#1. Basic of C Programming

Roll Number:
Date of Submission:

Aim:

Write C program to for the following requirements:

i. Prompt two numbers from user along with the option of arithmetic operation. Use appropriate data type to print the results. Run the program continuously until the user chooses an option for exit.
[Note: Use conditional and looping statements]

ii. Prompt 5 numbers from user and store it in an array. Print each value and their category as even/odd number using a pointer. Finally sort the array in ascending order and display the output.
[Note: Use pointers, conditional and looping statements]

iii. Perform the above operation using user defined functions.

Tools Required:

Text editor with C Compiler.

Experiment:

i)

Code

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

// Do Arithmetic Operations for two numbers till the user exits
int main() {
    int running = 1;

    while (running)
    {
        // switch variable
        char var = '+';
        printf("\nAddition – (+)\nSubtraction – (–)\nMultiplication – (*)\nDivision
– (/)\nExit – (x)\n"); // Explaining the Operations
        printf("Enter the Operation You Want to do : ");
        scanf(" %c", &var); // getting the operator with a space to ignore
preceding whitespace
        //printf("%c",var);
        if (var != 'x')
        {
            float x, y;
            printf("Enter the first number: ");
            scanf("%f", &x); // getting the 1st number
            printf("Enter the second number (for division don't give zero): ");
```

```c
        scanf("%f", &y); // getting the second number

        // switch statement
        switch (var)
        {
            case '+': // doing the addition operation
                printf("\nAddition Operation.\n");
                printf("Addition of numbers is: %f", x + y); // print the
result

                break;

            case '-': // doing the subtraction operation
                printf("\nSubtraction Operation.\n");
                printf("Subtraction of numbers is: %f", x - y); // print the
result

                break;

            case '*': // doing the multiplication operation
                printf("\nMultiplication Operation.\n");
                printf("Multiplication of numbers is: %f", x * y); // print the
result

                break;

            case '/': // doing the division operation
                printf("\nDivision Operation.\n");
                if (y == 0) {
                    printf("Division by zero is not possible.");
                } else {
                    printf("Division of numbers is: %f", x / y); // print the
result
                }
                break;
            default:
                printf("Invalid operation. Please try again.");
                break;
        }
        continue;
    }
    else
    {
        running = 0;
        continue; // exit the loop
    }
}
return 0;
}
```

Flow Chart

Result

```
cd "/Users/hari/Amrita/SEM 1/21ES601 – Embedded System Progrmming/" && gcc EXP_1.c –o EXP_1 && "/Users/hari/Amrita/SEM 1/21ES601 – Embed
ded System Progrmming/"EXP_1
/Users/hari/.zshrc:5: command not found: ng
(base) hari@Hariharans-MacBook-Pro 21ES601 – Embedded System Progrmming % cd "/Users/hari/Amrita/SEM 1/21ES601 – Embedded System Progrmm
ing/" && gcc EXP_1.c –o EXP_1 && "/Users/hari/Amrita/SEM 1/21ES601 – Embedded System Progrmming/"EXP_1

Addition – (+)
Subtraction – (–)
Multiplication – (*)
Division – (/)
Exit – (x)
Enter the Operation You Want to do : +
Enter the first number: 8
Enter the second number (for division don't give zero): 9

Addition Operation.
Addition of numbers is: 17.000000
Addition – (+)
Subtraction – (–)
Multiplication – (*)
Division – (/)
Exit – (x)
Enter the Operation You Want to do : /
Enter the first number: 1
Enter the second number (for division don't give zero): 0

Division Operation.
Division by zero is not possible.
Addition – (+)
Subtraction – (–)
Multiplication – (*)
Division – (/)
Exit – (x)
Enter the Operation You Want to do : x
(base) hari@Hariharans-MacBook-Pro 21ES601 – Embedded System Progrmming %
```

ii)

Code

```c
// Question 2
#include <stdio.h>
#include <stdlib.h>

int main() {
    int num;

    // Ask the user for the number of elements in the array
    printf("No of elements in array: ");
    scanf("%d", &num);

    // Declare an array of size 'num'
    int arr[num];

