

# МИФИ, 5 семестр, задачи по C++:

## 03-hello-c

Написать программу Hello, world! в C-style (с помощью `stdio.h` и `printf()`). Важно: строка в сообщении должна быть переведена. Это правило хорошего тона и требование по умолчанию ко всем программам, которые мы будем писать в рамках курса.

## 04-hello-c++

Написать программу Hello, world! в C++-style, т.е. с помощью `iostream` и `std::cout`. Внимание, важное требование! Строка после последнего вывода должна быть переведена. Это элемент хорошего стиля программирования, который становится для нас теперь обязательным к выполнению во всех задачах курса.

## 05-hello-name

Написать в C++-style программу, вводящую с клавиатуры имя и выдающую приветствие. Пример вывода:

```
Введите ваше имя: Саша
Привет, Саша!
```

(Помним про перевод строки в конце)

## 06-str\_copy

Циклы и массивы. Копирование строки. Ввести с клавиатуры строку, скопировать ее в другую строку и вывести ее. Пример вывода:

```
Введите строку: строкаляляля
Копия строки: строкаляляля
```

Нельзя использовать `strcpy()` или `string::copy()`, т.е. копирование должно быть совершено с помощью цикла!

Помним про перевод строки в конце!

## 07-quest\_num

Дихотомический алгоритм угадывания числа. Компьютер просит задумать число от 1 до 100 и начинает его угадывать. Берет допустимый интервал значений, делит его пополам и просит сравнить задуманное число с серединой интервала. Если середина больше -- допустимый интервал становится нижней половиной исходного, если наоборот -- верхней половиной. И так, пока не попадет точно в число. Пользователь играет честно (не обманывает, число не меняет в процессе).

Пример вывода:

```
Задумайте число N от 1 до 100, а я угадаю!
```

```
N больше 50 (1 да, 2 нет, 3 равно)?
```

```
1
```

```
N больше 75 (1 да, 2 нет, 3 равно)?
```

```
1
```

```
N больше 87 (1 да, 2 нет, 3 равно)?
```

3

Вы задумали 87! А я молодец! Количество попыток: 3

## 09-guess\_word

Поле чудес - алгоритм угадывания слова.

Пример:

Введите слово: транспорт

-----  
Введите букву: т

Есть такая буква. Вы молодец!

т \_ \_ \_ \_ \_ т

Введите букву: щ

Нет такой буквы. Попробуйте ещё раз!

Введите букву: р

Есть такая буква. Вы молодец!

т р \_ \_ \_ \_ р т

.....

Введите букву: а

транспорт

Вы угадали слово!

## 11-guess\_word-checking-

К Полю чудес добавляем:

- Проверку, что буква уже называлась. Если да, то выводить сообщение "Эта буква уже была, попробуйте еще раз."
- Возможность угадать слово. Если пользователь вводит более одного символа, трактуем это как попытку угадать слово. Если оно совпадает, пользователь победил. Если нет, надпись "К сожалению, вы не угадали слово. Вы проиграли." и конец.

## 12-factorial

Факториал числа. Пользователь вводит с клавиатуры число, программа выдает его факториал. Ввод проверяется на корректность:

- если пользователь ввел не число, а строку
  - если пользователь ввел отрицательное число
  - если пользователь ввел нецелое число
  - если пользователь ввел число более 100000
- то программа сообщает ему об этом и просит ввести число еще раз

## 13-fibonachi

Числа Фибоначчи. Пользователь вводит число N от 1 до 100 с клавиатуры, программа выдает первые N чисел Фибоначчи одно за другим через пробел. Ввод проверяется на корректность:

- если пользователь ввел не число, а строку
- если пользователь ввел число <1
- если пользователь ввел нецелое число
- если пользователь ввел число больше 100

## 14-hanoi

Задача со звездочкой. С клавиатуры вводится число N. Написать алгоритм (последовательность переключивания) классической (три стержня) ханойской пирамидки [https://ru.wikipedia.org/wiki/%D0%A5%D0%B0%D0%BD%D0%BE%D0%B9%D1%81%D0%BA%D0%B0%D1%8F\\_%D0%B1%D0%B0%D1%88%D0%BD%D1%8F](https://ru.wikipedia.org/wiki/%D0%A5%D0%B0%D0%BD%D0%BE%D0%B9%D1%81%D0%BA%D0%B0%D1%8F_%D0%B1%D0%B0%D1%88%D0%BD%D1%8F) с первого стержня на третий, используя второй как промежуточный. Годится любой алгоритм, главное, чтобы он работал. Я могу попросить вас объяснить и обосновать мне ваш алгоритм, если он будет мне непонятен.

