("Zero");
    return 0;
}
```

.....

OR

5.a) An electricity board charges the following rates for the use of electricity: For the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: above 300 units Rs 1 per unit. All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges. (5M)

Program – 5M

```
#include <stdio.h>
int main() {
    char name[50];
    int units;
    float totalAmount, meterCharge = 100.0, surcharge = 0.0;
    printf("Enter your name: ");
    gets(name);
    printf("Enter the number of units consumed: ");
    scanf("%d", &units);
    if (units <= 200) {
        totalAmount = units * 0.80;
    } else if (units <= 300) {
        totalAmount = 200 * 0.80 + (units - 200) * 0.90;
    } else {
        totalAmount = 200 * 0.80 + 100 * 0.90 + (units - 300) * 1.00;
    }
    totalAmount += meterCharge;
    if (totalAmount > 400) {
        surcharge = totalAmount * 0.15;
    }
    totalAmount += surcharge;
    printf("Total bill amount: Rs %.2f\n", totalAmount);
    return 0;
}
```

.....

5. b) Identify the differences between entry and exit controlled loop statements. (5M)

Any five differences - 5M

Aspect	Entry Control Loop	Exit Control Loop
Definition	The loop condition is checked before entering the loop body.	The loop condition is checked after executing the loop body.
Execution Guarantee	The loop body may not execute if the condition is false initially.	The loop body executes at least once, regardless of the condition.
Examples	`for` loop, `while` loop	`do-while` loop
Control Flow	Condition is evaluated before entering the loop.	Condition is evaluated after executing the loop body.
Use Case	Used when the number of iterations is known or condition must be met to enter.	Used when the loop must execute at least once, typically for post-validation scenarios.

.....

Unit -3

6.a) Develop a C program to count and print the number of duplicate entries in an integer array. (6M)

Program – 6M

Note: Any logic to count duplicate entries.

```
#include <stdio.h>
int main() {
    int size, arr[50],count = 0;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }
    for (int i = 0; i < size; i++) {
        if (arr[i] == -1) {
            continue;
        }
        for (int j = i + 1; j < size; j++) {
            if (arr[i] == arr[j]) {
                // Mark arr[j] as visited
                arr[j] = -1;
                count++;
            }
        }
    }
    printf("Number of duplicate entries: %d\n", count);
    return 0;
}
```

.....

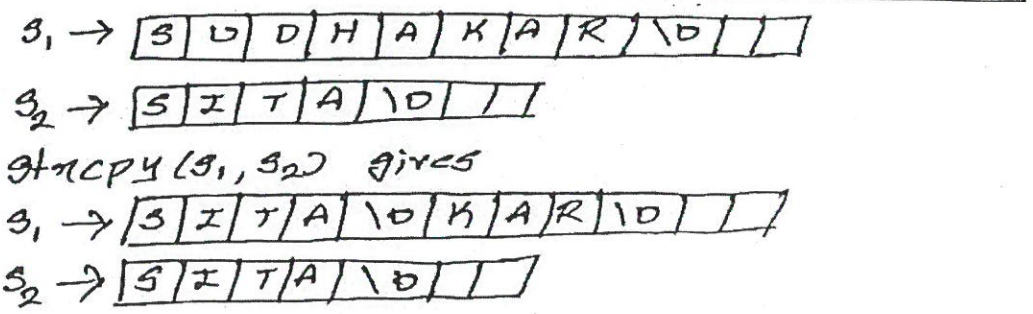
6.b) Explain pre-defined string functions strcpy and strrev with an example. (4M)

strcpy explanation and example – 2M
strrev explanation and example – 2M

strcpy(): this function assigns the contents of one string to another string

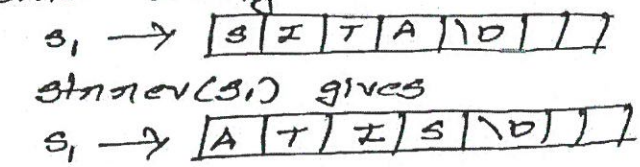
It takes the form `strcpy(s1, s2);`

Ex:



strrev(): this function is used to reverse a string. the reversed string is stored in some string

Ex:



.....

OR

7) Define array. How single dimension and two-dimension arrays are declared and initialized? (10M)

Array definition – 2M
1D array declaration and initialization – 4M
2D array declaration and initialization – 4M

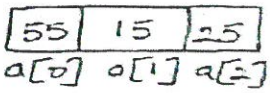
An array is a fixed-size sequenced collection of elements of the same data type

1. 1-D array (one-dimensional array): A list of item can be given one variable name using only one subscript and such a variable is called 1-D array

Declaration: type variable_name[size];

Ex: int a[20]; char s[20];

Initialization: int a[3] = {55, 15, 25}



Accessing: a[0] gives 55
a[1] gives 15

2. 2-D array: these are used to represent tables & matrices of same datatype

Declaration: type array_name[row_size][column_size];

Ex: int a[10][15]

Initialization: int a[2][3] = { {5, 55, 25}, {10, 40, 20} };

Accessing: a[1][2] gives 20

a[0][0] gives 5

	column 0	1	2
row 0	5	55	25
row 1	10	40	20

.....

Unit -4

8.a) Define pointer? Explain how the pointer variable is declared and initialized. (4M)

Pointer definition-1M

Declaration and initialization of pointer variable- 3M

A pointer is a derived datatype in c. pointers contain memory addresses as their values. pointers can be used to access and manipulate data stored in the memory

Declaration of pointer:

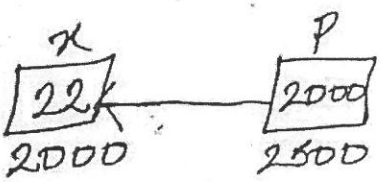
Syntax: data_type *ptr_name;

Example: int *p; float *q;

Initialization of pointer:

Example: int x = 22;

int *p = &x;



.....

8.b) Develop a C program to find the sum and mean of all elements in an array using pointers. (6M)

Using pointers with arrays – 2M

Logic to find the sum and mean – 4M

