1. ARRAY MENU

AIM:

To create a menu for array that will perform the sorting,reverse,input,display and their sub functions.

ALGORITHM:

· stdio.h is included to perform input and output operations.

· A macro SIZE is defined to fix the size of the array.

· Function prototypes are declared for getting, displaying, reversing, and sorting array elements.

· An integer array of fixed size is declared in the main() function.

· A menu is displayed showing different array operations.

· The program repeatedly asks the user to enter a menu choice until exit is selected.

· GetArray() is called to read new elements and replace the existing array values.

· DisplayArray() is called to print all, odd, or even elements based on user choice.

· ReverseArray() is called to reverse the elements of the array.

· SortArray() is called to sort the array in ascending or descending order.

· The sorted or reversed array is displayed after each operation.

· The program exits when the user selects the exit option.

PROGRAM:

```c
/*
*  Program to Create Menu that will have differnet functionaities of Arrays
*  Author   : MUTHUGANESH S
*  Date     : 16/01/2026
*  Filename : ArrayMenu.c
*  retval   : void
*/

#include <stdio.h>
#define SIZE 10

// Function Prototypes
void GetArray(int Array[],int Size);
void DisplayArray(int Array[],int Size,int Choice);
void ReverseArray(int Array[],int Size);
void SortArray(int Array[],int Size,int Choice);

int main(void){

    int Array[SIZE];
```

```c
    printf("Menu for Array Operations:\n");
    printf("1.Get Array\n");
    printf("2.Display Array\n");
    printf("3.Reverse Array\n");
    printf("4.Sort Array\n");
    printf("5.Exit\n");

    int Choice=0,SubChoice;
    while(Choice != 5){
        printf("\nEnter your choice (1-5): ");
        scanf("%d", &Choice);

        switch(Choice){
            case 1:

                GetArray(Array, SIZE);
                break;

            case 2:

                printf("1. Displaying the array elements:\n2.Odd
Elements\n3.Even Elements\n");
                printf("Enter your choice (1-3): ");
                scanf("%d", &SubChoice);

                DisplayArray(Array, SIZE, SubChoice);

                break;

            case 3:

                ReverseArray(Array, SIZE);
                break;

            case 4:

                printf("1.Ascending\n2.Descending\n");
                printf("Enter your choice (1-2): ");
                scanf("%d", &SubChoice);

                SortArray(Array, SIZE, SubChoice);

                break;

            case 5:

                printf("Exiting the program.\n");
                break;

            default:

                printf("Invalid choice. Please try again.\n");
                break;

        }
```

```c
    }
}

//Function to get array elements from user
void GetArray(int Array[],int Size){

    printf("Enter %d integers to initialize the array:\n", Size);

    for(int i = 0; i < Size; i++){
        scanf("%d", &Array[i]);
    }
}

//Function to display array elements based on choice
void DisplayArray(int Array[],int Size,int Choice){

    if(Choice==1){
        printf("Array elements:\n");
    }
    else if(Choice==2){
        printf("Odd elements:\n");
    }
    else if(Choice==3){
        printf("Even elements:\n");
    }

    for(int i=0;i<Size;i++){

        if(Choice == 1){
            printf("%d ",Array[i]);
        }
        else if(Choice == 2 && Array[i] % 2 != 0){
            printf("%d ",Array[i]);
        }
        else if(Choice == 3 && Array[i] % 2 == 0){
            printf("%d ",Array[i]);
        }
    }
    printf("\n");

}

//Function to reverse the array
void ReverseArray(int Array[],int Size){

    for(int i=0;i<Size/2;i++){

        int temp = Array[i];
        Array[i] = Array[Size - i - 1];
        Array[Size - i - 1] = temp;

    }

    DisplayArray(Array,Size,1);
```

```
    printf("\n");
}

//Function to sort the array based on choice
void SortArray(int Array[],int Size,int Choice){

    if(Choice == 1){

        printf("Ascending order:\n");

        // Ascending order
        for(int i=0;i<Size-1;i++){
            for(int j=0;j<Size-i-1;j++){

                if(Array[j] > Array[j+1]){

                    int temp = Array[j];
                    Array[j] = Array[j+1];
                    Array[j+1] = temp;

                }

            }
        }
    }
    else if(Choice == 2){

        printf("Descending order:\n");

        // Descending order
        for(int i=0;i<Size-1;i++){
            for(int j=0;j<Size-i-1;j++){

                if(Array[j] < Array[j+1]){

                    int temp = Array[j];
                    Array[j] = Array[j+1];
                    Array[j+1] = temp;

                }
            }
        }
    }

    DisplayArray(Array,Size,1);

    printf("\n");
}
```

OUTPUT:

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26200.6899]
(c) Microsoft Corporation. All rights reserved.

