

Clone with git

```
/*  
1.    Write a C program to find the sum of elements of an array  
*/
```

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

```
int main()  
{  
    const int size = 4;  
    int arr[size];  
    int iter = 0;  
    int sum = 0;  
    printf("Enter The Values to Add:\n");  
    for (iter = 0; iter < size; iter++)  
    {  
        scanf("%d", arr + iter);  
    }  
    for (iter = 0; iter < size; iter++)  
    {  
        sum += *(arr + iter);  
    }  
    printf("%d", sum);  
}
```

```
/*  
2.    Write a C program that reads N integer numbers and arrange them in ascending  
order using Bubble sort.  
Hint: Bubble sort, sometimes referred to as sinking sort, is a simple sorting  
algorithm that works by repeatedly stepping through the list to be sorted,  
comparing each pair of adjacent items and swapping them if they are in the wrong  
order. The pass through the list is repeated until no swaps are needed  
, which indicates that the list is sorted. The algorithm gets its name from the way  
smaller elements "bubble" to the top of the list.  
Because it only uses comparisons to operate on elements, it is a comparison sort.  
*/
```

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

```
int main()  
{  
    int i;  
    int n;  
    int flag = 1;  
    int arr[5];  
    for(n = 0; n < 5 ; n++)  
    {  
        scanf("%d" , &arr[n]);  
    }  
    while (flag == 1)  
    {  
        flag = 0;  
        for (i = 0; i < 5 - 1; i++)  
        {  
            if (arr[i] < arr[i + 1])
```

```

        {
            arr[i] = arr[i] + arr[i + 1];
            arr[i + 1] = arr[i] - arr[i + 1];
            arr[i] = arr[i] - arr[i + 1];
            flag = 1;
        }
    }
}
for (i = 0; i < 5; i++)
{
    printf("%d " , arr[i]);
}
}

```

/*
3. Write a C program that takes two Integer arrays as input and inserts the common elements of both arrays into third array
*/

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

```

int main()
{
    int size1, size2;
    printf("Enter Size of the First Array: ");
    scanf("%d" , &size1);
    printf("Enter Size of the Second Array: ");
    scanf("%d" , &size2);
    int arr1[size1] , arr2[size2];
    int arr3[size1>=size2?size1:size2];
    int iter = 0;
    int i = 0 , j = 0;
    printf("Enter Values of the first Array: \n");
    for(i = 0; i < size1; i++)
    {
        scanf("%d" , arr1+i);
    }
    printf("Enter Values of the Second Array:\n");
    for(i = 0; i < size2; i++)
    {
        scanf("%d" , arr2+i);
    }
    for(i=0; i < size1; i++)
    {
        for(j=0; j<size2; j++)
        {
            if(arr1[i] == arr2[j])
            {
                arr3[iter]=arr1[i];
                iter++;
            }
        }
    }
    printf("Common Elements are:\n");
    for(i = 0 ; i < iter; i++)
    {
        printf("%d\t" , *(arr3+i));
    }
}

```

```
}
```

```
/*
```

4. Write a C program to evaluate the square root of a series of numbers and print the results. The process stops when the number 9999 is typed. (Use break and continue)

```
*/
```

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

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    unsigned int n = 0;
```

```
    printf("Enter Values in to get their square root!\n(Type 9999 to exit the loop)\n");
```

```
    while (n != 9999)
```

```
    {
```

```
        scanf("%d", &n);
```

```
        printf("%.2f\n" , sqrt(n));
```

```
    }
```

```
}
```

```
/*
```

5. Write a C program to print Floyd's Triangle:

```
1
```

```
2      3
```

```
4      5      6
```

```
7      8      9      10
```

Hint: Using Nested Looping

```
*/
```

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

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    int i = 0;
```

```
    double null = 0;
```

```
    for (i = 1; i < 1000; i++)
```

```
    {
```

```
        printf("%d  ", i);
```

```
        if (modf(sqrt(8 * i + 1) , &null) == 0)
```

```
        {
```

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

```
        }
```

```
    }
```

```
}
```