

Stdlib	Containers	Batteries	Base
	CCArrayLabels.( - ) : int -> int -> int array		
	CCArrayLabels.( ^ ) : int -> int -> int array		
	CCArrayLabels.( >= ) : 'a array -> ('a -> 'b array) -> 'b array		
	CCArrayLabels.( >  ) : 'a array -> ('a -> 'b) -> 'b array		
	CCArrayLabels.( >  = ) : 'a array -> ('a -> 'b) -> 'b array		
	CCArrayLabels.( and* ) : 'a array -> 'b array -> ('a * 'b) array		
	CCArrayLabels.( and+ ) : 'a array -> 'b array -> ('a * 'b) array		
	CCArrayLabels.( let* ) : 'a array -> ('a -> 'b array) -> 'b array		
	CCArrayLabels.( let+ ) : 'a array -> ('a -> 'b) -> 'b array		
ArrayLabels.append : 'a array -> 'a array -> 'a array	CCArrayLabels.append : 'a array -> 'a array -> 'a array		Base.Array.append : 'a array -> 'a array -> 'a array
			Base.Array.binary_search : ('a array, 'a, 'key) Base.Binary_searchable_intf.binary_search
			Base.Array.binary_search_segmented : ('a array, 'a) Base.Binary_searchable_intf.binary_search_segmented
ArrayLabels.blit : src:'a array -> src_pos:int -> dst:'a array -> dst_pos:int -> len:int -> unit	CCArrayLabels.blit : src:'a array -> src_pos:int -> dst:'a array -> dst_pos:int -> len:int -> unit	BatArray.Labels.blit : src:'a array -> src_pos:int -> dst:'a array -> dst_pos:int -> len:int -> unit	Base.Array0.blit : src:'a array ? src_pos:int -> dst:'a array -> dst_pos:int -> len:int -> unit
			Base.Array.blito : ('a array, 'a array) Base.Blit_intf.blito
	CCArrayLabels.bsearch : cmp:('a -> 'a -> int) -> key:'a -> 'a array -> [ `All_bigger   `All_lower   `At of int   `Empty   `Just_after of int ]		
		BatArray.cartesian_product : 'a array -> 'b array -> ('a * 'b) array	Base.Array.cartesian_product : 'a array -> 'b array -> ('a * 'b) array
	CCArrayLabels.compare : 'a CCArrayLabels.ord -> 'a array CCArrayLabels.ord		Base.Array.compare : ('a -> 'a -> int) -> 'a array -> 'a array -> int
ArrayLabels.concat : 'a array list -> 'a array	CCArrayLabels.concat : 'a array list -> 'a array		Base.Array.concat : 'a array list -> 'a array
			Base.Array.concat_map : 'a array -> f:('a -> 'b array) -> 'b array
			Base.Array.concat_mapi : 'a array -> f:(int -> 'a -> 'b array) -> 'b array
ArrayLabels.copy : 'a array -> 'a array	CCArrayLabels.copy : 'a array -> 'a array		Base.Array.copy : 'a array -> 'a array
		BatArray.Labels.count_matching : f:('a -> bool) -> 'a array -> int	Base.Array.count : 'a array -> f:('a -> bool) -> int
			Base.Array.counti : 'a array -> f:(int -> 'a -> bool) -> int
ArrayLabels.create : int -> 'a -> 'a array	CCArrayLabels.create : int -> 'a -> 'a array	BatArray.Labels.create : int -> init:'a -> 'a array	Base.Array.create : len:int -> 'a -> 'a array
ArrayLabels.create_float : int -> float array	CCArrayLabels.create_float : int -> float array		
ArrayLabels.create_matrix : dimx:int -> dimy:int -> 'a -> 'a array array	CCArrayLabels.create_matrix : dimx:int -> dimy:int -> 'a -> 'a array array	BatArray.Labels.create_matrix : dimx:int -> dimy:int -> 'a -> 'a array array	
	CCArrayLabels.empty : 'a array = []		
	CCArrayLabels.equal : 'a CCArrayLabels.equal -> 'a array CCArrayLabels.equal		Base.Array.equal : ('a -> 'a -> bool) -> 'a array -> 'a array -> bool
	CCArrayLabels.except_idx : 'a array -> int -> 'a list		
ArrayLabels.exists : f:('a -> bool) -> 'a array -> bool	CCArrayLabels.exists : f:('a -> bool) -> 'a array -> bool	BatArray.Labels.exists : f:('a -> bool) -> 'a array -> bool	Base.Array.exists : 'a array -> f:('a -> bool) -> bool
ArrayLabels.exists2 : f:('a -> 'b -> bool) -> 'a array -> 'b array -> bool	CCArrayLabels.exists2 : f:('a -> 'b -> bool) -> 'a array -> 'b array -> bool		Base.Array.exists2_exn : 'a array -> 'b array -> f:('a -> 'b -> bool) -> bool
			Base.Array.existsi : 'a array -> f:(int -> 'a -> bool) -> bool