M:\training\LabAssesment2>gcc ArrayMenu.c -o Arraymenu.exe

M:\training\LabAssesment2>ArrayMenu
Menu for Array Operations:
1.Get Array
2.Display Array
3.Reverse Array
4.Sort Array
5.Exit

Enter your choice (1-5): 1
Enter 10 integers to initialize the array:
1 2 3 4 5 6 7 8 9 10

Enter your choice (1-5): 2
1. Displaying the array elements:
2.Odd Elements
3.Even Elements
Enter your choice (1-3): 1
Array elements:
1 2 3 4 5 6 7 8 9 10

Enter your choice (1-5): 2
1. Displaying the array elements:
2.Odd Elements
3.Even Elements
Enter your choice (1-3): 2
Odd elements:
1 3 5 7 9

Enter your choice (1-5): 2
1. Displaying the array elements:
2.Odd Elements
3.Even Elements
Enter your choice (1-3): 3
Even elements:
```

```
C:\Windows\System32\cmd.e

Enter your choice (1-5): 3
Array elements:
10 9 8 7 6 5 4 3 2 1


Enter your choice (1-5): 4
1.Ascending
2.Descending
Enter your choice (1-2): 1
Ascending order:
Array elements:
1 2 3 4 5 6 7 8 9 10


Enter your choice (1-5): 4
1.Ascending
2.Descending
Enter your choice (1-2): 2
Descending order:
Array elements:
10 9 8 7 6 5 4 3 2 1


Enter your choice (1-5): 1
Enter 10 integers to initialize the array:
10 11 12 13 1 2 3 2 1 7

Enter your choice (1-5): 2
1. Displaying the array elements:
2.Odd Elements
3.Even Elements
Enter your choice (1-3): 1
Array elements:
10 11 12 13 1 2 3 2 1 7

Enter your choice (1-5): 5
Exiting the program.
```

2. DATES

AIM:

To create a C program to find the number of days between 2 dates using the structure.

ALGORITHM:

· stdio.h is included to perform input and output operations.

· A structure is defined to store day, month, and year values of a date.

· Function prototypes are declared to check leap year, find days in a month, and calculate days between two dates.

· Two structure variables are declared to store the input dates.

· The user enters the first date in day, month, and year format.

· The user enters the second date in day, month, and year format.

· DaysBetweenDates() is called by passing the two date structures.

· The dates are compared and swapped if the first date is later than the second date.

· The number of days is initialized to zero.

· The date is incremented day by day until it reaches the second date.

· Month and year values are updated based on the number of days in each month.

· Leap year condition is checked while calculating days in February.

· The total number of days between the two dates is returned.

· The calculated number of days is displayed on the screen.

· The program ends after displaying the result.

PROGRAM:

```c
/*
* Program count number of days between two dates using structures
* Author   : MUTHUGANESH S
* Date     : 16/01/2026
* Filename : Dates.c
* retval   : void
*/

#include <stdio.h>
struct Date {
    int Day;
    int Month;
    int Year;
```

```c
};
int IsLeapYear(int Year);
int DaysInMonth(int Month, int Year);
int DaysBetweenDates(struct Date Date1, struct Date Date2);

int main(void){
    struct Date Date1;
    struct Date Date2;

    printf("Enter first date (DD MM YYYY): ");
    scanf("%d %d %d", &Date1.Day, &Date1.Month, &Date1.Year);

    printf("Enter second date (DD MM YYYY): ");
    scanf("%d %d %d", &Date2.Day, &Date2.Month, &Date2.Year);

    int Days=DaysBetweenDates(Date1, Date2);
    printf("Number of days between the two dates: %d\n", Days);

    return 0;
}

int IsLeapYear(int Year) {

    if((Year % 4 == 0 && Year % 100 != 0) || (Year % 400 == 0))
        return 1;
    else
        return 0;
}
int DaysInMonth(int Month, int Year) {
    if(Month == 2) {
        if(IsLeapYear(Year)) {
            return 29;
        }
        else {
            return 28;
        }
    }
    else if(Month == 4 || Month == 6 || Month == 9 || Month == 11) {
        return 30;
    }
    else {
        return 31;
    }
}

int DaysBetweenDates(struct Date Date1, struct Date Date2) {
    if (Date1.Year > Date2.Year ||
       (Date1.Year == Date2.Year && Date1.Month > Date2.Month) ||
       (Date1.Year == Date2.Year && Date1.Month == Date2.Month && Date1.Day >
Date2.Day)) {
        struct Date Temp = Date1;
        Date1 = Date2;
        Date2 = Temp;
    }
```

```
    int Days = 0;

    while (Date1.Year < Date2.Year || Date1.Month < Date2.Month || Date1.Day <
Date2.Day) {
        Days++;
        Date1.Day++;
        if (Date1.Day > DaysInMonth(Date1.Month, Date1.Year)) {
            Date1.Day = 1;
            Date1.Month++;
            if (Date1.Month > 12) {
                Date1.Month = 1;
                Date1.Year++;
            }
        }
    }

    return Days;
}
```

OUTPUT: