K. K. Wagh Institute of Engineering Education and Research, Nashik
(An Autonomous Institute from A.Y. 2022-23)

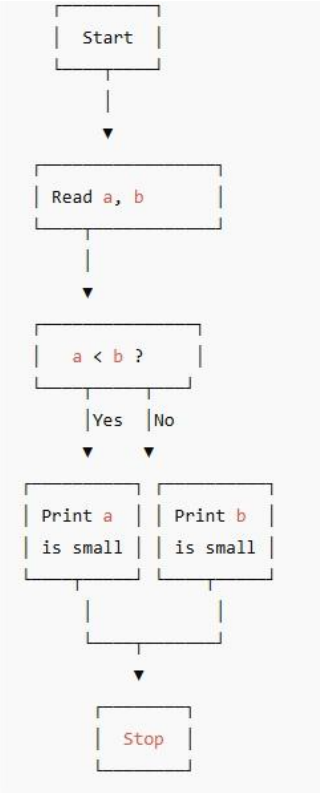
Model Answer
End-Sem Examination-I,Winter 2025

Academic Year: 2025-26	Semester: I
Class: FY	Program: B.Tech
Branch Code: : COM/CSD/AIDS/INT/ETC	Pattern: 2023
Name of Course: Programming in C	Course Code:2300108A

Q1. Illustrate with an algorithm and flowchart how the smallest of two numbers is identified. (6 Marks)

Algorithm

1. Start
2. Read two numbers a and b
3. If a < b
   Print "a is the smallest number"
4. Else
   Print "b is the smallest number"
5. Stop



Q2. Develop / Write a C program to read prices of grocery items from the user and display them in the given format.
(6 Marks)

#include <stdio.h>

```
int main()
{
   float oil, jawar, bajara;
   printf("Enter price of Cooking Oil: ");
   scanf("%f", &oil);
   printf("Enter price of Jawar: ");
   scanf("%f", &jawar);
   printf("Enter price of Bajara: ");
   scanf("%f", &bajara);
   printf("\n****List of grocery items****\n\n");
   printf("Item\t\t\tRate\n\n");
   printf("Cooking Oil\t\tRs. %.2f\n", oil);
   printf("Jawar\t\t\tRs. %.2f\n", jawar);
   printf("Bajara\t\t\tRs. %.2f\n", bajara);
   return 0;
}
```

Q3

a) Develop a C program to read 10 integer values and calculate the sum of only even numbers. (6 Marks)

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

   printf("Enter 10 integer values:\n");
   for(i = 1; i <= 10; i++)
   {
      scanf("%d", &num);
```

```
    if(num % 2 == 0)
    {
        sum = sum + num;
    }
  }
  printf("Sum of even numbers = %d", sum);
  return 0;
}
```

OR

b) Identify the key differences between entry controlled and exit controlled loops with suitable example 6 marks

Ans :

Feature	Entry-Controlled Loop	Exit-Controlled Loop
Condition Check	Condition is checked before executing the loop body	Condition is checked after executing the loop body
Execution	Loop may execute zero times if condition is false initially	Loop executes at least once, even if condition is false initially
Examples in C	for loop, while loop	do–while loop
Use Case	When number of iterations is known or condition must be checked first	When loop must execute at least once regardless of condition
Advantage	Safe: does not execute if condition is false	Ensures minimum one execution
Disadvantage	May skip execution if condition false	Slightly less efficient if condition fails initially

c) Make use of a one-dimensional array **and a for** loop **to store** n integer elements **entered by the user and** apply appropriate logic **to determine and display the** largest and smallest elements **in the array 5 marks**