```
#include <stdio.h>
int main() {
    int n, i,sum=0;
    float mean;
```

```
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n]; // Declare an array
    int *ptr = arr; // Pointer to the array
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", (ptr + i));
    }
    for (i = 0; i < n; i++) {
        sum += *(ptr + i);
    }
    mean = sum / n;
    printf("Sum: %d\n", sum);
    printf("Mean: %.2f\n", mean);
    return 0;
}
```

.....

OR

9.a) Develop a C program to declare structure book having data member as book_name, Book_id, book_price. Accept this data for 3 books and display it. (5M)

Program – 5M

```
#include <stdio.h>
struct Book {
    char book_name[30];
    int book_id;
    float book_price;
};
int main() {
    struct Book b[3];
    for (int i = 0; i < 3; i++) {
        printf("Enter book %d details: \n", i + 1);
        printf("Book Name: ");
        scanf("%s", b[i].book_name);
        printf("Book ID: ");
        scanf("%d", &b[i].book_id);
        printf("Book Price: ");
        scanf("%f", &b[i].book_price);
    }
    printf("\nBook Details:\n");
    for (int i = 0; i < 3; i++) {
        printf("Book Name: %s \n", b[i].book_name);
        printf("Book ID: %d \n",  b[i].book_id);
        printf("Book Price: %.2f\n", b[i].book_price);
    }
    return 0;
}
```

.....

9.b) List and explain the uses of Dynamic Memory Allocation Functions. Write a C program to allocate a block of memory using malloc(). (5M)

Uses of Dynamic Memory Allocation Functions – 3M
Program using malloc – 2M

The process of allocating memory at run time is known as dynamic memory allocation. In c language there are four library functions known as memory management functions

1. malloc 2. calloc 3. free 4. realloc

1. malloc: A block of memory may be allocated using the function malloc. This function reserves a block of memory of specified size and returns a pointer of type void. It takes the form

```
ptr = (cast-type*) malloc (byte-size);
```

Ex:

```
p = (int *) malloc (200);
```

2. calloc(): calloc() function allocates multiple blocks of storage, each of same size, and then sets all bytes to zero it takes the form

```
ptr = (cast-type*) calloc (n, elem-size);
```

This statement allocates contiguous space for n blocks, each of size elem-size bytes

Ex: `p = (int *) calloc (100, 2);`

3. free(): free() function is used to release the memory previously allocated. It takes the form

```
free(ptr);
```

ptr is a pointer to memory block

4. realloc(): realloc() function is used to resize the memory block previously allotted. It takes the form

```
ptr = realloc (ptr, newsize);
```

Ex:

```
p = (int *) malloc (100);
p = (int *) realloc (p, 150);
```

C program to explain the usage of `malloc()` function

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int n,*p,i;
    printf("Enter no. of elements:");
    scanf("%d",&n);
    p=(int *)malloc(n*sizeof(int));
    printf("Enter elements:\n");
    for(i=0;i<n;i++)
        scanf("%d",p+i);
    printf("Elements in array:\n");
    for(i=0;i<n;i++)
        printf("%d ",*(p+i));
    free(p);
}
```

.....

Unit -5

10.a) Develop a C program to read a text file, convert all the lower case characters into upper case characters and print the file data. (5M)

Program – 5M

```
#include<stdlib.h>
#include <stdio.h>
#include <ctype.h>
int main() {
    FILE *f;
    char ch;
    f = fopen("sudhakar.txt", "r");
    if (f== NULL) {
        printf("Could not open file");
        exit(0);
    }
    while ((ch = fgetc(f)) != EOF) {
        putchar(toupper(ch));
    }
    fclose(f);
    return 0;
}
```

.....

10.b) Develop a C program to find the factorial of a number using recursion. (5M)

Program – 5 M

```
#include<stdio.h>
int fact(int n);
void main()
{
    int n, ans;
    printf("Enter a number:");
    scanf("%d",&n);
    ans=fact(n);
    printf("Factorial of %d is %d",n,ans);
}
int fact(int n)
{
    if(n==0)          // Base case
        return(1);
    else
        return(n*fact(n-1));   // Recursive case
}
```

.....

OR

11.a) Explain briefly the scope and lifetime of variables. (4M)

scope of variables – 2M
lifetime of variables – 2M

Scope: The scope of a variable refers that to which different parts of a program have access to the variable, in other words, where the variable is visible

Local Scope: Variables declared inside a function.
Global Scope: Variables declared outside any function, accessible throughout the program.

Lifetime: The life time of variable refers that how long the variable persists in memory, or when the variable's storage is allocated and de-allocated.

Automatic Variables: Exist during the function call and are destroyed when the function exits.
Static Variables: Retain their value between function calls.
Global Variables: Exist throughout the program execution.

.....

11.b) Explain File accessing functions with an example. (6M)

Any 3 file functions- 2+2+2=6M

1. fopen(): fopen() function is used to open a file to perform operations such as reading, writing etc

Syntax: FILE *fp;
fp = fopen("filename","mode");

Here fp- pointer to the datatype FILE
filename - actual file name with full path of file
mode - purpose of opening file. Ex: r, w, a, r+ ..

2. fclose(): fclose() function closes the file that is being pointed by file pointers. It takes the form

fclose(fp);

3. feof(): feof() function is used to find the end of a file. It takes the form

feof(fp)., where fp - file pointer.

feof() returns non-zero if the end of file indicator was detected else returns '0'

4. ftell(): ftell() function takes a file pointer and returns a number of type long, that correspond to the current position. It takes the form n = ftell(fp); n. will give relative offset of the current position

5. fseek(): fseek() function is used to move the file position to a desired location within the file. It takes the form

fseek(file-ptr, offset, position);

Here file_ptr - pointer to the file concerned
offset - number of positions to be moved
position - 0: Beginning of file, 1: current position
2: end of file

6. rewind(): rewind() takes a file pointer and resets the position to the start of the file
It takes the form rewind(fp);

7. fscanf(): fscanf() is used to read set of characters from file. It takes the form

fscanf(fp, "control string", list);

Ex: fscanf(fp, "%d %f", &x, &y);

8. fprintf(): fprintf() is used to write set of characters into file. It takes the form

fprintf(fp, "control string", list);

Ex: fprintf(fp, "%d %f", 22, 20.5);

.....