ArrayLabels.fast_sort : cmp:('a -> 'a -> int) -> 'a array -> unit	CCArrayLabels.fast_sort : cmp:('a -> 'a -> int) -> 'a array -> unit	BatArray.Labels.fast_sort : cmp:('a -> 'a -> int) -> 'a array -> unit	
ArrayLabels.fill : 'a array -> pos:int -> len:int -> 'a -> unit	CCArrayLabels.fill : 'a array -> pos:int -> len:int -> 'a -> unit	BatArray.Labels.fill : 'a array -> pos:int -> len:int -> 'a -> unit	Base.Array.fill : 'a array -> pos:int -> len:int -> 'a -> unit

Stdlib	Containers	Batteries	Base
	CCArrayLabels.filter : f:(a -> bool) -> 'a array -> 'a array	BatArray.Labels.filter : f:(a -> bool) -> 'a array -> 'a array	Base.Array.filter : 'a array -> f:(a -> bool) -> 'a array
	CCArrayLabels.filter_map : f:(a -> 'b option) -> 'a array -> 'b array	BatArray.Labels.filter_map : f:(a -> 'b option) -> 'a array -> 'b array	Base.Array.filter_map : 'a array -> f:(a -> 'b option) -> 'b array
			Base.Array.filter_mapi : 'a array -> f:(int -> 'a -> 'b option) -> 'b array
			Base.Array.filter_opt : 'a option array -> 'a array
			Base.Array.filteri : 'a array -> f:(int -> 'a -> bool) -> 'a array
		BatArray.Labels.find : f:(a -> bool) -> 'a array -> 'a	Base.Array.find_exn : 'a array -> f:(a -> bool) -> 'a
			Base.Array.find : 'a array -> f:(a -> bool) -> 'a option
			Base.Array.find_consecutive_duplicate : 'a array -> equal:(a -> 'a -> bool) -> ('a * 'a) option
			Base.Array.find_map : 'a array -> f:(a -> 'b option) -> 'b option
			Base.Array.find_map_exn : 'a array -> f:(a -> 'b option) -> 'b
			Base.Array.find_mapi : 'a array -> f:(int -> 'a -> 'b option) -> 'b option
			Base.Array.find_mapi_exn : 'a array -> f:(int -> 'a -> 'b option) -> 'b
		BatArray.Labels.findi : f:(a -> bool) -> 'a array -> int	
			Base.Array.findi : 'a array -> f:(int -> 'a -> bool) -> (int * 'a) option
			Base.Array.findi_exn : 'a array -> f:(int -> 'a -> bool) -> int * 'a
	CCArrayLabels.find_idx : f:(a -> bool) -> 'a array -> (int * 'a) option		
	CCArrayLabels.find_map : f:(a -> 'b option) -> 'a array -> 'b option		
	CCArrayLabels.find_map_i : f:(int -> 'a -> 'b option) -> 'a array -> 'b option		
	CCArrayLabels.flat_map : f:(a -> 'b array) -> 'a array -> 'b array		
	CCArrayLabels.fold : f:(a -> 'b -> 'a) -> init:'a -> 'b array -> 'a	BatArray.Labels.fold : f:(a -> 'b -> 'a) -> init:'a -> 'b array -> 'a	Base.Array.fold : 'a array -> init:'accum -> f:(accum -> 'a -> 'accum) -> 'accum
	CCArrayLabels.fold2 : f:(acc -> 'a -> 'b -> 'acc) -> init:'acc -> 'a array -> 'b array -> 'acc		Base.Array.fold2_exn : 'a array -> 'b array -> init:'c -> f:(c -> 'a -> 'b -> 'c) -> 'c
ArrayLabels.fold_left : f:(a -> 'b -> 'a) -> init:'a -> 'b array -> 'a	CCArrayLabels.fold_left : f:(a -> 'b -> 'a) -> init:'a -> 'b array -> 'a	BatArray.Labels.fold_left : f:(a -> 'b -> 'a) -> init:'a -> 'b array -> 'a	
	CCArrayLabels.fold_map : f:(acc -> 'a -> 'acc * 'b) -> init:'acc -> 'a array -> 'acc * 'b array		Base.Array.fold_map : 'a array -> init:'b -> f:(b -> 'a -> 'b * 'c) -> 'b * 'c array
			Base.Array.fold_mapi : 'a array -> init:'b -> f:(int -> 'b -> 'a -> 'b * 'c) -> 'b * 'c array
			Base.Array.fold_result : 'a array -> init:'accum -> f:(accum -> 'a -> ('accum, 'e) Base.Result.t) -> ('accum, 'e) Base.Result.t
ArrayLabels.fold_right : f:(b -> 'a -> 'a) -> 'b array -> init:'a -> 'a	CCArrayLabels.fold_right : f:(b -> 'a -> 'a) -> 'b array -> init:'a -> 'a	BatArray.Labels.fold_right : f:(b -> 'a -> 'a) -> 'b array -> init:'a -> 'a	Base.Array.fold_right : 'a array -> f:(a -> 'b -> 'b) -> init:'b -> 'b
			Base.Array.fold_until : 'a array -> init:'accum -> f:(accum -> 'a -> ('accum, 'final) Base.Container_intf.Continue_or_stop.t) -> finish:(accum -> 'final) -> 'final
		BatArray.Labels.fold_while : p:(acc -> 'a -> bool) -> f:(acc -> 'a -> 'acc) -> init:'acc -> 'a array -> 'acc * int	
	CCArrayLabels.fold_while : f:(a -> 'b -> 'a * [ `Continue   `Stop ]) -> init:'a -> 'b array -> 'a		
	CCArrayLabels.foldi : f:(a -> int -> 'b -> 'a) -> init:'a -> 'b array -> 'a		Base.Array.foldi : 'a array -> init:'b -> f:(int -> 'b -> 'a -> 'b) -> 'b
			Base.Array.folding_map : 'a array -> init:'b -> f:(b -> 'a -> 'b * 'c) -> 'c array
			Base.Array.folding_mapi : 'a array -> init:'b -> f:(int -> 'b -> 'a -> 'b * 'c) -> 'c array
ArrayLabels.for_all : f:(a -> bool) -> 'a array -> bool	CCArrayLabels.for_all : f:(a -> bool) -> 'a array -> bool	BatArray.Labels.for_all : f:(a -> bool) -> 'a array -> bool	Base.Array.for_all : 'a array -> f:(a -> bool) -> bool
ArrayLabels.for_all2 : f:(a -> 'b -> bool) -> 'a array -> 'b array -> bool	CCArrayLabels.for_all2 : f:(a -> 'b -> bool) -> 'a array -> 'b array -> bool		Base.Array.for_all2_exn : 'a array -> 'b array -> f:(a -> 'b -> bool) -> bool
			Base.Array.for_alli : 'a array -> f:(int -> 'a -> bool) -> bool

Stdlib	Containers	Batteries	Base
ArrayLabels.get : 'a array -> int -> 'a	CCArrayLabels.get : 'a array -> int -> 'a		Base.Array.get : 'a array -> int -> 'a
	CCArrayLabels.get_safe : 'a array -> int -> 'a option		
ArrayLabels.init : int -> f:(int -> 'a) -> 'a array	CCArrayLabels.init : int -> f:(int -> 'a) -> 'a array	BatArray.Labels.init : int -> f:(int -> 'a) -> 'a array	Base.Array.init : int -> f:(int -> 'a) -> 'a array
			Base.Array.invariant : 'a Base.Invariant_intf.inv -> 'a array Base.Invariant_intf.inv
			Base.Array.is_empty : 'a array -> bool
			Base.Array.is_sorted : 'a array -> compare:(a -> 'a -> int) -> bool
			Base.Array.is_sorted_strictly : 'a array -> compare:(a -> 'a -> int) -> bool
ArrayLabels.iter : f:(a -> unit) -> 'a array -> unit	CCArrayLabels.iter : f:(a -> unit) -> 'a array -> unit	BatArray.Labels.iter : f:(a -> unit) -> 'a array -> unit	Base.Array.iter : 'a array -> f:(a -> unit) -> unit
ArrayLabels.iter2 : f:(a -> 'b -> unit) -> 'a array -> 'b array -> unit	CCArrayLabels.iter2 : f:(a -> 'b -> unit) -> 'a array -> 'b array -> unit	BatArray.Labels.iter2 : f:(a -> 'b -> unit) -> 'a array -> 'b array -> unit	Base.Array.iter2_exn : 'a array -> 'b array -> f:(a -> 'b -> unit) -> unit
		BatArray.Labels.iter2i : f:(int -> 'a -> 'b -> unit) -> 'a array -> 'b array -> unit	
ArrayLabels.iteri : f:(int -> 'a -> unit) -> 'a array -> unit	CCArrayLabels.iteri : f:(int -> 'a -> unit) -> 'a array -> unit	BatArray.Labels.iteri : f:(int -> 'a -> unit) -> 'a array -> unit	Base.Array.iteri : 'a array -> f:(int -> 'a -> unit) -> unit
			Base.Array.last : 'a array -> 'a
ArrayLabels.length : 'a array -> int	CCArrayLabels.length : 'a array -> int		Base.Array.length : 'a array -> int
	CCArrayLabels.lookup : cmp:'a CCArrayLabels.ord -> key:'a -> 'a array -> int option		
	CCArrayLabels.lookup_exn : cmp:'a CCArrayLabels.ord -> key:'a -> 'a array -> int		
ArrayLabels.make : int -> 'a -> 'a array	CCArrayLabels.make : int -> 'a -> 'a array		
ArrayLabels.make_float : int -> float array	CCArrayLabels.make_float : int -> float array		
ArrayLabels.make_matrix : dimx:int -> dimy:int -> 'a -> 'a array array	CCArrayLabels.make_matrix : dimx:int -> dimy:int -> 'a -> 'a array array	BatArray.Labels.make_matrix : dimx:int -> dimy:int -> 'a -> 'a array array	Base.Array.make_matrix : dimx:int -> dimy:int -> 'a -> 'a array array
ArrayLabels.map : f:(a -> 'b) -> 'a array -> 'b array	CCArrayLabels.map : f:(a -> 'b) -> 'a array -> 'b array	BatArray.Labels.map : f:(a -> 'b) -> 'a array -> 'b array	Base.Array.map : 'a array -> f:(a -> 'b) -> 'b array
ArrayLabels.map2 : f:(a -> 'b -> 'c) -> 'a array -> 'b array -> 'c array	CCArrayLabels.map2 : f:(a -> 'b -> 'c) -> 'a array -> 'b array -> 'c array		Base.Array.map2_exn : 'a array -> 'b array -> f:(a -> 'b -> 'c) -> 'c array
			Base.Array.map_inplace : 'a array -> f:(a -> 'a) -> unit
ArrayLabels.mapi : f:(int -> 'a -> 'b) -> 'a array -> 'b array	CCArrayLabels.mapi : f:(int -> 'a -> 'b) -> 'a array -> 'b array	BatArray.Labels.mapi : f:(int -> 'a -> 'b) -> 'a array -> 'b array	Base.Array.mapi : 'a array -> f:(int -> 'a -> 'b) -> 'b array
			Base.Array.max_elt : 'a array -> compare:(a -> 'a -> int) -> 'a option
			Base.Array.max_length : int = 18014398509481983
ArrayLabels.mem : 'a -> set:'a array -> bool	CCArrayLabels.mem : ?eq:(a -> 'a -> bool) -> 'a -> 'a array -> bool		Base.Array.mem : 'a array -> 'a -> equal:(a -> 'a -> bool) -> bool
ArrayLabels.memq : 'a -> set:'a array -> bool	CCArrayLabels.memq : 'a -> set:'a array -> bool		
			Base.Array.min_elt : 'a array -> compare:(a -> 'a -> int) -> 'a option
		BatArray.Labels.modify : f:(a -> 'a) -> 'a array -> unit	
		BatArray.Labels.modifyi : f:(int -> 'a -> 'a) -> 'a array -> unit	
	CCArrayLabels.monoid_product : f:(a -> 'b -> 'c) -> 'a array -> 'b array -> 'c array		
ArrayLabels.of_list : 'a list -> 'a array	CCArrayLabels.of_list : 'a list -> 'a array		Base.Array.of_list : 'a list -> 'a array
			Base.Array.of_list_map : 'a list -> f:(a -> 'b) -> 'b array
			Base.Array.of_list_mapi : 'a list -> f:(int -> 'a -> 'b) -> 'b array
			Base.Array.of_list_rev : 'a list -> 'a array
			Base.Array.of_list_rev_map : 'a list -> f:(a -> 'b) -> 'b array
			Base.Array.of_list_rev_mapi : 'a list -> f:(int -> 'a -> 'b) -> 'b array
ArrayLabels.of_seq : 'a Seq.t -> 'a array	CCArrayLabels.of_seq : 'a Seq.t -> 'a array		

Stdlib	Containers	Batteries	Base
		BatArray.partition : ('a -> bool) -> 'a array -> 'a array * 'a array	Base.Array.partition_tf : 'a array -> f:('a -> bool) -> 'a array * 'a array
			Base.Array.partitioni_tf : 'a array -> f:(int -> 'a -> bool) -> 'a array * 'a array
	CCArrayLabels.pp : ?pp_start:unit CCArrayLabels.printer -> ?pp_stop:unit CCArrayLabels.printer -> ?pp_sep:unit CCArrayLabels.printer -> 'a CCArrayLabels.printer -> 'a array CCArrayLabels.printer		
	CCArrayLabels.pp_i : ?pp_start:unit CCArrayLabels.printer -> ?pp_stop:unit CCArrayLabels.printer -> ?pp_sep:unit CCArrayLabels.printer -> (int -> 'a CCArrayLabels.printer) -> 'a array CCArrayLabels.printer		
	CCArrayLabels.random : 'a CCArrayLabels.random_gen -> 'a array CCArrayLabels.random_gen		
	CCArrayLabels.random_choose : 'a array -> 'a CCArrayLabels.random_gen		Base.Array.random_element_exn : ?random_state:Base.Random.State.t -> 'a array -> 'a
			Base.Array.random_element : ?random_state:Base.Random.State.t -> 'a array -> 'a option
	CCArrayLabels.random_len : int -> 'a CCArrayLabels.random_gen -> 'a array CCArrayLabels.random_gen		
	CCArrayLabels.random_non_empty : 'a CCArrayLabels.random_gen -> 'a array CCArrayLabels.random_gen		
			Base.Array.reduce : 'a array -> f:('a -> 'a -> 'a) -> 'a option
		BatArray.reduce : ('a -> 'a -> 'a) -> 'a array -> 'a	Base.Array.reduce_exn : 'a array -> f:('a -> 'a -> 'a) -> 'a
	CCArrayLabels.rev : 'a array -> 'a array		
	CCArrayLabels.reverse_in_place : 'a array -> unit		Base.Array.rev_inplace : 'a array -> unit
	CCArrayLabels.scan_left : f:('acc -> 'a -> 'acc) -> init:'acc -> 'a array -> 'acc array		
ArrayLabels.set : 'a array -> int -> 'a -> unit	CCArrayLabels.set : 'a array -> int -> 'a -> unit		Base.Array.set : 'a array -> int -> 'a -> unit
			Base.Array.sexp_of_t : ('a -> Sexplib0.Sexp.t) -> 'a array -> Sexplib0.Sexp.t
	CCArrayLabels.shuffle : 'a array -> unit		
	CCArrayLabels.shuffle_with : Random.State.t -> 'a array -> unit		Base.Array.permute : ?random_state:Base.Random.State.t -> 'a array -> unit
ArrayLabels.sort : cmp:('a -> 'a -> int) -> 'a array -> unit	CCArrayLabels.sort : cmp:('a -> 'a -> int) -> 'a array -> unit	BatArray.Labels.sort : cmp:('a -> 'a -> int) -> 'a array -> unit	Base.Array.sort : ?pos:int -> ?len:int -> 'a array -> compare:('a -> 'a -> int) -> unit
