Type: MCQ

Q1. Which among the following represents the definiteness property of an algorithm (0.5)
BL2 CO1

1. **Action in each step needs to specified unambiguously
2. Must terminate after finite number of steps
3. Operations must be sufficiently basic
4. Must provide zero or more inputs

Q2. Which of the following is true for variable names in C ?(0.5) BL2 CO2

1. **Variable names cannot start with a digit
2. They can contain alphanumeric characters as well as special characters
3. It is not an error to declare a variable to be one of the keywords
4. Variable names are not case sensitive

Q3. Which sections in the C program among the following may be absent when not required (0.5)
BL2 CO1

1. subprogram
2. **main
3. link
4. definition

Q4. Evaluate the following expression BL3 CO2

2>5!=1?5<8 && 8>2?!5?10:20:30:40 (0.5)

1. **20
2. 8
3. 30
4. 40

Q5. Evaluate the expression 10!=10|| 5<4 && 8 (0.5) BL3 CO2

1. 8
2. 1
3. **0
4. -1

Q6. How many times letter x is printed by executing the following C code? BL3 CO2

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

```
void main()
{
int i=5;
while(i)
{ printf("x");
i--; }
printf("x");
}
```
(0.5)

1. 4
2. **6
3. 5
4. 7

Q7. Find the output of (24<<2) (0.5) BL3 CO2

1. 48
2. **96
3. 2400
4. 12

Q8. Identify the output of the following program: BL3 CO3

```
int main()
{
   int a[][] = {{1,2},{3,4}};
   int i, j;
   for (i = 0; i < 2; i++)
      for (j = 0; j < 2; j++)
         printf("%d ", a[i][j]);
   return 0;
}
```
(0.5)

1. 1 2 3 4
2. ** Compiler Error
3. 4 garbage values.
4. 1 2 3 4 0 0

Q9. Predict the output of the following C code snippet BL3 CO2

```
int x = 3;
switch (x)
{
case 2:
printf("ZERO ");
break;
```

default:
printf("HELLO ");
} (0.5)

1. No output
2. **HELLO
3. HELLO ZERO
4. ZERO HELLO

Q10. Identify the output of the following code snippet executed on a 64 bit operating system
BL3 CO3

```
int main() {
   int arr[5] = {1};
   printf("%d\n", sizeof(arr));
   return 0;
} (0.5)
```

1. **20
2. 1
3. 10
4. 4

Type: DES

Q11. Write a C program to check whether the entered number is even/odd. If the number is even print the pattern a and if the number is odd print the pattern b BL4 CO2

Pattern a

```
      *
    * * *
  * * * * *
* * * * * * *
```

Pattern b

```
    1
  2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
```

.(4)

Ans:

If even/odd is checked correctly-------1m
Pattern a ----------1.5m
Pattern b----------1.5m

```
#include <stdio.h>
int main()
{
  int i, space, rows=5, k = 0,n, count = 0, count1 = 0;

  printf("Enter the number ");
  scanf("%d", &n);

  if (n%2==0)
  {

   for (i = 1; i <= rows; ++i, k = 0) {
    for (space = 1; space <= rows - i; ++space) {
      printf("  ");
    }
    while (k != 2 * i - 1) {
      printf("* ");
      ++k;
    }
    printf("\n");
  }}


 else
 {
   for (i = 1; i <= 5; ++i) {
    for (space = 1; space <= 5 - i; ++space) {
      printf("  ");
      ++count;
    }
    while (k != 2 * i - 1) {
      if (count <= rows - 1) {
        printf("%d ", i + k);
        ++count;
      } else {
        ++count1;
        printf("%d ", (i + k - 2 * count1));
      }
      ++k;
    }
    count1 = count = k = 0;
    printf("\n");
  }
 }
  return 0;
}
```

Q12 Write a C program to read a 1D array, remove prime numbers, and display the modified array in ascending order. (4) BL3 CO3

Ans: #include <stdio.h>

```
int main() {
    int arr[100];
    int n;
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);                                   0.5
    }
    printf("Original Array: "); //optional line
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    // Remove prime numbers
    for (int i = 0; i < n; i++) {
        int flag = 0;
        for (int j = 2; j <= arr[i] / 2; j++) {
            if (arr[i] % j == 0) {
                flag = 1;                                       1
                break;
            }
        }
        if (flag == 0) {
            for (int k = i; k < n - 1; k++) {
                arr[k] = arr[k + 1];
            }
            n--;
            i--;                                                1
        }
    }
```

```
   printf("Sorted Array after removing prime numbers: ")
```

