

Tutorial 6

```
#include<stdio.h>

int main()
{
    int size,i;

    printf("Enter the size of array: ");
    scanf("%d",&size);
    int arr1[size],arr2[size],vec[size],scalarProd=0;

    printf("Enter %d elements for first array:\n",size);
    for(i=0;i<size;i++)
        scanf("%d",&arr1[i]);

    printf("Enter %d elements for second array:\n",size);
    for(i=0;i<size;i++)
        scanf("%d",&arr2[i]);

    printf("Vector Product:\n");
    for(i=0;i<size;i++)
    {
        vec[i]=arr1[i]*arr2[i];
        printf("%d ",vec[i]);
    }

    for(i=0;i<size;i++)
        scalarProd+=arr1[i]*arr2[i];
}
```

```
printf("\nScalar Product: %d",scalarProd);  
return 0;  
}
```

Tutorial 7

01.

```
#include <stdio.h>  
  
void calSumAndDiff() {  
    int n1, n2;  
    printf("Enter two numbers: ");  
    scanf("%d %d", &n1, &n2);  
    int sum = n1 + n2;  
    int diff = n1 - n2;  
    printf("Sum: %d\n", sum);  
    printf("Difference: %d\n", diff);  
}  
  
int main() {  
    calSumAndDiff();  
    return 0;  
}
```

02.

```
#include <stdio.h>

void calSumAndDiff(int n1, int n2) {
    int sum = n1 + n2;
    int diff = n1 - n2;
    printf("Sum: %d\n", sum);
    printf("Difference: %d\n", diff);
}

int main() {
    int n1, n2;
    printf("Enter two numbers: ");
    scanf("%d %d", &n1, &n2);
    calSumAndDiff(n1, n2);
    return 0;
}
```

03.

```
#include <stdio.h>

int calProduct(int n1, int n2) {
    return n1 * n2;
}

int main() {
    int n1, n2;
```

```
printf("Enter two numbers: ");  
scanf("%d %d", &n1, &n2);  
int product = calProduct(n1, n2);  
printf("Product: %d\n", product);  
return 0;  
}
```

04.

```
#include <stdio.h>  
  
float calQuotient(int n1, int n2) {  
    if (n2 == 0) {  
        printf("Error: Division by zero.\n");  
        return 0;  
    }  
    return (float)n1 / n2;  
}  
  
int main() {  
    int n1, n2;  
    printf("Enter two numbers: ");  
    scanf("%d %d", &n1, &n2);  
    float quotient = calQuotient(n1, n2);  
    printf("Quotient: %.2f\n", quotient);  
    return 0;  
}
```

05.

```
#include <stdio.h>

void displaySum() {
    int n1, n2;
    printf("Enter two numbers: ");
    scanf("%d %d", &n1, &n2);
    int sum = n1 + n2;
    printf("Sum: %d\n", sum);
}

int main() {
    int i;
    for (i = 0; i < 3; i++) {
        displaySum();
    }
    return 0;
}
```

06.

```
#include <stdio.h>

void calandDisplay(int n1, int n2) {
    int sum = n1 + n2;
    int difference = n1 - n2;
    int product = n1 * n2;
```

```

    printf("Sum: %d, Difference: %d, Product: %d\n", sum, difference, product);
}

int main() {
    int n1, n2;
    printf("Enter two numbers: ");
    scanf("%d %d", &n1, &n2);
    calandDisplay(n1, n2);
    return 0;
}

```

07.

```

#include <stdio.h>

double calProduct(int n1, float n2) {
    return n1 * n2;
}

int main() {
    int n1;
    float n2;
    printf("Enter an integer and a float value: ");
    scanf("%d %f", &n1, &n2);
    double product = calProduct(n1, n2);
    printf("Product: %.2lf\n", product);
    return 0;
}

```

08.

Give the function header for each of the following functions:

- a. Function ``hypotenuse`` that takes two double-precision floating-point arguments, ``side1`` and ``side2``, and returns a double-precision floating-point result.

```
double hypotenuse(double side1, double side2);
```

- b. Function ``smallest`` that takes three integers, ``x``, ``y``, ``z``, and returns an integer.

```
int smallest(int x, int y, int z);
```

- c. Function ``instructions`` that does not receive any arguments and does not return a value.

```
void instructions(void);
```

- d. Function ``intToFloat`` that takes an integer argument, ``number``, and returns a floating-point result.

```
float intToFloat(int number);
```