Пример вывода:

Введите N: 3

```
1 -> 3
1 -> 2
3 -> 2
1 -> 3
2 -> 1
2 -> 3
1 -> 3
```

## 15-mach-figures

Написать код, вычисляющий для типов float, double и long double:

- машинный 0
- машинную бесконечность, + и -
- машинное эпсилон
- сумму и разность машинного нуля и машинного эпсилон.

## 16-sort\_swap

Передача параметров по ссылке и сортировка.

- Написать три варианта функции обмена значениями двух целых переменных. Т.е. на входе у нее два целых параметра i1 и i2, а на выходе i1 = i2, а i2=i1. Один из вариантов может быть через временную переменную, два других не должны иметь временной переменной.

- Реализовать ваш любимый метод сортировки массива. Если вы не знаете пузырек, можно пузырек. Если знаете, ищите другой, лучше с рекурсией! Массив заполнить случайными целыми числами. На входе сделать контроль ввода.

Образец вывода программы:

Введите число элементов в массиве: 5

Исходный массив: 4 178 93 -18 6

Отсортированный массив: -18 4 6 93 178

## 17-createClass

Реализовать класс CRational, рациональную дробь, содержащий:

- поля с целыми числителем и знаменателем.
- функции сложения и умножения двух дробей
- вывод дроби в формате "числитель/знаменатель". если дробь смешанная, то "целая часть числитель/знаменатель". если дробь целое число, то "целая часть".
- инициализацию дроби двумя числами (числитель, знаменатель).
- если дробь в результате любой операции становится сократимой, ее надо сократить
- знаменатель не должен быть нулем

## 18-GeomIntersect

Даны два прямоугольника на плоскости, стороны параллельны координатным осям. Все числа вещественные.

- Определить и вывести на экран ответ, пересекаются эти прямоугольники или нет.

Пересечение определяется в математическом смысле, т.е. если есть хотя бы одна общая точка, то да, иначе нет.

- Для этого: придумать и реализовать архитектуру класса для представления прямоугольника. Все данные, в открытии которых нет необходимости, должны быть закрыты (private). В классе должен быть конструктор со списком инициализации, который позволяет создать прямоугольник вызовом вида

`CRectangleClass rectangle1(param1, param2, ...);`

- Все необходимые числа, сколько бы их ни было нужно, вводятся из файла через пробел.

## 19-LIFO-FIFO

Реализовать классы для FIFO (очереди) и LIFO (стека). Классы должны содержать:

- контейнер данных — динамический массив целых чисел с начальной длиной N, управляемый через `new` и `delete`. Реализовать через массив `int*`, контейнеры C++ типа `std::vector<>` не использовать.

- конструктор, инициализирующий объект

- функции `push()` и `pop()` для помещения числа в контейнер и извлечения его из контейнера. помним, что помещаются данные в обеих этих структурах в начало, а извлекаются потом из разных мест — в очереди из конца, в стеке из начала. функция `pop()` возвращает извлеченный объект, удаляя его при этом из контейнера

- функцию для вывода объекта на экран, можно перегрузить для этого `ostream& operator<<`

- предполагаемая реализация: унаследовать их обе от базового класса с массивом и перегрузить отличающиеся функции. второй вариант: абстрактный базовый класс, а сходства и различия положить целиком в реализации.

- деструктор, корректно очищающий память

## 20-CSlice

Реализовать класс `CSlice`, который делает с массивом целых чисел то же, что слайс в Питоне.

Класс содержит:

- конструктор для инициализации со списком инициализации элементов

- парсер строки, который разбирает выражение вида `a[1:-1:5]`, `a[:-1]`, `a[:-5]` (абсолютно любой допустимый в Питоне ввод слайса, включая слайс с пропущенными параметрами) на параметры. Парсер возвращает ошибку, если синтаксис вызова слайса некорректен.

Имеем дело только с ошибками в самом слайсе с точки зрения Питона. Вводу пользователя (нужные скобки, цифры и тому, что он не сядет на клавиатуру) верим.

- функцию, создающую и возвращающую объект — результат действия слайса

- деструктор, корректно очищающий память при удалении объекта.

На входе можно использовать массив `int[]` с указанием длины (т.е. передавать два параметра, указатель и число элементов). Если удобно, можно вместо `int[]` использовать контейнер C++, например, `std::vector<int>`.

## 22-TECPLLOT

Написать код и построить в Tecplot графики следующих функций:

-  $y=\sin(x)$ ,  $0 < x < 2\pi$ , на равномерной сетке в 100 ячеек

-  $y=x*\sin(x)$ ,  $-100 < x < 100$ , на равномерной сетке в 200 ячеек

-  $z=\sin(x)*\sin(y)$ ,  $0 < x, y < 2\pi$ , равномерная сетка в  $100*100$  ячеек

-  $z=F(x,y)$  где  $F(x,y)$  — плотность двумерного стандартного нормального распределения.

В реп кладите кроме кода датасеты в формате Текплот и экспортированные пнг-файлы.

Графики должны быть красивыми и пригодными для публикации в журнал:

- указаны названия осей и переменных
- выбрана опция отрисовки линии, параллельной оси, в противоположной части вычислительной области. Т.е. на выходе график окружается прямоугольником/параллелепипедом, а не открытой с одной стороны областью.
- есть легенда с расшифровками переменных.
- присутствует композиция, т.е. графические элементы расположены на рисунке, уравнивая друг друга визуально.
- в пнг-файле отключены заголовок и границы (редактор пригрозил завернуть картинки и отодвинуть дедлайн выпуска всего тома журнала, если вы этого не сделаете).

Дополнительное задание:

- в последних двух пунктах сделать кроме 3D-графиков двумерные контурные диаграммы
- сделать кроме png векторные картинки с графиками в форматах eps и pdf любым удобным вам способом. Обязательное требование: весь текст в pdf должен быть selectable (выделяться и копироваться), т.е. никаких растров в тексте при экспорте/редактировании. Российский журнал пропустил пнгшки, но редактор западного журнала из Q1 с open access согласен только на векторный формат.

upd.: про стандартное нормальное распределение — здесь мы говорим о независимых по осям случайных величинах, т.е.  $F(x,y) = F_1(x)*F_2(y)$

## 23-LIFO-FIFO-INHERITANCE

Сделать задачу про LIFO и FIFO с помощью наследования от абстрактного базового класса (см.зад.19).

## 24-GraphicEditor

Графический редактор.

Реализовать класс 2D-объект, базовый абстрактный либо интерфейс. Класс должен содержать чистые виртуальные функции move() (передвинуть объект на вектор), show() (нарисовать объект), rotate() (повернуть на заданный угол) и scale() (растянуть/сжать объект). От него унаследовать два класса: круг и квадрат, определить для них эти функции и задать необходимые параметры-данные. В функции main() создать массив из нескольких объектов разного типа, нарисовать их. Затем произвести с тремя из них базовые действия (move, scale, rotate), и снова нарисовать. Рисование объекта происходит в виде выдачи текстового сообщения, например, "Circle, centre (0,100), r=5".

## 25-Exceptions

Исключения C++. В задании по CRational либо CSlice по вашему выбору сделать 6 воспроизводимых проверок с помощью исключений:

- 3 исключения стандартных типа деления на 0 или ошибки типа
  - 3 исключения ваших собственных для обработки ошибок
- Уметь объяснить, почему именно эти ситуации вы выбрали для исключений.