29/08/2021 DS Lab Assignment - 2

Implement the following programs using pointers and user defined functions only

Q.1. Accept an integer _NUM from the user and display its address using pointer methodologies.

Source Code:

```
#include <stdio.h>
void display (int *ptr){
    printf("Value of NUM : %d :: Address of NUM : %p",
      *ptr, ptr);
}
int main(){
    int NUM, *ptrNUM = &NUM;
    printf("Enter an integer NUM : ");
    scanf("%d", &NUM);
    display(ptrNUM);

    return 0;

}
```

Output :-

```
Enter an integer NUM : 4
Value of NUM : 4 :: Address of NUM : 0x7fffd422d97c
```

Q.2. Implement a pointer to an array to display all the array elements alongside the addresses.

Source Code:

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

void display (int *ptr, int size) {
    for (int i=0 ; i< size ; i++){
    printf ("Value of a[%d] = %d :: Address of a[%d] = %p \n",
    i, *(ptr+i), i, (ptr+i));

    }
}

int main () {
    int *ptr_arr, *arr, n;
    printf ("Enter the size of array : ");
    scanf ("%d", &n);
    arr = (int *)malloc(sizeof(int)*n);

    for (int i =0 ; i<n ; i++) {
        printf("a[%d] : ", i);
        scanf ("%d", &arr[i]);

    }


    ptr_arr = &arr[0];
    display (ptr_arr, n);

    return 0 ;

}
```

Output :-

Enter the size of array : 4

a[0] = 1

a[1] = 2

a[2] = 3

a[3] = 4

Value of a[0] = 1 :: Address of a[0] = 0x55874a361ac0

Value of a[1] = 2 :: Address of a[1] = 0x55874a361ac4

Value of a[2] = 3 :: Address of a[2] = 0x55874a361ac8

Value of a[3] = 4 :: Address of a[3] = 0x55874a361acc

Q.3. Accept an array of integers ARR[] from the user to implement a call by reference and find the sum of the odd numbers and even numbers separately.

Source Code :-

```
#include <stdio.h>
#include <stdlib.h>
void sum (int *ptr, int size) {
    int even = 0, odd = 0;
    for (int i=0 ; i<size ; i++) {
        if (*(ptr+i) %2 == 0)
            even += *(ptr+i);
        else
            odd += *(ptr+i);
    }
    printf("Sum of Odd Values : %d \n", odd);
    printf("Sum of Even Values : %d ", even);
}
int main() {
    int * ARR;
    int n;
    printf("Enter the size of array : ");
    scanf ("%d", &n);
    ARR = (int *) malloc(sizeof(int)*n);
    for (int i=0 ; i<n ; i++) {

        printf("ARR[%d] : ", i);
        scanf("%d", &ARR[i]);
    }
    sum (ARR, n);
    return 0;
}
```

Output :-

```
Enter the size of array : 5
ARR[0] : 3
ARR[1] : -4
ARR[2] : 0
ARR[3] : 2
ARR[4] : -7
Sum of Odd Values : -4
Sum of Even Values : -2
```

0.40. Design a 2D Matrix of size 4x4 and display the elements using pointer to array.

Source Code :

```
# include <stdio.h>
void display (int * p) {
    printf("The elements of the 4x4 Matrix are : \n");
    for (int i = 0 ; i<4 ; i++) {
        for(int j=0 ; j<4 ; j++)
            printf("%d ", *((p+i*4)+j));
        printf("\n");
    }
}

int main (){
    int arr[4][4] , *ptr =& arr[0][0];
    for (int i=0 ; i<4 ; i++) {
        printf("Enter your elements for row %d : ", i+1);
        for (int j=0; j<4 ; j++)
            scanf ("%d", &arr[i][j]);
    }

    display (ptr);

    return 0;
}
```

Output :-

```
Enter your elements For row 1: 1 2 3 4
Enter your elements For row 2: 5 6 7 8
Enter your elements For row 3: 9 10 11 12
Enter your elements For row 4: 13 14 15 16

The elements of the 4x4 matrix are :
1  2  3  4
5  6  7  8
9  10 11 12
13 14 15 16
```