ДИСЦИПЛИНА **Методы верификации и валидации характеристик программного обеспечения**
(полное наименование дисциплины без сокращений)

ИНСТИТУТ **информационных технологий**

КАФЕДРА **математического обеспечения и стандартизации информационных технологий**
(полное наименование кафедры)

ВИД УЧЕБНОГО МАТЕРИАЛА **Материалы для практических/семинарских занятий**
(в соответствии с пп.1-11)

ПРЕПОДАВАТЕЛЬ **Петренко Александр Анатольевич**
(фамилия, имя, отчество)

СЕМЕСТР **3, 2023-2024**
(указать семестр обучения, учебный год)

Инструменты дедуктивной верификации программ

На основе изучения материала лекций по дисциплине «Методы верификации и валидации характеристик программного обеспечения» требуется выполнить следующее.

1. Определите на языке ACSL контракты следующих функций:

a). нахождение НОД двух натуральных чисел;

b). сортировка числового массива.

2. Реализуйте на языке C следующие функции:

a). нахождение суммы элементов числового массива;

b). сортировка числового массива «методом пузырька».

Для указанных функций определите на языке ACSL инварианты циклов.

3. Реализуйте на языке C и докажите с использованием Frama-C корректность следующих функций:

a). целочисленное деление (функция idiv)

b). дихотомический поиск в упорядоченном массиве (функция bsearch)

4. Проверифицируйте с помощью Frama-C ваши реализации следующих функций:

a). нахождение суммы элементов числового массива;

b). сортировка числового массива «методом пузырька».

5. Специфицируйте на языке ACSL и проверифицируйте с помощью Frama-C следующие функции:

a). нахождение максимального делителя натурального числа, отличного от самого числа:

```
int maxDivisor(int n) {
 int m = 1;
 for (int i = n - 1; i > 0; i--) {
```

```
    if (n % i == 0) {
                    if (i > m) {
                                m = i;
                               }
                        }
                                }
 return m;
}
```

b). нахождение максимального простого делителя натурального числа:

```
int maxPrimeFactor(int n) {
 int min = 1;
 do {
      n /= min;
      min = n;
      for (int i = n - 1; i > 1; i--) {
              if (i * i <= n && n % i == 0) {
              min = i;
                                              }
                                        }
   } while (min < n);
 return n;
}
```

c). вычисление целой части квадратного корня целого неотрицательного числа:

```
int isqrt(int n) {
 int a = 0;
 int b = 1;
 int c = 1;
 for(; b <= n; a++) {
                          c += 2;
```

```
                    b += c;
                    }
 return a;
}
```

d). поиск наиболее дешевого варианта с пользой не менее заданной:

```
int minCostForGivenValue(int n, int *cost, int *value, int k) {
 int r = -1;
 for (int i = 0; i < n; i++) {
                    if (value[i] >= k) {
                              if (r == -1) {
                                        r = i;
                              } else if (cost[i] < cost[r]) {
                                                  r = i;
                                        }
                    }
 }
return r;
}
```