1M for sorting (any sorting technique can be used)

```
for(i=0;i<n-1;i++) { //pass                    Bubble sort used here
  for(j=0;j<n-i-1;j++) {
    if(a[j]>a[j+1]){ // comparison
            // interchange
            temp=a[j];
            a[j]=a[j+1];
            a[j+1]=temp;
           }
  }
}


   for (int i = 0; i < n; i++) {
      printf("%d ", arr[i]);
   }                                                              0.5M for printing
   printf("\n");
   return 0;
}
```

Q13. Write a menu driven C program to perform the following tasks using switch statement (**Note**: The program should continue till the user inputs Exit option) BL3 CO2

1. Reverse of a given number
2. Sum of digits
3. Exit. (3)

Ans:

```
#include<stdio.h>
main()
{
int choice, num, I;
int rev=1;
while(1)
{
```

```
 printf("\n\n 1. Reverse");
 printf("\n\n 2. Odd/Even");
 printf("\n\n 3. Exit");                    Menu 0.5M
 printf("\nYour Choice?");
 scanf("%d",&choice);                              0.5M
 switch(choice)                    using switch 0.5M
{
 case 1 : // logic for reversing a number          0.5M
 break;
 case 2: // logic for finding sum of digits        0.5M
 break;
 case 3: exit()                                    0.5M
}
}
}
```

Q14. Develop an algorithm and draw the flowchart to extract digits from a number and display the factorial of each digit. (3) BL3 CO1

Ans:

Step 1: Start

Step 2: Read number n

Step 3: If number not equal to 0 Go to Step 4 , else Go to Step 6

Step 4: digit = number % 10

Step 5: number = number / 10

Step 2: Declare and initialize variables fact = 1 and i = 1

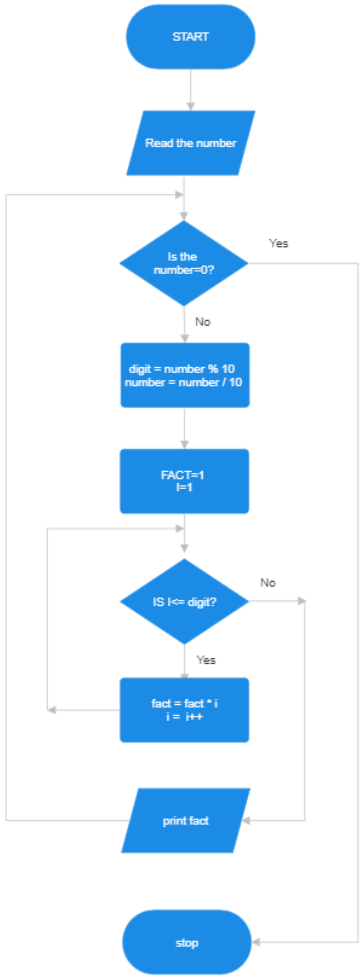
Step 4: Repeat the loop until i<=digit

fact = fact * i

i = i++

Step 5: Print fact and goto step 3

Step 6: Stop



SCHEME

Algorithm-1.5 Mark
Flowchart-1.5 Mark

Q15. Explain the different types of memory involved in the computer system. (3) BL2 CO1

Ans: **Computer memory classifications**

- Main memory-Primary storage
- Secondary memory-Auxiliary storage
- Cache memory

Main memory

- Memory where the data and instructions, currently being executed are stored
  - Located outside CPU
  - High speed
  - Data and instructions stored get erased when the power goes off

- Also referred as **primary** / **temporary** memory
    - Semiconductor memory
    - Measured in terms of megabytes and gigabytes

- RAM stands for Random Access Memory
    - Read and write memory
    - Information typed by the user are stored in this memory
    - Any memory location can be accessed directly without scanning it sequentially (random access memory)
    - During power failure the information stored in it will be erased → volatile memory
- ROM stands for Read Only Memory
    - Permanent memory and non volatile
    - Contents in locations in ROM can not be changed
    - Stores mainly stored program and basic input output system programs