    // Declare a pointer 'a' and point it to the beginning of the array
    int *a = arr;

    // Ask the user to enter the elements of the array separated by spaces
    printf("Enter elements of array with space: ");
    for (int i = 0; i < num; i++) {
        scanf("%d", (a + i));
    }
```

```c
    // Loop through the array to determine if each element is odd or even
    for (int i = 0; i < num; i++) {
        if (*(a + i) & 1) {
            printf("\nOdd number : %d", *(a + i));
        } else {
            printf("\nEven number : %d", *(a + i));
        }
    }


    // Sort the array in ascending order using bubble sort
    for (int i = 0; i < num - 1; i++) {
        for (int j = i + 1; j < num; j++) {
            if (*(a + i) > *(a + j)) {
                // Swap the elements if they are out of order
                *(a+i) = *(a+i) + *(a+j);
                *(a + j) = *(a+i) - *(a + j);
                *(a + i) =  *(a+i) - *(a + j);
            }
        }
    }


    // Print the sorted array in ascending order
    printf("\nAscending Order");
    for (int i = 0; i < num; i++) {
        printf(" -> %d", *(a++));
    }

    return 0;
}
```

Flow Chart

Result

```
  EXP_2.c -o EXP_2 && "/Users/hari/Amrita/SEM 1/21ES601
No of elements in array: 6
Enter elements of array with space: 9 6 3 1 7 2

Odd number : 9
Even number : 6
Odd number : 3
Odd number : 1
Odd number : 7
Even number : 2
 Ascending Order -> 1 -> 2 -> 3 -> 6 -> 7 -> 9
(base) hari@Hariharans-MacBook-Pro C Proagramming %
```

iii)

Code

```c
// Question 3
#include <stdio.h>
#include <stdlib.h>

// Function to check if a number is odd or even
int check_odd_even(int* a) {
    if (*(a) & 1) {
        return 1; // Return 1 if the number is odd
    } else {
        return 0; // Return 0 if the number is even
    }
}

// Function to sort an array in ascending order using bubble sort
int* ascend(int* a, int num) {
    for (int i = 0; i < num - 1; i++) {
        for (int j = i + 1; j < num; j++) {
            if (*(a + i) > *(a + j)) {
                // Swap the elements if they are out of order
                *(a + i) = *(a+i) + *(a+j);
                *(a + j) = *(a+i) - *(a + j);
                *(a + i) =  *(a+i) - *(a + j);
            }
        }
    }
    return a; // Return the sorted array
}
```

```c
int main() {
    int num;

    // Ask the user for the number of elements in the array
    printf("No of elements in array: ");
    scanf("%d", &num);

    // Declare an array of size 'num'
    int arr[num];

    // Declare a pointer 'a' and point it to the beginning of the array
    int *a = arr;

    // Ask the user to enter the elements of the array separated by spaces
    printf("Enter elements of array with space: ");
    for (int i = 0; i < num; i++) {
        scanf("%d", (a + i));
    }

    // Loop through the array to determine if each element is odd or even
    for (int i = 0; i < num; i++) {
        if (check_odd_even((a + i))) {
            printf("\nOdd number : %d", *(a + i));
        } else {
            printf("\nEven number : %d", *(a + i));
        }
    }

    // Sort the array in ascending order
    int *ascending_arr = ascend(arr, num);

    // Print the sorted array in ascending order
    printf("\nAscending Order");
    for (int i = 0; i < num; i++) {
        printf(" -> %d", *(ascending_arr++));
    }

    return 0;
}
```

Flow Chart

<u>Result</u>

```
(base) hari@Hariharans-MacBook-Pro C Proagramming % cd
 tempCodeRunnerFile.c -o tempCodeRunnerFile && "/Users/
ile
No of elements in array: 5
Enter elements of array with space: 8 3 21 12 15

Even number : 8
Odd number : 3
Odd number : 21
Even number : 12
Odd number : 15
Ascending Order -> 3 -> 8 -> 12 -> 15 -> 21%
(base) hari@Hariharans-MacBook-Pro C Proagramming %
```

Inference and Result: