

1 PRZEZNACZENIE BIBLIOTEKI

Biblioteka «libmatrix» ma cel przedstawić użytkownikowi funkcji do posługiwania się na macierzach

2 OPIS FUNKCJI

Prototyp funkcji	Opis funkcji
matrix make_matrix (const int <i>n_rows</i> , const int <i>n_columns</i>)	Funkcja tworzy nową macierz o <i>n_rows</i> wierszach i <i>n_columns</i> kolumnach.
double determinant (const matrix*)	Zwraca wartość wyznacznika macierzy, podanej jako parametr.
int copy_matrix (IN const matrix* , OUT matrix*)	Kopiuje macierz IN do macierzy OUT, zwracając 1 w przypadku sukcesu i 0 w przeciwnym przypadku.
int transponent (IN const matrix* , OUT matrix*)	Zapisuje do macierzy OUT transponowaną macierz IN. W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int multiplex_matrixes (IN const matrix* <i>in_matrix_1</i> , IN const matrix* <i>in_matrix_2</i> , OUT matrix*)	Zapisuje do macierzy OUT iloczyn macierzy <i>in_matrix_1</i> przez macierz <i>in_matrix_2</i> . W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int triangle_form (IN const matrix* , OUT matrix*)	Zapisuje do macierzy OUT macierz IN w postaci schodkowej. W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int show_matrix (IN const matrix*)	Wypisuje macierz IN w konsoli. W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int init_matrix (OUT matrix*)	Inicjalizuje macierz, podaną w parametrze wartościami, odczytanymi ze standardowego wejścia. W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int init_matrix_by_random (OUT matrix* , const int32_t <i>down</i> , const int32_t <i>up</i>)	Inicjalizuje macierz OUT wartościami losowymi typu int w zakresie od <i>down</i> do <i>up</i> . Parametry <i>down</i> i <i>up</i> muszą być typy int32_t lub mieć rozmiar nie więcej, niż 32 bity. W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int init_matrix_as_unit (OUT matrix*)	Inicjalizuje macierz OUT jako jedynkową, to znaczy 1 na głównej przekątnej i 0 na pozostałych miejscach. W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int init_matrix_by_function (OUT matrix* , init_user)	Inicjalizuje macierz OUT, używając funkcję init_user . W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int delete_matrix (matrix*)	Wyzwala pamięć dynamiczną, zajęta macierzą, podaną jako parametr. W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.

int rows_swap (const int <i>i_row_1</i> , const int <i>i_row_2</i> , MODYFIED matrix *)	Zamienia miejscami wierszy <i>i_row_1</i> i <i>i_row_2</i> macierzy MODYFIED. W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int columns_swap (const int <i>i_col_1</i> , const int <i>i_col_2</i> , MODYFIED matrix *)	Zamienia miejscami kolumny <i>i_col_1</i> i <i>i_col_2</i> macierzy MODYFIED. W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int row_mult_on_const (const double <i>factor</i> , const int <i>i_row</i> , MODYFIED matrix *)	Mnoży wiersz <i>i_row</i> macierzy MODYFIED przez wartość <i>factor</i> . W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int column_mult_on_const (const double <i>factor</i> , const int <i>i_column</i> , MODYFIED matrix *)	Mnoży kolumnę <i>i_column</i> macierzy MODYFIED przez wartość <i>factor</i> . W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int rows_sub (const double <i>factor</i> , const int <i>i_subtracted_row</i> , const int <i>i_subtracting_row</i> , MODYFIED matrix *)	Odejmuje od wiersza <i>i_subtracted_row</i> macierzy MODYFIED wiersz <i>i_subtracting_row</i> , mnożony przez <i>factor</i> . W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int columns_sub (const double <i>factor</i> , const int <i>i_subtracted_column</i> , const int <i>i_subtracting_column</i> , MODYFIED matrix *)	Odejmuje od kolumny <i>i_subtracted_column</i> macierzy MODYFIED kolumnę <i>i_subtracting_column</i> , mnożoną przez <i>factor</i> . W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int copy_row_to_other_matrix (const int <i>i_source</i> , const int <i>i_reciever</i> , IN const matrix * <i>in_matrix</i> , OUT matrix * <i>out_matrix</i>)	Kopiuje wiersz <i>i_source</i> macierzy <i>in_matrix</i> do wierszu <i>i_reciever</i> macierzy <i>out_matrix</i> . W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int copy_column_to_other_matrix (const int <i>i_source</i> , const int <i>i_reciever</i> , IN const matrix * <i>in_matrix</i> , OUT matrix * <i>out_matrix</i>)	Kopiuje kolumnę <i>i_source</i> macierzy <i>in_matrix</i> do kolumny <i>i_reciever</i> macierzy <i>out_matrix</i> . W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.
int inverse_matrix (IN const matrix * <i>in_matrix</i> , OUT matrix * <i>out_matrix</i>)	Zapisuje do macierzy <i>out_matrix</i> macierz odwrotną do macierzy <i>in_matrix</i> . W przypadku sukcesu zwraca 1, w przeciwnym przypadku 0.