```
#include <stdio.h>

int main()
{
  int a[100], n, i;
  int largest, smallest;

  printf("Enter number of elements: ");
  scanf("%d", &n);
```

```
   printf("Enter %d integer elements:\n", n);
   for(i = 0; i < n; i++)
   {
      scanf("%d", &a[i]);
   }

   largest = smallest = a[0];

   for(i = 1; i < n; i++)
   {
      if(a[i] > largest)
         largest = a[i];

      if(a[i] < smallest)
         smallest = a[i];
   }
   printf("Largest element = %d\n", largest);
   printf("Smallest element = %d", smallest);
   return 0;
}
```

OR

d) Develop a C program using a two-dimensional array and a nested for **loop to read matrix A of of size 3X3 and display it's transpose 5 marks**

```
#include <stdio.h>
int main()
{
   int a[3][3], i, j;
   printf("Enter elements of 3x3 matrix:\n");
   for(i = 0; i < 3; i++)
   {
      for(j = 0; j < 3; j++)
      {
```

```
            scanf("%d", &a[i][j]);
        }
    }
    printf("\nTranspose of the matrix:\n");
    for(i = 0; i < 3; i++)
    {
        for(j = 0; j < 3; j++)
        {
            printf("%d ", a[j][i]);
        }
        printf("\n");
    }
    return 0;
}
```

e) Develop **a C program that** makes use of a **do-while** loop **to read integer numbers from the user until a** negative number is entered**, and** apply appropriate logic **to calculate and display the** sum of all positive numbers **entered.**
(5marks)

```
#include <stdio.h>

int main()
{
    int num;
    int sum = 0;

    printf("Enter positive integers (enter a negative number to stop):\n");

    do
    {
        scanf("%d", &num);

        if(num >= 0)
        {
            sum = sum + num;
        }

    } while(num >= 0);

    printf("Sum of all positive numbers = %d\n", sum);
```

```
   return 0;
}
```

OR

f) Identify and correct errors in the following code. Rewrite the corrected code and write the expected output. 5 marks

Ans:

```
#include <stdio.h>
int main()
{
   int arr[5], i, sum = 0;

   for(i = 0; i <= 5; i++);        // error   semicolon  added
   {
      scanf("%d", arr[i]);       //  error & missing
      sum = sum + arr[i];
   }

   printf("Sum of elements = %d\n" sum);   // error , missing
   return;                          //  return 0   0 missing
}
```

Q4

a) **Identify the differences between library function and user defined function(6 marks)**

Library Function	User Defined Function
Predefined in C language	Defined by the programmer
Ready to use	Must be written and declared by programmer
Examples: printf(), scanf(), strlen()	Examples: sum(), swap()
Reduces programming effort	Programmer has full control
No need to define the function	Function must be defined before use
May be optimized for speed and efficiency	Efficiency depends on programmer's implementation

Explanation

- Library functions are pre-written functions in C, which can be directly used.
- User-defined functions are custom functions created by the programmer to perform specific tasks.

OR

b) Make use of standard string library functions to compare two strings entered by the user and print whether they are equal or not. Write an appropriate C program 6 marks

Ans :

```
#include <stdio.h>
#include <string.h>

int main()
{
   char str1[100], str2[100];

   printf("Enter first string: ");
   gets(str1);  // Note: For simplicity; in practice use fgets()

   printf("Enter second string: ");
   gets(str2);

   if(strcmp(str1, str2) == 0)
      printf("Strings are equal\n");
   else
      printf("Strings are not equal\n");

   return 0;
}
```

c) Apply appropriate C string library functions **to read a string from the user and** display the length of the string . write an appropriate C program **(5marks)**

Ans :

```
#include <stdio.h>
#include <string.h>

int main()
```

```
{
   char str[100];

   printf("Enter a string: ");
   gets(str); // For simplicity; in practice use fgets()

   printf("Length of the string = %lu\n", strlen(str));

   return 0;
}
```

OR

d) Make use of user defined function **to** swap two numbers

- **using** call by value
- **using** call by reference 5 marks

Ans

Using Call by Value

```
#include <stdio.h>
void swap(int a, int b)
{
   int temp = a;
   a = b;
   b = temp;
   printf("Inside function: a=%d, b=%d\n", a, b);
}
int main()
{
   int x = 10, y = 20;
   swap(x, y);
   printf("Outside function: x=%d, y=%d\n", x, y);
   return 0;
}
```

Using call by reference

```
#include <stdio.h>

void swap(int *a, int *b)
{
   int temp = *a;
   *a = *b;
   *b = temp;
}
int main()
{
   int x = 10, y = 20;
   swap(&x, &y);
   printf("After swapping: x=%d, y=%d\n", x, y);
   return 0;
}
```

e) Develop a program to calculate total stationary bill using a function. Which accepts number of pencils, notebooks , pencil_rate, notebook_rate as input and return total bill amount as an output 5 marks

Ans

```
#include <stdio.h>

float calculateBill(int pencils, int notebooks, float pencil_rate, float notebook_rate)
{
   return (pencils * pencil_rate + notebooks * notebook_rate);
}

int main()
{
   int pencils, notebooks;
   float pencil_rate, notebook_rate, total;

   printf("Enter number of pencils: ");
   scanf("%d", &pencils);
```

```
    printf("Enter rate of pencil: ");
    scanf("%f", &pencil_rate);

    printf("Enter number of notebooks: ");
    scanf("%d", &notebooks);
    printf("Enter rate of notebook: ");
    scanf("%f", &notebook_rate);

    total = calculateBill(pencils, notebooks, pencil_rate, notebook_rate);
    printf("Total bill amount = Rs. %.2f\n", total);

    return 0;
}
```

OR

f) Identify and explain advantages and disadvantages of call by value and call by reference using suitable examples 5 marks

Ans

Call by Value	Call by Reference
Copies actual value into function	Passes address of variable to function
Original value does not change	Original value can be modified
Safer, avoids accidental modification	Efficient for large data, allows modification
Slower for large data (copying required)	Slightly more complex (pointers)
Example: void swap(int a, int b)	Example: void swap(int *a, int *b)

Q5

a) Apply the concepts of arrays and structures to compare their usage with a suitable programming example. (6 marks)

Feature	Array	Structure
Definition	A collection of elements of the same data type, stored in contiguous memory locations.	A user-defined data type that can store multiple variables of different types under a single name.
Data Type	Homogeneous (all elements must be of the same type)	Heterogeneous (members can be of different types)
Access	Accessed using index (e.g., arr[0])	Accessed using member names (e.g., student.name)

Memory	Continuous block of memory	Memory allocated for each member according to its type
Usage	Useful when working with lists of similar items	Useful for representing complex data like student records, employee details
Example	int marks[5];	struct Student { int roll; char name[20]; float marks; };
Operations	Mainly numeric/iterative operations	Organizes related data; can combine operations on different types
Limitation	Cannot store different types together	Slightly more memory overhead compared to simple arrays

OR

b) Develop a program using structure to store details of 10 students such as Name, Class,rollno and display the details for 10 students 5 marks

```
#include <stdio.h>
struct Student {
    char name[20];
    int roll;
    char className[10];
};
int main() {
    struct Student s[10];
    // Reading details
    for(int i=0; i<10; i++) {
        printf("Enter details of student %d\n", i+1);
        printf("Name: ");
        scanf("%s", s[i].name);
        printf("Roll No: ");
        scanf("%d", &s[i].roll);
        printf("Class: ");
        scanf("%s", s[i].className);
    }
    // Displaying details
    printf("\nDetails of 10 students:\n");
    for(int i=0; i<10; i++) {
        printf("Name: %s, Roll No: %d, Class: %s\n", s[i].name, s[i].roll, s[i].className);
```

```
    }
    return 0;
}
```

c) Apply an array of structures **to** store movie details **(movie name and rating). Write an appropriate C program to display** all movie details**..** **(5 marks)**

```
#include <stdio.h>
struct Movie {
    char name[50];
    float rating;
};
int main() {
    struct Movie m[5];  // Array of 5 movies
    int n, i;
    printf("Enter number of movies: ");
    scanf("%d", &n);
    for(i=0; i<n; i++) {
        printf("Enter name of movie %d: ", i+1);
        scanf(" %[^\n]", m[i].name);  // To read string with spaces
        printf("Enter rating of movie %d: ", i+1);
        scanf("%f", &m[i].rating);
    }
    printf("\nMovie Details:\n");
    for(i=0; i<n; i++) {
        printf("Name: %s, Rating: %.1f\n", m[i].name, m[i].rating);
    }
   return 0;
}
```

OR

d) Make use of structures in C to define a structure Employee having members employee ID, employee name, and salary. write a program to read this information for 2 employees from keyboard and print the same on the screen

```
#include <stdio.h>
struct Employee {
    int id;
    char name[20];
    float salary;
};
int main() {
    struct Employee e[2];
    for(int i=0; i<2; i++) {
        printf("Enter details of employee %d\n", i+1);
        printf("ID: ");
        scanf("%d", &e[i].id);
        printf("Name: ");
        scanf("%s", e[i].name);
        printf("Salary: ");
        scanf("%f", &e[i].salary);
    }
    printf("\nEmployee Details:\n");
    for(int i=0; i<2; i++) {
        printf("ID: %d, Name: %s, Salary: %.2f\n", e[i].id, e[i].name, e[i].salary);
    }
    return 0;
}
```

e) Develop a C program **using** structures **to** store passenger details **(*name, age, train number, and seat number*). Display the** passenger list for a given train number**.**
(5marks)

Ans :

```
#include <stdio.h>
#include <string.h>

struct Passenger {
    char name[50];
    int age;
    int trainNumber;
    int seatNumber;
};

int main() {
    struct Passenger p[10]; // Array to store details of 10 passengers
    int n, train;

    printf("Enter number of passengers: ");
    scanf("%d", &n);

    // Read passenger details
    for(int i=0; i<n; i++) {
        printf("\nPassenger %d details:\n", i+1);
        printf("Name: ");
        scanf(" %[^\n]", p[i].name); // Read string with spaces
        printf("Age: ");
        scanf("%d", &p[i].age);
        printf("Train Number: ");
        scanf("%d", &p[i].trainNumber);
        printf("Seat Number: ");
        scanf("%d", &p[i].seatNumber);
    }

    // Display passengers for a specific train number
    printf("\nEnter train number to display passenger list: ");
    scanf("%d", &train);

    printf("\nPassenger list for train %d:\n", train);
    for(int i=0; i<n; i++) {
        if(p[i].trainNumber == train) {
            printf("Name: %s, Age: %d, Seat Number: %d\n", p[i].name, p[i].age,
p[i].seatNumber);
        }
    }
```

```
    return 0;
}
```

OR

f) Develop a C program **using** structures **to** store participant details **(*participant ID, name, and event name*). Display** participants registered for a specific event.

Ans :

```
#include <stdio.h>
#include <string.h>

struct Participant {
    int id;
    char name[50];
    char event[30];
};

int main() {
    struct Participant part[10]; // Array to store 10 participants
    int n;
    char searchEvent[30];

    printf("Enter number of participants: ");
    scanf("%d", &n);

    // Read participant details
    for(int i=0; i<n; i++) {
        printf("\nParticipant %d details:\n", i+1);
        printf("ID: ");
        scanf("%d", &part[i].id);
        printf("Name: ");
        scanf(" %[^\n]", part[i].name);
        printf("Event Name: ");
```

```
    scanf(" %[^\n]", part[i].event);
  }
  // Display participants for a specific event
  printf("\nEnter event name to display participants: ");
  scanf(" %[^\n]", searchEvent);
  printf("\nParticipants registered for event %s:\n", searchEvent);
  for(int i=0; i<n; i++) {
    if(strcmp(part[i].event, searchEvent) == 0) {
      printf("ID: %d, Name: %s\n", part[i].id, part[i].name);
    }
  }

  return 0;
}
```