



1.

Analyze the program below. Before you execute it, try to predict what value will be stored in the variable a after calling each of the functions.

#include <stdio.h>

```
void fun1(int b);
void fun2(int a);
void fun3(int *p);
int fun4(int d);
```

int main() {

```
int a = 1;
    int b = 2;
    fun1(a);
    printf("%d n", a);
    fun2(a);
    printf("%d n", a);
    fun3(\&a);
    printf("%d n", a);
    a = fun4(a);
    printf("%d n", a);
    return 0;
void fun1(int b)
    b = 2;
}
void fun2(int a){
    a = 3;
}
void fun3(int *p)
    *p = 4;
```

}

```
int fun4(int d)
{
  d = 5;
  return d;
```

2.

3

Write a program which will make some operations on a set of randomly generated numbers. First write a function generating N random numbers of type double from the range < 0, 1 > and stores them inside an array. The array should be declared local inside the main function. The size of the array should be defined by preprocessor command #define. The array and its actual size should be passed to the function as arguments. Write a function which will print all the values from the array.

3.

Extend the program from the previous point by adding two functions. First should calculate the average value of the array of the generated numbers. Second should calculate the standard deviation.

average: $\bar{s} = \frac{1}{N}(s_1 + s_2 + \ldots + s_N) = \frac{1}{N}\sum_{i=1}^N s_i$, standard deviation: $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (s_i - \bar{s})}$

4.

Add a function which finds maximum from the array of random numbers and returns also the index of the maximum element.

5.

Add a function reversing order of the elements inside the array. Do not use additional array during the reversing operation.