Secondary memory

- Main memory is volatile and limited
    - Hence it is essential for other types of storage devices where programs and data can be stored when they are no longer being processed
- Installed within the computer at the factory or added later as needed
- Non-volatile memory
- Made up of magnetic material
- Stores large amount of information for long time
- Low speed
- Holds programs not currently being executed

Cache memory

- High speed memory placed between CPU and main memory
- Stores data and instructions currently to be executed
- More costlier but less capacity than main memory
- Users can not access this memory

SCHEME

Explanation of Main memory- 1.5 Mark

Explanation of Secondary memory-1 Mark

Explanation of Cache memory-0.5 Mark

Q16. Demonstrate usage of break and continue statements with program(s).
. (3) BL3 CO2

Ans: Any 1 or 2 program codes which demonstrate the usage of break and continue required.

```
#include <stdio. h>
int main() {
int i = 1;
while (i<= 10) {
if (i == 5) {
break;
else printf("%d",i);
}
```

Proper syntax of break within a loop 1M, explanation 0.5M

Here the above program will loop with i values from 1 to 10, printing 1 to 4 but when i becomes 5 it will break out of the loop

```
#include <stdio. h>
int main() {
int i = 1;
while (i<= 10) {
if (i == 5) {
continue;
else printf("%d",i);
}
```

Proper syntax of break within a loop 1M, explanation 0.5M

Here the above program will loop with i values from 1 to 10, printing 1 to 4 but when i becomes 5 it will skip and print 6 to 10 skipping 5.

Q17. Create a 2D array where each row corresponds to a student, and each column represents a subject. Each cell in the array holds the marks of a student of a particular subject. Develop a C program to compute the average marks of each student and the average marks of each subject. (Inputs are read from the user). (3) BL4 CO3

Ans:

```
#include <stdio.h>
int main() {
int marks[10][10];
int NUM_STUDENTS, NUM_SUBJECTS;
printf("Enter the number of students and number of subjects");
scanf("%d%d",NUM_STUDENTS, NUM_SUBJECTS);
```

0.5M

```
    // Input grades for each student and each subject
    printf("Enter marks for %d students for %d subjects:\n", NUM_STUDENTS, NUM_SUBJECTS);
    for (int i = 0; i < NUM_STUDENTS; i++) {
        printf("Enter marks for Student %d:\n", i + 1);
        for (int j = 0; j < NUM_SUBJECTS; j++) {
            printf("Subject %d: ", j + 1);
            scanf("%d", &marks[i][j]);                                  (0.5)

        }
    }

    // Calculate and print average grade for each student
    printf("\nAverage grade for each student:\n");
    for (int i = 0; i < NUM_STUDENTS; i++) {
        int sum = 0;
        for (int j = 0; j < NUM_SUBJECTS; j++) {
            sum += marks[i][j];
        }
        float average = (float)sum / NUM_SUBJECTS;
        printf("Student %d: %.2f\n", i + 1, average);                   (1)
    }

    // Calculate and print average grade for each subject
    printf("\nAverage grade for each subject:\n");
    for (int j = 0; j < NUM_SUBJECTS; j++) {
        int sum = 0;
        for (int i = 0; i < NUM_STUDENTS; i++) {
            sum += marks[i][j];
        }
        float average = (float)sum / NUM_STUDENTS;
        printf("Subject %d: %.2f\n", j + 1, average);                   (1)
    }

    return 0;
}
```

Q18. Write a C program that inputs a string from the user and outputs the string after removing all characters except alphabets

```
Enter a string: HJGFHG6757880@@@54';,jy
Output String: HJGFHGjy
```

. (2) BL3 CO3

Ans:

```
#include <stdio.h>
int main() {
  char line[150];
  printf("Enter a string: ");
```

```
  gets(line);                                   0.5M


  for (int i = 0, j; line[i] != '\0'; ++i) {

    // enter the loop if the character is not an alphabet
    // and not the null character
    while (!(line[i] >= 'a' && line[i] <= 'z') && !(line[i] >= 'A' && line[i] <= 'Z') && !(line[i] == '\0')) {
      for (j = i; line[j] != '\0'; ++j) {

        // if jth element of line is not an alphabet,
        // assign the value of (j+1)th element to the jth element
        line[j] = line[j + 1];
      }
      line[j] = '\0';
    }                                           logic 1M
  }
  printf("Output String: ");                    0.5M
  puts(line);
  return 0;
}
```