// SPDX-License-Identifier: MIT pragma solidity ^0.6.2; interface IUniswapV2Router01 { function factory() external pure returns (address); function WETH() external pure returns (address); function addLiquidity( address tokenA, address tokenB, uint amountADesired, uint amountBDesired, uint amountAMin, uint amountBMin, address to, uint deadline ) external returns (uint amountA, uint amountB, uint liquidity); function addLiquidityETH( address token, uint amountTokenDesired, uint amountTokenMin, uint amountETHMin, address to, uint deadline ) external payable returns (uint amountToken, uint amountETH, uint liquidity); function removeLiquidity( address tokenA, address tokenB, uint liquidity, uint amountAMin, uint amountBMin, address to, uint deadline ) external returns (uint amountA, uint amountB); function removeLiquidityETH( address token, uint liquidity, uint amountTokenMin, uint amountETHMin, address to, uint deadline ) external returns (uint amountToken, uint amountETH); function removeLiquidityWithPermit( address tokenA, address tokenB, uint liquidity, uint amountAMin, uint amountBMin, address to, uint deadline, bool approveMax, uint8 v, bytes32 r, bytes32 s ) external returns (uint amountA, uint amountB); function removeLiquidityETHWithPermit( address token, uint liquidity, uint amountTokenMin, uint amountETHMin, address to, uint deadline, bool approveMax, uint8 v, bytes32 r, bytes32 s ) external returns (uint amountToken, uint amountETH); function swapExactTokensForTokens( uint amountIn, uint amountOutMin, address[] calldata path, address to, uint deadline ) external returns (uint[] memory amounts); function swapTokensForExactTokens( uint amountOut, uint amountInMax, address[] calldata path, address to, uint deadline ) external returns (uint[] memory amounts); function swapExactETHForTokens(uint amountOutMin, address[] calldata path, address to, uint deadline) external payable returns (uint[] memory amounts); function swapTokensForExactETH(uint amountOut, uint amountInMax, address[] calldata path, address to, uint deadline) external returns (uint[] memory amounts); function swapExactTokensForETH(uint amountIn, uint amountOutMin, address[] calldata path, address to, uint deadline) external returns (uint[] memory amounts); function swapETHForExactTokens(uint amountOut, address[] calldata path, address to, uint deadline) external payable returns (uint[] memory amounts); function quote(uint amountA, uint reserveA, uint reserveB) external pure returns (uint amountB); function getAmountOut(uint amountIn, uint reserveIn, uint reserveOut) external pure returns (uint amountOut); function getAmountIn(uint amountOut, uint reserveIn, uint reserveOut) external pure returns (uint amountIn); function getAmountsOut(uint amountIn, address[] calldata path) external view returns (uint[] memory amounts); function getAmountsIn(uint amountOut, address[] calldata path) external view returns (uint[] memory amounts); } // pragma solidity >=0.6.2; interface IUniswapV2Router02 is IUniswapV2Router01 { function removeLiquidityETHSupportingFeeOnTransferTokens( address token, uint liquidity, uint amountTokenMin, uint amountETHMin, address to, uint deadline ) external returns (uint amountETH); function removeLiquidityETHWithPermitSupportingFeeOnTransferTokens( address token, uint liquidity, uint amountTokenMin, uint amountETHMin, address to, uint deadline, bool approveMax, uint8 v, bytes32 r, bytes32 s ) external returns (uint amountETH); function swapExactTokensForTokensSupportingFeeOnTransferTokens( uint amountIn, uint amountOutMin, address[] calldata path, address to, uint deadline ) external; function swapExactETHForTokensSupportingFeeOnTransferTokens( uint amountOutMin, address[] calldata path, address to, uint deadline ) external payable; function swapExactTokensForETHSupportingFeeOnTransferTokens( uint amountIn, uint amountOutMin, address[] calldata path, address to, uint deadline ) external; }