	CCArrayLabels.sort_generic : (module CCArrayLabels.MONO_ARRAY with type elt = 'elt and type t = 'arr) -> cmp:('elt -> 'elt -> int) -> 'arr -> unit		
	CCArrayLabels.sort_indices : f:('a -> 'a -> int) -> 'a array -> int array		
	CCArrayLabels.sort_ranking : f:('a -> 'a -> int) -> 'a array -> int array		
	CCArrayLabels.sorted : f:('a -> 'a -> int) -> 'a array -> 'a array		Base.Array.sorted_copy : 'a array -> compare:('a -> 'a -> int) -> 'a array
ArrayLabels.stable_sort : cmp:('a -> 'a -> int) -> 'a array -> unit	CCArrayLabels.stable_sort : cmp:('a -> 'a -> int) -> 'a array -> unit	BatArray.Labels.stable_sort : cmp:('a -> 'a -> int) -> 'a array -> unit	Base.Array.stable_sort : 'a array -> compare:('a -> 'a -> int) -> unit
ArrayLabels.sub : 'a array -> pos:int -> len:int -> 'a array	CCArrayLabels.sub : 'a array -> pos:int -> len:int -> 'a array	BatArray.Labels.sub : 'a array -> pos:int -> len:int -> 'a array	Base.Array0.sub : 'a array -> pos:int -> len:int -> 'a array
			Base.Array.subo : ('a array, 'a array) Base.Blit_intf.subo
			Base.Array.sum : (module Base.Container_intf.Summable with type t = 'sum) -> 'a array -> f:('a -> 'sum) -> 'sum
	CCArrayLabels.swap : 'a array -> int -> int -> unit		Base.Array.swap : 'a array -> int -> int -> unit
			Base.Array.t_of_sexp : (Sexplib0.Sexp.t -> 'a) -> Sexplib0.Sexp.t -> 'a array
			Base.Array.t_sexp_grammar : Base.Ppx_sexp_conv_lib.Sexp.Private.Raw_grammar.t...
			Base.Array.to_array : 'a array -> 'a array
	CCArrayLabels.to_gen : 'a array -> 'a CCArrayLabels.gen		

Stdlib	Containers	Batteries	Base
	CCArrayLabels.to_iter : 'a array -> 'a CCArryLabels.iter		
ArrayLabels.to_list : 'a array -> 'a list	CCArrayLabels.to_list : 'a array -> 'a list		Base.Array.to_list : 'a array -> 'a list
ArrayLabels.to_seq : 'a array -> 'a Seq.t	CCArrayLabels.to_seq : 'a array -> 'a Seq.t		
ArrayLabels.to_seqi : 'a array -> (int * 'a) Seq.t	CCArrayLabels.to_seqi : 'a array -> (int * 'a) Seq.t		
			Base.Array.to_sequence : 'a array -> 'a Base.Sequence.t
			Base.Array.to_sequence_mutable : 'a array -> 'a Base.Sequence.t
	CCArrayLabels.to_string : ?sep:string -> ('a -> string) -> 'a array -> string		
			Base.Array.transpose : 'a array array -> 'a array array option
			Base.Array.transpose_exn : 'a array array -> 'a array array
			Base.Array.unsafe_blit : ('a array, 'a array) Base.Blit.blit
ArrayLabels.unsafe_get : 'a array -> int -> 'a	CCArrayLabels.unsafe_get : 'a array -> int -> 'a		Base.Array.unsafe_get : 'a array -> int -> 'a
ArrayLabels.unsafe_set : 'a array -> int -> 'a -> unit	CCArrayLabels.unsafe_set : 'a array -> int -> 'a -> unit		Base.Array.unsafe_set : 'a array -> int -> 'a -> unit
		BatArray.split : ('a * 'b) array -> 'a array * 'b array	Base.Array.unzip : ('a * 'b) array -> 'a array * 'b array
			Base.Array.zip : 'a array -> 'b array -> ('a * 'b) array option
			Base.Array.zip_exn : 'a array -> 'b array -> ('a * 'b) array
	CCArray.( - ) : int -> int -> int array		
	CCArray.( - ^ ) : int -> int -> int array		
	CCArray.( >= ) : 'a array -> ('a -> 'b array) -> 'b array		
	CCArray.( >  ) : 'a array -> ('a -> 'b) -> 'b array		
	CCArray.( >  = ) : 'a array -> ('a -> 'b) -> 'b array		
	CCArray.( and* ) : 'a array -> 'b array -> ('a * 'b) array		
	CCArray.( and+ ) : 'a array -> 'b array -> ('a * 'b) array		
	CCArray.( let* ) : 'a array -> ('a -> 'b array) -> 'b array		
	CCArray.( let+ ) : 'a array -> ('a -> 'b) -> 'b array		
Array.append : 'a array -> 'a array -> 'a array	CCArray.append : 'a array -> 'a array -> 'a array	BatArray.append : 'a array -> 'a array -> 'a array	
		BatArray.avg : int array -> float	
		BatArray.backwards : 'a array -> 'a BatEnum.t	
Array.blit : 'a array -> int -> 'a array -> int -> int -> unit	CCArray.blit : 'a array -> int -> 'a array -> int -> int -> unit	BatArray.blit : 'a array -> int -> 'a array -> int -> int -> unit	
	CCArray.bsearch : cmp:('a -> 'a -> int) -> 'a -> 'a array -> [ 'All_bigger   'All_lower   'At of int   'Empty   'Just_after of int ]	BatArray.bsearch : 'a BatOrd.ord -> 'a array -> 'a -> [ 'All_bigger   'All_lower   'At of int   'Empty   'Just_after of int ]	
		BatArray.cartesian_product : 'a array -> 'b array -> ('a * 'b) array	
	CCArray.compare : 'a CCArry.ord -> 'a array CCArry.ord	BatArray.compare : 'a BatOrd.comp -> 'a array BatOrd.comp	
Array.concat : 'a array list -> 'a array	CCArray.concat : 'a array list -> 'a array	BatArray.concat : 'a array list -> 'a array	
Array.copy : 'a array -> 'a array	CCArray.copy : 'a array -> 'a array	BatArray.copy : 'a array -> 'a array	
		BatArray.count_matching : ('a -> bool) -> 'a array -> int	
Array.create : int -> 'a -> 'a array	CCArray.create : int -> 'a -> 'a array	BatArray.create : int -> 'a -> 'a array	
Array.create_float : int -> float array	CCArray.create_float : int -> float array	BatArray.create_float : int -> float array	
Array.create_matrix : int -> int -> 'a -> 'a array array	CCArray.create_matrix : int -> int -> 'a -> 'a array array	BatArray.create_matrix : int -> int -> 'a -> 'a array array	

Stdlib	Containers	Batteries	Base
		BatArray.decorate_fast_sort : ('a -> 'b) -> 'a array -> 'a array	
		BatArray.decorate_stable_sort : ('a -> 'b) -> 'a array -> 'a array	
	CCArray.empty : 'a array = []		
		BatArray.enum : 'a array -> 'a BatEnum.t	
	CCArray.equal : 'a CCArray.equal -> 'a array CCArray.equal	BatArray.equal : 'a BatOrd.eq -> 'a array BatOrd.eq	
	CCArray.except_idx : 'a array -> int -> 'a list		
Array.exists : ('a -> bool) -> 'a array -> bool	CCArray.exists : ('a -> bool) -> 'a array -> bool	BatArray.exists : ('a -> bool) -> 'a array -> bool	
Array.exists2 : ('a -> 'b -> bool) -> 'a array -> 'b array -> bool	CCArray.exists2 : ('a -> 'b -> bool) -> 'a array -> 'b array -> bool	BatArray.exists2 : ('a -> 'b -> bool) -> 'a array -> 'b array -> bool	
Array.fast_sort : ('a -> 'a -> int) -> 'a array -> unit	CCArray.fast_sort : ('a -> 'a -> int) -> 'a array -> unit	BatArray.fast_sort : ('a -> 'a -> int) -> 'a array -> unit	
		BatArray.favg : float array -> float	
Array.fill : 'a array -> int -> int -> 'a -> unit	CCArray.fill : 'a array -> int -> int -> 'a -> unit	BatArray.fill : 'a array -> int -> int -> 'a -> unit	
	CCArray.filter : ('a -> bool) -> 'a array -> 'a array	BatArray.filter : ('a -> bool) -> 'a array -> 'a array	
	CCArray.filter_map : ('a -> 'b option) -> 'a array -> 'b array	BatArray.filter_map : ('a -> 'b option) -> 'a array -> 'b array	
		BatArray.filteri : (int -> 'a -> bool) -> 'a array -> 'a array	
		BatArray.find : ('a -> bool) -> 'a array -> 'a	
		BatArray.find_all : ('a -> bool) -> 'a array -> 'a array	
	CCArray.find_idx : ('a -> bool) -> 'a array -> (int * 'a) option		
	CCArray.find_map : ('a -> 'b option) -> 'a array -> 'b option		
	CCArray.find_map_i : (int -> 'a -> 'b option) -> 'a array -> 'b option		
		BatArray.findi : ('a -> bool) -> 'a array -> int	
	CCArray.flat_map : ('a -> 'b array) -> 'a array -> 'b array		
	CCArray.fold : ('a -> 'b -> 'a) -> 'a -> 'b array -> 'a	BatArray.fold : ('a -> 'b -> 'a) -> 'a -> 'b array -> 'a	
	CCArray.fold2 : ('acc -> 'a -> 'b -> 'acc) -> 'acc -> 'a array -> 'b array -> 'acc		
Array.fold_left : ('a -> 'b -> 'a) -> 'a -> 'b array -> 'a	CCArray.fold_left : ('a -> 'b -> 'a) -> 'a -> 'b array -> 'a	BatArray.fold_left : ('a -> 'b -> 'a) -> 'a -> 'b array -> 'a	
		BatArray.fold_lefti : ('a -> int -> 'b -> 'a) -> 'a -> 'b array -> 'a	
	CCArray.fold_map : ('acc -> 'a -> 'acc * 'b) -> 'acc -> 'a array -> 'acc * 'b array		
Array.fold_right : ('b -> 'a -> 'a) -> 'b array -> 'a -> 'a	CCArray.fold_right : ('b -> 'a -> 'a) -> 'b array -> 'a -> 'a	BatArray.fold_right : ('b -> 'a -> 'a) -> 'b array -> 'a -> 'a	
		BatArray.fold_righti : (int -> 'b -> 'a -> 'a) -> 'b array -> 'a -> 'a	
	CCArray.fold_while : ('a -> 'b -> 'a * ['Continue   'Stop]) -> 'a -> 'b array -> 'a		
		BatArray.fold_while : ('acc -> 'a -> bool) -> ('acc -> 'a -> 'acc) -> 'acc -> 'a array -> 'acc * int	
	CCArray.foldi : ('a -> int -> 'b -> 'a) -> 'a -> 'b array -> 'a		
Array.for_all : ('a -> bool) -> 'a array -> bool	CCArray.for_all : ('a -> bool) -> 'a array -> bool	BatArray.for_all : ('a -> bool) -> 'a array -> bool	
Array.for_all2 : ('a -> 'b -> bool) -> 'a array -> 'b array -> bool	CCArray.for_all2 : ('a -> 'b -> bool) -> 'a array -> 'b array -> bool	BatArray.for_all2 : ('a -> 'b -> bool) -> 'a array -> 'b array -> bool	
		BatArray.fsum : float array -> float	
Array.get : 'a array -> int -> 'a	CCArray.get : 'a array -> int -> 'a	BatArray.get : 'a array -> int -> 'a	
	CCArray.get_safe : 'a array -> int -> 'a option		

Stdlib	Containers	Batteries	Base
		BatArray.head : 'a array -> int -> 'a array	
Array.init : int -> (int -> 'a) -> 'a array	CCArray.init : int -> (int -> 'a) -> 'a array	BatArray.init : int -> (int -> 'a) -> 'a array	
		BatArray.insert : 'a array -> 'a -> int -> 'a array	
		BatArray.is_sorted_by : ('a -> 'b) -> 'a array -> bool	
Array.iter : ('a -> unit) -> 'a array -> unit	CCArray.iter : ('a -> unit) -> 'a array -> unit	BatArray.iter : ('a -> unit) -> 'a array -> unit	
Array.iter2 : ('a -> 'b -> unit) -> 'a array -> 'b array -> unit	CCArray.iter2 : ('a -> 'b -> unit) -> 'a array -> 'b array -> unit	BatArray.iter2 : ('a -> 'b -> unit) -> 'a array -> 'b array -> unit	
		BatArray.iter2i : (int -> 'a -> 'b -> unit) -> 'a array -> 'b array -> unit	
Array.iteri : (int -> 'a -> unit) -> 'a array -> unit	CCArray.iteri : (int -> 'a -> unit) -> 'a array -> unit	BatArray.iteri : (int -> 'a -> unit) -> 'a array -> unit	
		BatArray.kahan_sum : float array -> float	
		BatArray.left : 'a array -> int -> 'a array	
Array.length : 'a array -> int	CCArray.length : 'a array -> int	BatArray.length : 'a array -> int	
	CCArray.lookup : cmp:'a CCArray.ord -> 'a -> 'a array -> int option		
	CCArray.lookup_exn : cmp:'a CCArray.ord -> 'a -> 'a array -> int		
Array.make : int -> 'a -> 'a array	CCArray.make : int -> 'a -> 'a array	BatArray.make : int -> 'a -> 'a array	
Array.make_float : int -> float array	CCArray.make_float : int -> float array	BatArray.make_float : int -> float array	
Array.make_matrix : int -> int -> 'a -> 'a array array	CCArray.make_matrix : int -> int -> 'a -> 'a array array	BatArray.make_matrix : int -> int -> 'a -> 'a array array	
Array.map : ('a -> 'b) -> 'a array -> 'b array	CCArray.map : ('a -> 'b) -> 'a array -> 'b array	BatArray.map : ('a -> 'b) -> 'a array -> 'b array	
Array.map2 : ('a -> 'b -> 'c) -> 'a array -> 'b array -> 'c array	CCArray.map2 : ('a -> 'b -> 'c) -> 'a array -> 'b array -> 'c array	BatArray.map2 : ('a -> 'b -> 'c) -> 'a array -> 'b array -> 'c array	
Array.mapi : (int -> 'a -> 'b) -> 'a array -> 'b array	CCArray.mapi : (int -> 'a -> 'b) -> 'a array -> 'b array	BatArray.mapi : (int -> 'a -> 'b) -> 'a array -> 'b array	
		BatArray.max : 'a array -> 'a	
Array.mem : 'a -> 'a array -> bool	CCArray.mem : ?eq:('a -> 'a -> bool) -> 'a -> 'a array -> bool	BatArray.mem : 'a -> 'a array -> bool	
Array.memq : 'a -> 'a array -> bool	CCArray.memq : 'a -> 'a array -> bool	BatArray.memq : 'a -> 'a array -> bool	
		BatArray.min : 'a array -> 'a	
		BatArray.min_max : 'a array -> 'a * 'a	
		BatArray.modify : ('a -> 'a) -> 'a array -> unit	
		BatArray.modifyi : (int -> 'a -> 'a) -> 'a array -> unit	
	CCArray.monoid_product : ('a -> 'b -> 'c) -> 'a array -> 'b array -> 'c array		
		BatArray.of_backwards : 'a BatEnum.t -> 'a array	
		BatArray.of_enum : 'a BatEnum.t -> 'a array	
Array.of_list : 'a list -> 'a array	CCArray.of_list : 'a list -> 'a array	BatArray.of_list : 'a list -> 'a array	
Array.of_seq : 'a Seq.t -> 'a array	CCArray.of_seq : 'a Seq.t -> 'a array	BatArray.of_seq : 'a Seq.t -> 'a array	
		BatArray.ord : 'a BatOrd.ord -> 'a array BatOrd.ord	
		BatArray.partition : ('a -> bool) -> 'a array -> 'a array * 'a array	
		BatArray.pivot_split : 'a BatOrd.ord -> 'a array -> 'a -> int * int	
	CCArray.pp : ?pp_start:unit CCArray.printer -> ?pp_stop:unit CCArray.printer -> ? pp_sep:unit CCArray.printer -> 'a CCArray.printer -> 'a array CCArray.printer		
	CCArray.pp_i : ?pp_start:unit CCArray.printer -> ?pp_stop:unit CCArray.printer -> ? pp_sep:unit CCArray.printer -> (int -> 'a CCArray.printer) -> 'a array CCArray.printer		

Stdlib	Containers	Batteries	Base
		BatArray.print : ?first:string -> ?last:string -> ?sep:string -> ('a, 'b) BatIO.printer -> ('a array, 'b) BatIO.printer	
	CCArray.random : 'a CCArray.random_gen -> 'a array CCArray.random_gen		
	CCArray.random_choose : 'a array -> 'a CCArray.random_gen		
	CCArray.random_len : int -> 'a CCArray.random_gen -> 'a array CCArray.random_gen		
	CCArray.random_non_empty : 'a CCArray.random_gen -> 'a array CCArray.random_gen		
		BatArray.range : 'a array -> int BatEnum.t	
		BatArray.reduce : ('a -> 'a -> 'a) -> 'a array -> 'a	
		BatArray.remove_at : int -> 'a array -> 'a array	
	CCArray.rev : 'a array -> 'a array	BatArray.rev : 'a array -> 'a array	
	CCArray.reverse_in_place : 'a array -> unit	BatArray.rev_in_place : 'a array -> unit	
		BatArray.right : 'a array -> int -> 'a array	
	CCArray.scan_left : ('acc -> 'a -> 'acc) -> 'acc -> 'a array -> 'acc array		
Array.set : 'a array -> int -> 'a -> unit	CCArray.set : 'a array -> int -> 'a -> unit	BatArray.set : 'a array -> int -> 'a -> unit	
	CCArray.shuffle : 'a array -> unit	BatArray.shuffle : ?state:Random.State.t -> 'a array -> unit	
	CCArray.shuffle_with : Random.State.t -> 'a array -> unit		
		BatArray.singleton : 'a -> 'a array	
Array.sort : ('a -> 'a -> int) -> 'a array -> unit	CCArray.sort : ('a -> 'a -> int) -> 'a array -> unit	BatArray.sort : ('a -> 'a -> int) -> 'a array -> unit	
	CCArray.sort_generic : (module CCArray.MONO_ARRAY with type elt = 'elt and type t = 'arr) -> cmp:('elt -> 'elt -> int) -> 'arr -> unit		
	CCArray.sort_indices : ('a -> 'a -> int) -> 'a array -> int array		
	CCArray.sort_ranking : ('a -> 'a -> int) -> 'a array -> int array		
	CCArray.sorted : ('a -> 'a -> int) -> 'a array -> 'a array		
		BatArray.split : ('a * 'b) array -> 'a array * 'b array	
Array.stable_sort : ('a -> 'a -> int) -> 'a array -> unit	CCArray.stable_sort : ('a -> 'a -> int) -> 'a array -> unit	BatArray.stable_sort : ('a -> 'a -> int) -> 'a array -> unit	
Array.sub : 'a array -> int -> int -> 'a array	CCArray.sub : 'a array -> int -> int -> 'a array	BatArray.sub : 'a array -> int -> int -> 'a array	
		BatArray.sum : int array -> int	
	CCArray.swap : 'a array -> int -> int -> unit		
		BatArray.tail : 'a array -> int -> 'a array	
	CCArray.to_gen : 'a array -> 'a CCArray.gen		
	CCArray.to_iter : 'a array -> 'a CCArray.iter		
Array.to_list : 'a array -> 'a list	CCArray.to_list : 'a array -> 'a list	BatArray.to_list : 'a array -> 'a list	
Array.to_seq : 'a array -> 'a Seq.t	CCArray.to_seq : 'a array -> 'a Seq.t	BatArray.to_seq : 'a array -> 'a Seq.t	
Array.to_seqi : 'a array -> (int * 'a) Seq.t	CCArray.to_seqi : 'a array -> (int * 'a) Seq.t	BatArray.to_seqi : 'a array -> (int * 'a) Seq.t	
	CCArray.to_string : ?sep:string -> ('a -> string) -> 'a array -> string		
Array.unsafe_get : 'a array -> int -> 'a	CCArray.unsafe_get : 'a array -> int -> 'a	BatArray.unsafe_get : 'a array -> int -> 'a	
Array.unsafe_set : 'a array -> int -> 'a -> unit	CCArray.unsafe_set : 'a array -> int -> 'a -> unit	BatArray.unsafe_set : 'a array -> int -> 'a -> unit	