double f\_test(double x, double u); // Функция для тестовой задачи  
  
double anal\_sol\_test(double x,double u0); // Аналитическое решение тестовой задачи  
  
double f\_main\_1(double x, double u); // Функция для основной задачи-1  
  
std::pair<double,double> f\_main\_2(std::pair<double,double> u, double a); // Функция для основной задачи-2  
  
void rK\_step(double (\*f)(double, double),double& x,double& u,double step); // Шаг метода РК4  
  
// Шаг метода РК4 для двумерной системы  
void rK\_step(std::pair<double,double> (\*f)(std::pair<double,double>, double),double& x,std::pair<double,double>& u,double step,double a);  
  
// Проверка выхода на правую границу  
bool inside(double x,double b,double eps\_b);  
  
// Метод для тестовой задачи  
// -------------------------  
// u0 - начальное значение  
// Nmax - макс. число шагов  
// b - правая граница  
// eps\_b - контроль выхода на правую границу  
// eps - контроль лок. погрешности  
// step - начальный шаг  
// ------------------------  
// В файл-1 уходит:  
// - точное + приближённое решение + то, что нужно для таблицы (удвоения, ОЛП и прочее)  
// В файл-2 уходит:  
// - выходные данные:  
// - - n - число проделанных итераций  
// - - b-x\_n - отход от границы  
// - - max|ОЛП|  
// - - всего удвоений шага  
// - - всего делений шага  
// - - max h = "" при x = ""  
// - - min h= "" при x = ""  
// - - max|u\_i-v\_i| = "" при x = ""  
void run\_test\_method(double u0, int Nmax,double b, double eps\_b, double eps, double step);  
  
// Метод для основной задачи - 1  
// -------------------------  
// u0 - начальное значение  
// Nmax - макс. число шагов  
// b - правая граница  
// eps\_b - контроль выхода на правую границу  
// eps - контроль лок. погрешности  
// step - начальный шаг  
// ------------------------  
// В файл-1 уходит:  
// - приближённое решение + то, что нужно для таблицы (удвоения, ОЛП и прочее)  
// В файл-2 уходит:  
// - выходные данные:  
// - - n - число проделанных итераций  
// - - b-x\_n - отход от границы  
// - - max|ОЛП|  
// - - всего удвоений шага  
// - - всего делений шага  
// - - max h = "" при x = ""  
// - - min h= "" при x = ""  
  
void run\_main\_method\_1(double u0, int Nmax,double b, double eps\_b, double eps, double step);  
  
// Евклидова норма  
double euclid\_norm(std::pair<double,double> v1, std::pair<double,double> v2);  
  
// Метод для основной задачи - 2  
// -------------------------  
// (u0,u0\_dot) - начальные значения  
// Nmax - макс. число шагов  
// b - правая граница  
// eps\_b - контроль выхода на правую границу  
// eps - контроль лок. погрешности  
// step - начальный шаг  
// a - параметр в уравнении  
// ------------------------  
// В файл-1 уходит:  
// - приближённое решение + то, что нужно для таблицы (удвоения, ОЛП и прочее)  
// В файл-2 уходит:  
// - выходные данные:  
// - - n - число проделанных итераций  
// - - b-x\_n - отход от границы  
// - - max|ОЛП|  
// - - всего удвоений шага  
// - - всего делений шага  
// - - max h = "" при x = ""  
// - - min h= "" при x = ""  
  
void run\_main\_method\_2(double u0,double u0\_dot, int Nmax,double b, double eps\_b, double eps, double step,double a);