

## Практикум №2 (опциональный)

Требуется реализовать алгоритм DPLL на языке С или С++.

Срок сдачи: до 23:59 03 ноября.

К реализации предъявляются следующие требования:

- не используется рекурсия;
- применяется только правило распространения единицы;
- входом является файл с КНФ в формате DIMACS;
- отсутствуют утечки памяти, а также ошибки доступа к памяти;
- реализация должна решать как минимум все задачи из наборов UF и UUF до 150 переменных включительно;
- обеспечивается приемлемая производительность и потребление памяти (время работы программы замеряется на целевой машине с процессором Ryzen 5900x):
  - время решения задачи hanoi4 не превосходит 20с;
  - потребление памяти при решении задачи hanoi4 не превосходит 8МБ (потребление памяти оценивается только для задачи hanoi4);
  - реализация должна решать все задачи из набора uf150-645 с ограничением по времени 800с **в совокупности**;
  - реализация должна решать все задачи из набора uuf150-645 с ограничением по времени 2000с **в совокупности**.

Программа должна принимать в качестве аргумента командной строки путь к файлу с КНФ.

В результате работы программа должна вывести на стандартный поток вывода вердикт:

- строку SAT, если входная КНФ выполнима;
- строку UNSAT в противном случае.

Посторонний вывод программы при проверке не допускается.

Пример запуска программы:

```
$ ./dpll hanoi4.cnf  
SAT
```

Для проверки реализации используются следующие наборы задач:

- <https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html>

Архив в формате zip с решением должен содержать только файлы реализации на верхнем уровне. Дополнительные файлы или каталоги в архиве не допускаются. Имя архива должно содержать фамилию и инициалы автора в транслитерации, например, `ivanov_ii_dpll.zip`, иной формат имени архива не допускается. Архив с решением требуется направить на почту `kotsynyak@ispras.ru`, тема письма: “FV-2025: DPLL Фамилия И.О.”.

Программа должна быть реализована самостоятельно. При обнаружении плагиата оценка за задание аннулируется у всех участников. При выполнении задания допустимо использовать парсер формата DIMACS, поставляемый в приложении к условию задания.

Команда запуска компилятора C++ при проверке задания:

```
g++ -std=c++17 -Wpedantic -Werror -O2
```

Команда запуска компилятора С при проверке задания:

```
gcc -std=c11 -Wpedantic -Werror -O2
```