EAC+dC 1 02 06 Inday		bool IsPowerOf2 < <i>T</i> >	#define UINT8_O_FROM_UINT16(w)	DateTime((uint32_t nYear,		EAFixed16MulDivSafe
EAStdC 1.03.06 Index	EAAlignment.h	uint32_t RoundUpToPowerOf2 <t></t>	#define UINT8_1_FROM_UINT16 (w)		uint32_t nMonth,		EAFixed16Mod EAFixed16ModSafe
namespace EA::StdC	#define EAAlignOf	<pre>bool IsPowerOf2Multiple<t,int n=""> bool IsPowerOf2Minus1<t></t></t,int></pre>	#define UINT8_0_FROM_UINT32 (d)		<pre>uint32_t nDayOfMonth, uint32 t nHour = 0,</pre>		EAFixed16MbdSaTe EAFixed16Abs
	size t AlignOf <t></t>	bool CrossesPower0f2 <t></t>	#define UINT8 1 FROM UINT32(d)		uint32 t nMinute = 0,	E/II IACUIO	EM IXCUIONDS
v1.0	size t AlignOf <t></t>	bool CrossesPowerOf2 < <i>T</i> , int <i>n</i> >	#define UINT8_2_FROM_UINT32(d)		uint32_t nSecond = 0)		int upShiftInt,
	8	uint32_t GetHighestBitPowerOf2	#define UINT8_3_FROM_UINT32(d)	1			ShiftInt, int upMulInt,
Luc Isaak	<pre>T AlignUp<t, a="" size_t=""></t,></pre>	uint32_t GetHighestBitPowerOf2	#4-Cinc HINTO O FROM HINTS (/-)	int	Compare(const DateTime&, bool bCompDate,	int downl	
EAStdC written and maintained by:	T* AlignUp <t, a="" size_t=""></t,>	<pre>T GetNextGreaterEven<t> T GetNextGreaterOdd<t></t></t></pre>	#define UINT8_0_FROM_UINT64(q) #define UINT8 1 FROM UINT64(q)		bool bCompTime)	struct FPTemp	ıate
Paul Pedriana	T AlignUp <t> T* AlignUp<t></t></t>	T RoundUpTo <t, int="" n=""></t,>	#define UINT8_2_FROM_UINT64(q)		boot beomptime)	void	FromFixed
rauireulialia	T AlignDown <t, a="" size="" t=""></t,>	int32 t RoundUpToEx <t, int="" n=""></t,>	#define UINT8_3_FROM_UINT64 (q)	uint32_t	GetParameter	T	AsFixed
	T* AlignDown <t, a="" size="" t=""></t,>	T RoundDownTo <t, int="" n=""></t,>	#define UINT8_4_FROM_UINT64 (q)		SetParameter	int	AsInt
FAAP	T AlignDown <t></t>	T RoundDownToEx <t, int="" n=""></t,>	#define UINT8_5_FROM_UINT64(q)		Set(TimeFrame)	unsigned int	
EAAlignment.h1	T* AlignDown <t></t>	RoundUpToMultiple <pre>T</pre>	#define UINT8_6_FROM_UINT64(q) #define UINT8 7 FROM UINT64(q)	void	<pre>Set(uint32_t nYear, uint32_t nMonth,</pre>	long	AsLong g AsUnsignedLong
EABitTricks.h1	<pre>size_t GetAlignment<t> bool IsAligned<t, a="" size_t=""></t,></t></pre>	uint32 t Log2	#define dinis_/_FROM_dinis4(q)		uint32_t nDayOfMonth,	float	AsFloat
	bool IsAligned<7> Size_t a>	uint32 t CeilLog2	#define UINT16 0 FROM UINT32 (d)		uint32_t nHour,	double	AsDouble
EAByteCrackers.h1	15/1118		#define UINT16_1_FROM_UINT32 (d)		uint32_t nMinute,		
eactype.h1	template <t, count,<="" int="" td=""><td>// Overflow</td><td>III 6: UTNESS & FROM UTNESS ()</td><td>yoid</td><td>uint32_t nSecond)</td><td>bool</td><td>operator<</td></t,>	// Overflow	III 6: UTNESS & FROM UTNESS ()	yoid	uint32_t nSecond)	bool	operator<
	int alignment>	bool	#define UINT16_0_FROM_UINT64(q)	void int64 t	AddTime GetSeconds	bool bool	operator> operator>=
EADateTime.h1	class AlignedĀrray	SignedAdditionWouldOverflow <t> bool</t>	#define UINT16_1_FROM_UINT64(q) #define UINT16_2_FROM_UINT64(q)	}		bool	operator<=
EAFixedPoint.h1	<pre>template<t, alignment="" int=""></t,></pre>	SignedSubtractionWouldOverflow <t></t>	#define UINT16 3 FROM UINT64(q)			bool	operator==
	class AlignedObject	bool		bool	IsLeapYear	bool	operator!=
EAGlobal.h2			#define UINT16_FROM_UINT8 (b1, b0)	uint32_t	GetDaysInYear	FPTemplate	operator~
EAHashCRC.h2	uint16_t ReadMisalignedUint16	bool	#define HINTER O FROM HINT(4/a)	uint32_t uint32 t	GetDaysInMonth GetDayOfYear	FPTemplate	operator-
	uint32_t ReadMisalignedUint32 uint64 t ReadMisalignedUint64	UnsignedSubtractionWouldOverflow <t>bool UnsignedMultiplyWouldOverflow</t>	#define UINT32_0_FROM_UINT64(q)	int	Convert4DigitTo2DigitYear	FPTemplate FPTemplate&	operator+ operator+=
EAHashString.h2	void WriteMisalignedUint16	bool SignedMultiplyWouldOverflow		int	Convert2DigitTo4DigitYear	FPTemplate&	operator-=
EAMathHelp.h2	void WriteMisalignedUint32	bool UnsignedDivisionWouldOverflow	#define UINT32_FROM_UINT8(b3, b2,	uint32_t	GetCurrent	FPTemplate&	operator*=
<u>. </u>	void WriteMisalignedUint64	bool SignedDivisionWouldOverflow	b1, b0)	bool	IsDST	FPTemplate&	operator/=
EAMemory.h2	EABitTricks.h	int GetAverage	#define UINT32_FROM_UINT16 (w1, w0)	int64_t int64 t	GetDaylightSavingsBias GetTimeZoneBias	FPTemplate&	operator%=
	LADICITICKS.II	int GetAverage_Ceiling		11104_0	detiimezonebias	FPTemplate&	operator =
FARandom h 2			I#detine UINT64 FROM UINT8(b7. b6.			I FPTemplateX	onerator%=
EARandom.h2	// Bit manipulation	// Miscellaneous	#define UINT64_FROM_UINT8 (b7, b6, b5, b4,	void	DateTimeToTm	FPTemplate& FPTemplate&	operator&= operator^=
EARandom.h2 EARandomDistribution.h2	<pre>TurnOffLowestBit<t></t></pre>	// Miscellaneous int GetParity	b5, b4, b3, b2,	void	TmToDateTime	FPTemplate& FPTemplate&	
EARandomDistribution.h2	<pre>T TurnOffLowestBit<t> T IsolateLowestBit<t></t></t></pre>	int GetParity bool GetIsBigEndian	b5, b4, b3, b2, b1, b0)	void void	TmToDateTime DateTimeToFileTime	FPTemplate& FPTemplate& FPTemplate	operator^= operator^= operator<<
EARandomDistribution.h	TurnOffLowestBit⟨T⟩	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2,	void void void	TmToDateTime DateTimeToFileTime FileTimeToDateTime	FPTemplate& FPTemplate& FPTemplate FPTemplate	operator^= operator^= operator<< operator>>
EARandomDistribution.h2	<pre>T TurnOffLowestBit<t> T IsolateLowestBit<t> T IsolateLowestOBit<t></t></t></t></pre>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t></t>	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16 (w3, w2, w1, w0)	void void	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate&	<pre>operator^= operator<= operator<< operator>> operator<<=</pre>
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2	<pre>TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBit<tounditc< td=""><td>int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 7 ToggleBetweenIntegers<7> bool IsBetweenOAndValue</td><td>b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0)</td><td>void void void void</td><td>TmToDateTime DateTimeToFileTime FileTimeToDateTime</td><td>FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate&</td><td>operator^= operator<= operator<< operator>> operator<<= operator>>=</td></tounditc<></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></pre>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 7 ToggleBetweenIntegers<7> bool IsBetweenOAndValue	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0)	void void void void	TmToDateTime DateTimeToFileTime FileTimeToDateTime	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate&	operator^= operator<= operator<< operator>> operator<<= operator>>=
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2	TurnOffLowestBit⟨T⟩	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t></t>	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16 (w3, w2, w1, w0)	void void void void void bool	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator==	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate&	<pre>operator^= operator<= operator<< operator>> operator<<=</pre>
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2	<pre>TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBit TurnOffLowe</t></t></t></t></t></t></t></t></pre>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign</t></t></t>	b5, b4, b3, b2, b1, b0) b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h	void void void void void bool bool	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator== operator>	FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate	operator^= operator^= operator<< operator>> operator>> operator>>= operator>> operator operator
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	<pre>TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBit<t> TurnOffLowestBitDownward<t> TurnOffLowestContiguousBits<t> TurnOntLowestBitDownward<t> TurnOntLowestBit<t> TurnOntLowestBit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnon< td=""><td>int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers<t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSignEx</t></t></t></td><td>b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int Isalnum int Isalpha</td><td>void void void void void bool bool bool</td><td><pre>TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator== operator> operator>=</pre></td><td>FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate&</td><td>operator^= operator<< operator>> operator>> operator>> operator>> operator>+ operator-+ operator</td></turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnontlowestbit<turnon<></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></t></pre>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSignEx</t></t></t>	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int Isalnum int Isalpha	void void void void void bool bool bool	<pre>TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator== operator> operator>=</pre>	FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate&	operator^= operator<< operator>> operator>> operator>> operator>> operator>+ operator-+ operator
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2	<pre>TurnOffLowestBit<t> T</t></pre>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSignEx int32_t SignExtend12</t></t></t>	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void bool bool bool bool	<pre>TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator== operator> operator>= operator<</pre>	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate	operator^= operator<= operator<< operator<> operator<>= operator<>= operator++ operator operator operator
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	<pre>TurnOffLowestBit<t> T</t></pre>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSignEx</t></t></t>	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void bool bool bool	<pre>TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator== operator> operator>=</pre>	FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate	operator^= operator^= operator<< operator>> operator>> operator>>= operator>> operator operator
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T TurnOffLowestBitDownward<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> unsigned GetNextWithEqualBitCount unsigned IsolateSingleBits Unsigned IsolateSingleBits</t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EATsUnsigned</t></t></t></t>	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void void bool bool bool bool bool bool	<pre>TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator== operator> operator> operator< operator<= operator!=</pre>	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate	operator^= operator^= operator<< operator>> operator>> operator>>= operator>+ operator++ operator operator Derator Abs DivSafe DivSafeAssign
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	<pre>TurnOffLowestBit<t> T</t></pre>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> EALSUnsigned bool IsTwosComplement</t></t></t></t>	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int Isalnum int Isalpha int Isdigit int Isxdigit int Isgraph int Islower int Isupper	void void void void void bool bool bool bool bool bool bool	<pre>TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator== operator> operator> operator< operator<= operator<= operator<!-- operator!= </pre--></pre>	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate STATIC FPTEMPLATE FPTEMPLATE FPTEMPLATE FPTEMPLATE FPTEMPLATE FPTEMPLATE STATIC FPTEMPLATE FPTEMPLATE STATIC FPTEMPLATE STATIC FPTEMPLATE FPTEMPLA	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafe DivSafeAssign FixedMul
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailing1AndOBits<t> T GetTrailing1AndOBits<t> T PropogateLowestBitDownward<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> unsigned GetNextWithEqualBitCount unsigned IsolateSingleBits unsigned IsolateSingleOBits unsigned int CountTrailingOBits</t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned bool IsTwosComplement bool IsOnesComplement</t></t></t>	b5, b4, b3, b2, b1, b0) #define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void bool bool bool bool bool bool bool boo	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToDateTime SystemTimeToDateTime operator== operator> operator> operator<< operator<= operator Dint.h mt32_t EAFixed16</td <td>FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FTTemplate FTTemplate FTTemplate Static T static T</td> <td>operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator operator Abs DivSafe DivSafeAssign FixedMul FixedDiv</td>	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FTTemplate FTTemplate FTTemplate Static T static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator operator Abs DivSafe DivSafeAssign FixedMul FixedDiv
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailing1AndOBits<t> T FropogateLowestBitDownward<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> T TurnOnLowestOBit<t> unsigned GetNextWithEqualBitCount unsigned IsolateSingleBits unsigned IsolateSingleOBits unsigned IsolateSingleOAnd1Bits int32_t ShiffRightSigned int CountTrailingOBits int CountLeadingOBits</t></t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EAIsUnsigned bool IsTwosComplement bool IsSignMagnitude</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int Isalpha int Isalpha int Isdigit int Isxdigit int Isydigit int Isydigit int Islower int Isupper int Isupper int Isprint int Isprint int Ispunct	void void void void void bool bool bool bool bool typedef ir // 16:16 fixe	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToDateTime SystemTimeToDateTime operator== operator> operator> operator<= operator operator!= bint.h ht32_t EAFixed16 ed point (16 bits of fraction)</td <td>FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FTEmplate FTEmplate FTEmplate FTEmplate Static T Static T</td> <td>operator^= operator^= operator<< operator>> operator>> operator>>= operator>>= operator-+ operator operator Abs DivSafe DivSafeAssign FixedMul FixedDiv FixedDiv FixedDivSafe</td>	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FTEmplate FTEmplate FTEmplate FTEmplate Static T Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator>>= operator-+ operator operator Abs DivSafe DivSafeAssign FixedMul FixedDiv FixedDiv FixedDivSafe
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailing1AndOBits<t> T PropogateLowestBitDownward<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> Unsigned GetNextWithEqualBitCount unsigned unsigned unsigned IsolateSingleBits unsigned int32 t ShiftRightSigned int CountTrailingOBits int CountBits</t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EAIsUnsigned bool IsTwosComplement bool IsOnesComplement bool IsOffsetBinary</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int Isalnum int Isalpha int Isdigit int Isxdigit int Isyaper int Isupper int Isprint int Isprint int Ispunct int Isspace	void void void void void void bool bool bool bool bool bool typedef ir // 16:16 fixe typedef FF	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator== operator> operator> operator<- operator<= operator!= Dint.h nt32_t EAFixed16 ed point (16 bits of fraction) PTemplate<> SFixed16	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FTTemplate FTTemplate FTTemplate Static T static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator operator Abs DivSafe DivSafeAssign FixedMul FixedDiv
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailing1AndOBits<t> T FropogateLowestBitDownward<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> T TurnOnLowestOBit<t> unsigned GetNextWithEqualBitCount unsigned IsolateSingleBits unsigned IsolateSingleOBits unsigned IsolateSingleOAnd1Bits int32_t ShiffRightSigned int CountTrailingOBits int CountLeadingOBits</t></t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EAIsUnsigned bool IsTwosComplement bool IsSignMagnitude</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int Isalnum int Isalpha int Isdigit int Isxdigit int Isyaper int Isupper int Isprint int Isprint int Ispunct int Isspace	void void void void bool bool bool bool bool bool typedef ir // 16:16 fixe typedef FF	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToDateTime DateTimeToDateTime SystemTimeToDateTime operator= operator> operator> operator< operator(= operat	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMod
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailing1AndOBits<t> T PropogateLowestBitDownward<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> Unsigned GetNextWithEqualBitCount unsigned unsigned unsigned IsolateSingleBits unsigned int32 t ShiftRightSigned int CountTrailingOBits int CountBits</t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EAISUnsigned bool IsTwosComplement bool IsSignMagnitude bool IsOffsetBinary #define EAArrayCount #define EAOffsetOf</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int Isalpha int Isdigit int Isxdigit int Isydigit int Isuper int Isuper int Isuper int Isprint int Isprint int Isprint int Isprint int Ispace int Isscatel int Isscati	void void void void void bool bool bool bool bool typedef ir // 16:16 fixe typedef FF typedef FF	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToDateTime DateTimeToDateTime SystemTimeToDateTime operator== operator>- operator>- operator<- operator!= Sint.h nt32_t	FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FTEmplate FTEmplate FTEmplate FTEmplate FTEmplate Static T Static T Static T Static T	operator^= operator^= operator<< operator<> operator<> operator>> operator>> operator++ operator operator Abs DivSafe DivSafe DivSafeAssign FixedMul FixedDivSafe FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit(T) T IsolateLowestBit(T) T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailing1AndOBits<t> T PropogateLowestBitDownward<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> unsigned GetNextWithEqualBitCount unsigned IsolateSingleBits unsigned IsolateSingleOBits unsigned IsolateSingleOBits int CountTrailingOBits int CountBits int CountBits int CountBits int CountBits64 uint32_t RotateLeft uint32_t RotateRight</t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int GetSignExtend12 int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EAISUnsigned tool IsTwosComplement bool IsOnesComplement bool IsOffsetBinary #define EAArrayCount #define EAOffsetOf EAByteCrackers.h</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void bool bool bool bool bool typedef if typedef FF typedef FF #define #define	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime DateTimeToSystemTime SystemTimeToDateTime operator= operator> operator> operator< operator operator!= bint.h nt32_t EAFixed16 ed point (16 bits of fraction) PTemplate< SFixed16 PTemplate<> UFixed16 EAFixed16ToInt EAIntToFixed16	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMod
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingIAndOBits<t> T TurnOffLowestBitDownward<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> unsigned GetNextWithEqualBitCount unsigned IsolateSingleBits unsigned IsolateSingleOBits unsigned IsolateSingleOBits int CountTrailingOBits int CountBits int CountBits</t></t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EAIsUnsigned bool IsOnesComplement bool IsOnesComplement bool IsOffsetBinary #define EAArrayCount #define EAOffsetOf EAByteCrackers.h Conventions:</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int Isalnum int Isalpha int Isdigit int Isxdigit int Isyace int Isupper int Isprint	void void void void void bool bool bool bool typedef FF typedef FF #define #define #define	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime DateTimeToSystemTime SystemTimeToDateTime operator= operator> operator> operator< operator<= operator operator!= pint.h ptag2</td <td>FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T Static T Static T Static T Static T Static T Static T</td> <td>operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMod</td>	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMod
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> T TurnOnLowestOBit<t> Unsigned GetNextWithEqualBitCount unsigned IsolateSingleOBits unsigned IsolateSingleOBits unsigned IsolateSingleOMnd1Bits int32_t CountFailingOBits int CountTrailingOBits int CountBits Int Coun</t></t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EALSUnsigned bool IsTwosComplement bool IsSignMagnitude bool IsOffsetBinary #define EAArrayCount #define EAOffsetOf EAByteCrackers.h Conventions: 0 (zero) refers to the lowest byte,</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void bool bool bool bool bool typedef if typedef FF typedef FF #define #define	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToSystemTime DateTimeToSystemTime SystemTimeToDateTime operator= operator> operator> operator< operator operator!= bint.h nt32_t EAFixed16 ed point (16 bits of fraction) PTemplate< SFixed16 PTemplate<> UFixed16 EAFixed16ToInt EAIntToFixed16	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMulDivSafe FixedMod
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> T TurnOnLowestOBit<t> Unsigned GetNextWithEqualBitCount unsigned IsolateSingleOBits unsigned IsolateSingleOBits unsigned IsolateSingleOAnd1Bits int32_t ShiftRightSigned int CountTrailingOBits int CountBits int Count</t></t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> Hool IsTwosComplement bool IsOnesComplement bool IsOffsetBinary #define EATATOWN #def</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int Isalnum int Isalpha int Isdigit int Isxdigit int Isyace int Isupper int Isprint	void void void void void bool bool bool bool bool typedef in typedef Ff typedef Ff typedef Ff typedef ine #define #define #define #define #define	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator= operator> operator> operator< operator operator!= Jint.h nt32_t EAFixed16 ed point (16 bits of fraction) PTemplate< SFixed16 PTemplate<> UFixed16 EAFixed16ToInt EAIntToFixed16 EAFixed16ToDouble EADoubleToFixed16 EAFixed16ToFloat EAFloatToFixed16	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMulDivSafe FixedMod
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingIAndOBits<t> T GetTrailingIAndOBits<t> T TurnOffLowestContiguousBits<t> T TurnOffLowestContiguousBits<t> Unsigned GetNextWithEqualBitCount unsigned IsolateSingleBits unsigned IsolateSingleOBits unsigned IsolateSingleOBits int CountTrailingOBits int CountBits int CountBits</t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> Hool IsTwosComplement bool IsOnesComplement bool IsOffsetBinary #define EATATOWN #def</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void bool bool bool bool bool typedef ir // 16:16 fixe typedef FF #define #define #define #define #define	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToDateTime DateTimeToDateTime SystemTimeToDateTime operator= operator> operator> operator< operator(= operator) poperator(= op	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMulDivSafe FixedMod
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingIAndOBits<t> T GetTrailingIAndOBits<t> T TurnOffLowestContiguousBits<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> unsigned GetNextWithEqualBitCount unsigned IsolateSingleBits unsigned IsolateSingleOBits unsigned IsolateSingleOBits int CountTrailingOBits int CountBits Int Co</t></t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EAIsUnsigned bool IsTwosComplement bool IsSonesComplement bool IsSignMagnitude bool IsSignMagnitude bool IsOffsetBinary #define EAArrayCount #define EAOffsetOf EAByteCrackers.h Conventions: 0 (zero) refers to the lowest byte, 1 refers to the second lowest byte, etc. d b means 8 bit byte w means 16 bit word d means 32 bit dword</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, b1, b0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void bool bool bool bool bool typedef ir // 16:16 fixe typedef Ff #define #define #define #define #define #define #define	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator= operator> operator> operator= operator(a) operator!= Dint.h htt32_t EAFixed16 ed point (16 bits of fraction) PTemplate<> SFixed16 PTemplate<> Ufixed16 EAFixed16ToInt EAIntToFixed16 EAFixed16ToFloat EAFixed16ToFloat EAFixed16ToFloat EAFloatToFixed16 EAFixed16Negate	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMulDivSafe FixedMod
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T TurnOffLowestContiguousBits<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> Unsigned GetNextWithEqualBitCount unsigned IsolateSingleOBits unsigned IsolateSingleOBits unsigned IsolateSingleOAnd1Bits int32_t ShiftRightSigned int CountTrailingOBits int CountBits Int CountBit</t></t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 7 ToggleBetweenIntegers bool IsBetweenOAndValue void ExchangeValues 7 FloorMod int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned bool IsSunexcomplement bool IsSunexcomplement bool IsSignMagnitude bool IsOffsetBinary #define EAArrayCount #define EAOffsetOf EAByteCrackers.h Conventions: 0 (zero) refers to the lowest byte, 1 refers to the second lowest byte, etc. d b means 8 bit byte w means 16 bit word	#define UINT64_FROM_UINT16(w3, w2, w1, w0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void bool bool bool bool bool typedef ir // 16:16 fixe typedef FF #define	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToDateTime DateTimeToDateTime SystemTimeToDateTime operator= operator> operator> operator(= operator!= pint.h nt32_t EAFixed16 ed point (16 bits of fraction) PTemplate<> SFixed16 PTemplate<> UFixed16 EAFixed16ToInt EAIntToFixed16 EAFixed16ToDouble EADoubleToFixed16 EAFixed16ToFloat EAFloatToFixed16 EAFixed16Negate EAFixed16Negate EAFixed16Mul	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMulDivSafe FixedMod
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingIAndOBits<t> T GetTrailingIAndOBits<t> T TurnOffLowestContiguousBits<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> unsigned GetNextWithEqualBitCount unsigned IsolateSingleBits unsigned IsolateSingleOBits unsigned IsolateSingleOBits int CountTrailingOBits int CountBits Int Co</t></t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EAIsUnsigned bool IsTwosComplement bool IsSonesComplement bool IsSignMagnitude bool IsSignMagnitude bool IsOffsetBinary #define EAArrayCount #define EAOffsetOf EAByteCrackers.h Conventions: 0 (zero) refers to the lowest byte, 1 refers to the second lowest byte, etc. d b means 8 bit byte w means 16 bit word d means 32 bit dword</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, b1, b0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void bool bool bool bool bool typedef ir // 16:16 fixe typedef Ff #define #define #define #define #define #define #define	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToDateTime DateTimeToDateTime SystemTimeToDateTime operator= operator> operator> operator< operator<= operator!= Dint.h mt32_t EAFixed16 ed point (16 bits of fraction) PTemplate<> SFixed16 PTemplate<> UFixed16 EAFixed16ToInt EAIntToFixed16 EAFixed16ToDouble EADoubleToFixed16 EAFixed16ToFloat EAFixed16ToFloat EAFloatToFixed16 EAFixed16Mul EAFixed16Div EAFixed16Div EAFixed16Div EAFixed16DivSafe	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator>> operator>> operator>>= operator++ operator operator Abs DivSafe DivSafeBuivSafe FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMulDivSafe FixedMod
EARandomDistribution.h 2 EAScanf.h 2 EASprintf.h 2 EAString.h 2 EATextUtil.h 2	TurnOffLowestBit <t> T IsolateLowestBit<t> T IsolateLowestBit<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T GetTrailingOBits<t> T TurnOffLowestContiguousBits<t> T TurnOffLowestContiguousBits<t> T TurnOnLowestOBit<t> Unsigned GetNextWithEqualBitCount unsigned IsolateSingleOBits unsigned IsolateSingleOBits unsigned IsolateSingleOAnd1Bits int32_t ShiftRightSigned int CountTrailingOBits int CountBits Int CountBit</t></t></t></t></t></t></t></t></t></t>	int GetParity bool GetIsBigEndian int ToggleBetweenOAnd1 T ToggleBetweenIntegers <t> bool IsBetweenOAndValue void ExchangeValues<t> T FloorMod<t> int GetSign int GetSign int GetSignEx int32_t SignExtend12 int32_t SignExtend24 bool IsUnsigned<t> #define EAIsUnsigned bool IsTwosComplement bool IsSonesComplement bool IsSignMagnitude bool IsSignMagnitude bool IsOffsetBinary #define EAArrayCount #define EAOffsetOf EAByteCrackers.h Conventions: 0 (zero) refers to the lowest byte, 1 refers to the second lowest byte, etc. d b means 8 bit byte w means 16 bit word d means 32 bit dword</t></t></t></t>	#define UINT64_FROM_UINT16(w3, w2, b1, b0) #define UINT64_FROM_UINT32(d1, d0) eactype.h int	void void void void void bool bool bool bool bool typedef in typedef if typedef FF typedef FF typedef ine #define #define #define #define #define #define #define #define #define	TmToDateTime DateTimeToFileTime FileTimeToDateTime DateTimeToDateTime DateTimeToDateTime DateTimeToSystemTime SystemTimeToDateTime operator= operator> operator> operator< operator operator! bint.h mt32_t</td <td>FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T Static T Static T Static T Static T Static T Static T</td> <td>operator^= operator^= operator<< operator<> operator<> operator<>= operator>> operator++ operator operator Abs DivSafe DivSafeAssign FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMod</td>	FPTemplate& FPTemplate& FPTemplate FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate& FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate FPTemplate Static T	operator^= operator^= operator<< operator<> operator<> operator<>= operator>> operator++ operator operator Abs DivSafe DivSafeAssign FixedMul FixedDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDiv FixedMulDivSafe FixedMod

	bool	EACconf h	int Strnicmp	uint64 t GetPartUint64	// Logical operators
5161111	bool IsDenormalized	EAScanf.h	int StrcmpAlnum	void SetPartUint8	int128 t operator^
EAGlobal.h	bool IsDenormalized	int Cscanf	int StricmpAlnum	void SetPartUint16	int128 t operator
template <t, kglobalid="</td" t="" uint32=""><td></td><td>int Fscanf</td><td></td><td>void SetPartUint32</td><td>int128 t operator&</td></t,>		int Fscanf		void SetPartUint32	int128 t operator&
T::kGlobalId>	EAMemory.h	int Scanf	char t* EcvtBuf	void SetPartUint64	int128 t& operator^=
class GlobalPtr	void* alloca	int Sscanf	char_t* FcvtBuf		int128_t& operator =
	uint8 t* Memset8	int Vcscanf		bool IsZero	int128_t& operator&=
<pre>struct OSGlobalNode : public</pre>	uint16 t* Memset16	int Vfscanf	char_t* I32toa	void SetZero	
EA::StdC::intrusive_list_node	uint32 ⁻ t* Memset32	int Vscanf	char_t* U32toa	void TwosComplement	// Equality operators
	uint64 ⁻ t* Memset64	int Vsscanf	char_t* I64toa	void InverseTwosComplement	int compare
typedef OSGlobalNode	void* MemsetN	EASprintf.h	char_t* U64toa	}	bool operator==
*(*OSGlobalFactoryPtr)()	void* MemsetPointer	•	double Strtod	1	bool operator!=
0001 10 10 0 0 0 0 0 0	const char_t* Memchr	int Cprintf	double StrtodEnglish	class int128_t : int128_t_base	bool operator>
OSGlobalNode* GetOSGlobal	int Memcmp	int Fprintf	int32_t StrtoI32	class uint128_t: int128_t_base	bool operator>=
bool SetOSGlobal	char_t* Memcpy	int Printf	uint32_t StrtoU32	int128 t()	bool operator (
bool ReleaseOSGlobal	char_t* Memmove	int Sprintf	int64_t StrtoI64 uint64_t StrtoU64	int128_t(uint32_t nPart0,	bool operator<=
template <t, id="" t="" uint32=""></t,>	EARandom.h	<pre>int Snprintf int Vcprintf</pre>	dinto4_t 3titto004	uint32 t nPart1,	// Operators to convert back to basis tunes
class AutoOSGlobalPtr		int Vfprintf	int32 t AtoI32	uint32 t nPart2,	// Operators to convert back to basic types int8 t AsInt8
Class Autoosdiobaliti	void GetRandomSeed	int Vprintf	uint32 t AtoU32	uint32 t nPart3)	int16 t AsInt16
template <t, id="" t="" uint32=""></t,>	1 2 1 1 2 2 1 1 1	int Vsprintf	int64 t AtoI64	int128 t(uint64 t nPart0,	int32 t AsInt32
class AutoStaticOSGlobalPtr	class RandomLinearCongruential	int Vsnprintf	uint64 t AtoU64	uint64 t nPart1)	int64 t AsInt64
	class RandomMersenneTwister	'	double Atof	int128 t(int8 t value)	float AsFloat
EAHashCRC.h	{ uint32 t GetSeed	EAString.h	double AtofEnglish	int128_t(uint8_t value)	double AsDouble
uint16 t CRC16		// Consider using Strlcpy as a safe	char t* Ftoa	<pre>int128_t(int16_t value)</pre>	
uint32 t CRC24	<pre>void SetSeed uint32 t operator()</pre>	alternative to Strcpy or Strncpy	char_t* FtoaEnglish	<pre>int128_t(uint16_t value)</pre>	// Misc. Functions
uint32 t CRC32	uint32 t RandomUint32Uniform	char t* Strcpy	<pre>size_t ReduceFloatString</pre>	int128_t(int32_t value)	void Negate
uint32 t CRC32Reverse	double RandomDoubleUniform	char t* Strncpy	EATextUtil.h	int128_t(uint32_t value)	bool IsNegative
uint64 t CRC64	}	size_t Strlcpy	EATEXIOUI.II	int128_t(int64_t value)	bool IsPositive
_	· ·	_ ''	bool WildcardMatch	int128_t(uint64_t value) int128_t(float value)	void Modulus
EAHashString.h	typedef RandomLinearCongruential	// Consider using Strlcat as a safe	<pre>bool ParseDelimitedText</pre>	int128_t(float value) int128_t(double value)	
uint32 t FNV1	typedef RandomLinearCongruential	alternative to Strcat, StringnCat or Strncat	<pre>void ConvertBinaryDataToASCIIArray</pre>	int128 t(const char t*,	// String conversion functions
uint32 t FNV1_String8	Random	char_t* Strcat	bool ConvertASCIIArrayToBinaryData	int nBase = 10)	int128_t StrToInt128
uint32_t FNV1_String16	RandomSmall	char_t* Strncat	const char_t*	1110 115050 107	void Int128ToStr
	RandomFast	char_t* StringnCat	GetTextLine	// Unary arithmetic/logic operators	}
uint32_t DJB2	typedef RandomMersenneTwister		haal CulitTakamDalimitad	int128 t operator-	// Dinamy apparators
uint32_t DJB2_String8	RandomQuality		bool SplitTokenDelimited bool SplitTokenSeparated	int128 t& operator++	// Binary operators int128 t
uint32_t DJB2_String16	EARandomDistribution.h	size_t Strlen	boot Spiitiokenseparateu	int128_t& operator	uint128 t operator+
		size_t StrlenUTF8Decoded size t StrlenUTF8Encoded	int BoyerMooreSearch	int128_t operator++	operator-
// Compile-time string hash. Generates	bool RandomBool <random></random>	size_t StrlenUTF8Encoded	,	int128_t operator	operator*
FNV1 string hashes via compile-time	int32_t Random2 <random></random>	char t* Strend	Int128_t.h	int128_t operator~	operator/
template expansion. Both single byte and	int32_t Random4 <random> int32_t Random8<random></random></random>	char t* Strdup	class int128 t base	int128_t operator+	operator%
Unicode characters are accepted.	int32_t Random16 <random></random>	void Strdel	{	use it	operator^
template <>	int32 t Random32 <random></random>	char t* Strupr	void operatorPlus	// Math operators	operator
class CTStringHash	int32 t Random64 <random></random>	char t* Strlwr	void operatorMinus	int128_t operator+ int128_t operator-	operator&
	int32 t Random128 <random></random>	char_t* Strmix	void operatorMul	int128 t operator*	. ,
EAMathHelp.h	int32 t Random256 <random></random>	char_t* Strchr	void operatorShiftRight	int128 t operator/	int compare
uint32 t RoundToUint32	int32_t RandomPowerOfTwo <random< td=""><td>size_t Strcspn</td><td>void operatorShiftLeft</td><td>int128 t operator%</td><td>bool operator== bool operator!=</td></random<>	size_t Strcspn	void operatorShiftLeft	int128 t operator%	bool operator== bool operator!=
int32 t RoundToInt32	int32_t	size_t Strcspn			bool operator!= bool operator>
int32 t FloorToInt32	RandomInt32UniformRange <random></random>	char_t* Strpbrk	bool operator!	int128 t& operator+=	bool operator>=
	Idouble	char_t* Strrchr	void operatorXOR	int128_t& operator-=	bool operators
int32_t CeilToInt32	double		void operatorOR	1	
int32_t Cell101nt32 int32_t TruncateToInt32	RandomDoubleUniformRange <random></random>	size_t Strspn	· ·	int128_t& operator*=	hool operator<=
<pre>int32_t TruncateToInt32 int32_t FastRoundToInt23</pre>	<pre>RandomDoubleUniformRange<random> uint32_t</random></pre>	char_t* Strstr	void operatorAND	int128_t& operator/=	bool operator<=
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random></random></random>	char_t* Strstr char_t* Stristr	void operatorAND		bool operator<=
<pre>int32_t TruncateToInt32 int32_t FastRoundToInt23</pre>	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random> int32_t</random></random>	char_t* Strstr char_t* Stristr char_t* Strrstr	void operatorAND bool AsBool	int128_t& operator/= int128_t& operator%=	
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8 uint8_t ClampUnitFloatToUint8	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random> int32_t RandomInt32GaussianRange<random></random></random></random>	char_t* Strstr char_t* Stristr char_t* Stristr char_t* Strrstr char_t* Strirstr	void operatorAND bool AsBool uint8_t AsUint8	int128_t& operator/= int128_t& operator%= // Shift operators	const int128_t INT128_MIN const int128_t INT128_MAX
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8 uint8_t ClampUnitFloatToUint8 bool IsInvalid	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random> int32_t RandomInt32GaussianRange<random> Float</random></random></random>	char_t* Strstr char_t* Stristr char_t* Strrstr char_t* Stristr char_t* Stristr char_t* Striok	void operatorAND bool AsBool uint8 t AsUint8 uint16_t AsUint16	int128_t& operator/= int128_t& operator%= // Shift operators int128_t operator>>	const int128_t INT128_MIN const int128_t INT128_MAX const uint128_t UINT128_MIN
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8 uint8_t ClampUnitFloatToUint8 bool IsInvalid bool IsInvalid	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random> int32_t RandomInt32GaussianRange<random> Float RandomFloatGaussianRange<random,< td=""><td>char_t* Strstr char_t* Stristr char_t* Stristr char_t* Strrstr char_t* Strirstr</td><td>void operatorAND bool AsBool uint8_t AsUint8 uint16_t AsUint16 uint32_t AsUint32</td><td><pre>int128_t& operator/= int128_t& operator%= // Shift operators int128_t operator>> int128_t operator<</pre></td><td>const int128_t INT128_MIN const int128_t INT128_MAX</td></random,<></random></random></random>	char_t* Strstr char_t* Stristr char_t* Stristr char_t* Strrstr char_t* Strirstr	void operatorAND bool AsBool uint8_t AsUint8 uint16_t AsUint16 uint32_t AsUint32	<pre>int128_t& operator/= int128_t& operator%= // Shift operators int128_t operator>> int128_t operator<</pre>	const int128_t INT128_MIN const int128_t INT128_MAX
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8 uint8_t ClampUnitFloatToUint8 bool IsInvalid bool IsNormal	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random> int32_t RandomInt32GaussianRange<random> Float RandomFloatGaussianRange<random, float=""></random,></random></random></random>	char_t* Strstr char_t* Stristr char_t* Strrstr char_t* Strrstr char_t* Strirstr char_t* Strick const char_t*	void operatorAND bool AsBool uint8 t AsUint8 uint16_t AsUint16	<pre>int128_t& operator/= int128_t& operator%= // Shift operators int128_t operator>> int128_t operator<< int128_t& operator>>=</pre>	const int128 t INT128 MIN const int128 t INT128 MAX const uint128 t UINT128 MIN const uint128 t UINT128 MAX
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8 uint8_t ClampUnitFloatToUint8 bool IsInvalid bool IsNozmal bool IsNozmal	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random> int32_t RandomInt32GaussianRange<random> Float RandomFloatGaussianRange<random, float=""> int32_t</random,></random></random></random>	char_t* Strstr char_t* Stristr char_t* Stristr char_t* Strirstr char_t* Strirstr char_t* Strirstr char_t* Strick const char_t* Strtok2	void operatorAND bool AsBool uint8_t AsUint8 uint16_t AsUint16 uint32_t AsUint32	<pre>int128_t& operator/= int128_t& operator%= // Shift operators int128_t operator>> int128_t operator<</pre>	const int128_t INT128_MIN const int128_t INT128_MAX const uint128_t UINT128_MIN const uint128_t UINT128_MAX #define INT128_C(x)
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8 uint8_t ClampUnitFloatToUint8 bool IsInvalid bool IsNormal bool IsNormal bool IsNOrmal bool IsNAN	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random> int32_t RandomInt32GaussianRange<random> Float RandomFloatGaussianRange<random, float=""></random,></random></random></random>	char_t* Strstr char_t* Stristr char_t* Stristr char_t* Stristr char_t* Stristr char_t* Stristr char_t* Strtok const char_t* Strtok2 char_t* Strset	void operatorAND bool AsBool uint8_t AsUint8 uint16_t AsUint16 uint32_t AsUint32 uint64_t AsUint64	<pre>int128_t& operator/= int128_t& operator%= // Shift operators int128_t operator>> int128_t operator<< int128_t& operator>>=</pre>	const int128 t INT128 MIN const int128 t INT128 MAX const uint128 t UINT128 MIN const uint128 t UINT128 MAX
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8 uint8_t ClampUnitFloatToUint8 bool IsInvalid bool IsInvalid bool IsNormal bool IsNormal bool IsNormal bool IsNAN bool IsNAN	RandomDoubleUniformRange <random> uint32_t RandomUnit32WeightedChoice<random> int32_t RandomInt32GaussianRange<random> Float RandomFloatGaussianRange<random, float=""> int32_t RandomInt32TriangleRange<random></random></random,></random></random></random>	<pre>char_t* Strstr char_t* Stristr char_t* Strrstr char_t* Strrstr char_t* Stristr char_t* Stristr char_t* Strtok const char_t*</pre>	void operatorAND bool AsBool uint8_t AsUint8 uint16_t AsUint16 uint32_t AsUint32 uint64_t AsUint64 int GetBit void SetBit uint8_t GetPartUint8	<pre>int128_t& operator/= int128_t& operator%= // Shift operators int128_t operator>> int128_t operator<< int128_t& operator>>=</pre>	const int128_t INT128_MIN const int128_t INT128_MAX const uint128_t UINT128_MIN const uint128_t UINT128_MAX #define INT128_C(x)
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8 uint8_t ClampUnitFloatToUint8 bool IsInvalid bool IsNormal bool IsNormal bool IsNormal bool IsNoN bool IsNAN bool IsNAN bool IsNAN	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random> int32_t RandomInt32GaussianRange<random> Float RandomFloatGaussianRange<random, float=""> int32_t RandomInt32TriangleRange<random> Float</random></random,></random></random></random>	<pre>char_t* Strstr char_t* Stristr char_t* Stristr char_t* Strirstr char_t* Strirstr char_t* Striok const char_t*</pre>	void operatorAND bool AsBool uint8_t AsUint8 uint16_t AsUint16 uint32_t AsUint12 uint64_t AsUint64 int GetBit void SetBit uint8_t GetPartUint8 uint16_t GetPartUint16	<pre>int128_t& operator/= int128_t& operator%= // Shift operators int128_t operator>> int128_t operator<< int128_t& operator>>=</pre>	const int128_t INT128_MIN const int128_t INT128_MAX const uint128_t UINT128_MIN const uint128_t UINT128_MAX #define INT128_C(x)
int32_t TruncateToInt32 int32_t FastRoundToInt23 uint8_t UnitFloatToUint8 uint8_t ClampUnitFloatToUint8 bool IsInvalid bool IsNormal bool IsNormal bool IsNormal bool IsNAN bool IsNAN	RandomDoubleUniformRange <random> uint32_t RandomUint32WeightedChoice<random> int32_t RandomInt32GaussianRange<random> Float RandomFloatGaussianRange<random, float=""> int32_t RandomInt32TriangleRange<random> Float RandomFloatTriangleRange<random> Float RandomFloatTriangleRange<random></random></random></random></random,></random></random></random>	<pre>char_t* Strstr char_t* Stristr char_t* Strrstr char_t* Strrstr char_t* Stristr char_t* Stristr char_t* Strtok const char_t*</pre>	void operatorAND bool AsBool uint8_t AsUint8 uint16_t AsUint16 uint32_t AsUint32 uint64_t AsUint64 int GetBit void SetBit uint8_t GetPartUint8	<pre>int128_t& operator/= int128_t& operator%= // Shift operators int128_t operator>> int128_t operator<< int128_t& operator>>=</pre>	const int128_t INT128_MIN const int128_t INT128_MAX const uint128_t UINT128_MIN const uint128_t UINT128_MAX #define INT128_C(x)

EAStdC Index