/\* ----------------------------------------------------------------------------

\* This file was automatically generated by SWIG (http://www.swig.org).

\* Version 4.0.2

\*

\* Do not make changes to this file unless you know what you are doing--modify

\* the SWIG interface file instead.

\* ----------------------------------------------------------------------------- \*/

package com.twitter.ann.faiss;

import com.twitter.ann.faiss.NativeUtils;

public class swigfaissJNI {

static {

try {

if (NativeUtils.getOperatingSystemType() == NativeUtils.OSType.MacOS) {

NativeUtils.loadLibraryFromJar("/com/twitter/ann/faiss/swig/resources/swigfaiss.dylib");

} else {

NativeUtils.loadLibraryFromJar("/com/twitter/ann/faiss/swig/resources/libstdc++.so.6");

NativeUtils.loadLibraryFromJar("/com/twitter/ann/faiss/swig/resources/libgcc\_s.so.1");

NativeUtils.loadLibraryFromJar("/com/twitter/ann/faiss/swig/resources/libgomp.so.1");

NativeUtils.loadLibraryFromJar("/com/twitter/ann/faiss/swig/resources/libquadmath.so.0");

NativeUtils.loadLibraryFromJar("/com/twitter/ann/faiss/swig/resources/libgfortran.so.5");

NativeUtils.loadLibraryFromJar("/com/twitter/ann/faiss/swig/resources/swigfaiss.so");

}

} catch (Exception e) {

System.err.println("Native code library failed to load. \n" + e);

System.exit(1);

}

}

public final static native long new\_intArray(int jarg1);

public final static native void delete\_intArray(long jarg1);

public final static native int intArray\_getitem(long jarg1, intArray jarg1\_, int jarg2);

public final static native void intArray\_setitem(long jarg1, intArray jarg1\_, int jarg2, int jarg3);

public final static native long intArray\_cast(long jarg1, intArray jarg1\_);

public final static native long intArray\_frompointer(long jarg1);

public final static native long new\_floatArray(int jarg1);

public final static native void delete\_floatArray(long jarg1);

public final static native float floatArray\_getitem(long jarg1, floatArray jarg1\_, int jarg2);

public final static native void floatArray\_setitem(long jarg1, floatArray jarg1\_, int jarg2, float jarg3);

public final static native long floatArray\_cast(long jarg1, floatArray jarg1\_);

public final static native long floatArray\_frompointer(long jarg1);

public final static native long new\_longArray(int jarg1);

public final static native void delete\_longArray(long jarg1);

public final static native long longArray\_getitem(long jarg1, longArray jarg1\_, int jarg2);

public final static native void longArray\_setitem(long jarg1, longArray jarg1\_, int jarg2, long jarg3);

public final static native long longArray\_cast(long jarg1, longArray jarg1\_);

public final static native long longArray\_frompointer(long jarg1);

public final static native long new\_doubleArray(int jarg1);

public final static native void delete\_doubleArray(long jarg1);

public final static native double doubleArray\_getitem(long jarg1, doubleArray jarg1\_, int jarg2);

public final static native void doubleArray\_setitem(long jarg1, doubleArray jarg1\_, int jarg2, double jarg3);

public final static native long doubleArray\_cast(long jarg1, doubleArray jarg1\_);

public final static native long doubleArray\_frompointer(long jarg1);

public final static native long new\_FloatVector();

public final static native void FloatVector\_push\_back(long jarg1, FloatVector jarg1\_, float jarg2);

public final static native void FloatVector\_clear(long jarg1, FloatVector jarg1\_);

public final static native long FloatVector\_data(long jarg1, FloatVector jarg1\_);

public final static native long FloatVector\_size(long jarg1, FloatVector jarg1\_);

public final static native float FloatVector\_at(long jarg1, FloatVector jarg1\_, long jarg2);

public final static native void FloatVector\_resize(long jarg1, FloatVector jarg1\_, long jarg2);

public final static native void FloatVector\_reserve(long jarg1, FloatVector jarg1\_, long jarg2);

public final static native void FloatVector\_swap(long jarg1, FloatVector jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native void delete\_FloatVector(long jarg1);

public final static native long new\_DoubleVector();

public final static native void DoubleVector\_push\_back(long jarg1, DoubleVector jarg1\_, double jarg2);

public final static native void DoubleVector\_clear(long jarg1, DoubleVector jarg1\_);

public final static native long DoubleVector\_data(long jarg1, DoubleVector jarg1\_);

public final static native long DoubleVector\_size(long jarg1, DoubleVector jarg1\_);

public final static native double DoubleVector\_at(long jarg1, DoubleVector jarg1\_, long jarg2);

public final static native void DoubleVector\_resize(long jarg1, DoubleVector jarg1\_, long jarg2);

public final static native void DoubleVector\_reserve(long jarg1, DoubleVector jarg1\_, long jarg2);

public final static native void DoubleVector\_swap(long jarg1, DoubleVector jarg1\_, long jarg2, DoubleVector jarg2\_);

public final static native void delete\_DoubleVector(long jarg1);

public final static native long new\_ByteVector();

public final static native void ByteVector\_push\_back(long jarg1, ByteVector jarg1\_, short jarg2);

public final static native void ByteVector\_clear(long jarg1, ByteVector jarg1\_);

public final static native long ByteVector\_data(long jarg1, ByteVector jarg1\_);

public final static native long ByteVector\_size(long jarg1, ByteVector jarg1\_);

public final static native short ByteVector\_at(long jarg1, ByteVector jarg1\_, long jarg2);

public final static native void ByteVector\_resize(long jarg1, ByteVector jarg1\_, long jarg2);

public final static native void ByteVector\_reserve(long jarg1, ByteVector jarg1\_, long jarg2);

public final static native void ByteVector\_swap(long jarg1, ByteVector jarg1\_, long jarg2, ByteVector jarg2\_);

public final static native void delete\_ByteVector(long jarg1);

public final static native long new\_CharVector();

public final static native void CharVector\_push\_back(long jarg1, CharVector jarg1\_, char jarg2);

public final static native void CharVector\_clear(long jarg1, CharVector jarg1\_);

public final static native String CharVector\_data(long jarg1, CharVector jarg1\_);

public final static native long CharVector\_size(long jarg1, CharVector jarg1\_);

public final static native char CharVector\_at(long jarg1, CharVector jarg1\_, long jarg2);

public final static native void CharVector\_resize(long jarg1, CharVector jarg1\_, long jarg2);

public final static native void CharVector\_reserve(long jarg1, CharVector jarg1\_, long jarg2);

public final static native void CharVector\_swap(long jarg1, CharVector jarg1\_, long jarg2, CharVector jarg2\_);

public final static native void delete\_CharVector(long jarg1);

public final static native long new\_Uint64Vector();

public final static native void Uint64Vector\_push\_back(long jarg1, Uint64Vector jarg1\_, long jarg2);

public final static native void Uint64Vector\_clear(long jarg1, Uint64Vector jarg1\_);

public final static native long Uint64Vector\_data(long jarg1, Uint64Vector jarg1\_);

public final static native long Uint64Vector\_size(long jarg1, Uint64Vector jarg1\_);

public final static native long Uint64Vector\_at(long jarg1, Uint64Vector jarg1\_, long jarg2);

public final static native void Uint64Vector\_resize(long jarg1, Uint64Vector jarg1\_, long jarg2);

public final static native void Uint64Vector\_reserve(long jarg1, Uint64Vector jarg1\_, long jarg2);

public final static native void Uint64Vector\_swap(long jarg1, Uint64Vector jarg1\_, long jarg2, Uint64Vector jarg2\_);

public final static native void delete\_Uint64Vector(long jarg1);

public final static native long new\_LongVector();

public final static native void LongVector\_push\_back(long jarg1, LongVector jarg1\_, long jarg2);

public final static native void LongVector\_clear(long jarg1, LongVector jarg1\_);

public final static native long LongVector\_data(long jarg1, LongVector jarg1\_);

public final static native long LongVector\_size(long jarg1, LongVector jarg1\_);

public final static native long LongVector\_at(long jarg1, LongVector jarg1\_, long jarg2);

public final static native void LongVector\_resize(long jarg1, LongVector jarg1\_, long jarg2);

public final static native void LongVector\_reserve(long jarg1, LongVector jarg1\_, long jarg2);

public final static native void LongVector\_swap(long jarg1, LongVector jarg1\_, long jarg2, LongVector jarg2\_);

public final static native void delete\_LongVector(long jarg1);

public final static native long new\_IntVector();

public final static native void IntVector\_push\_back(long jarg1, IntVector jarg1\_, int jarg2);

public final static native void IntVector\_clear(long jarg1, IntVector jarg1\_);

public final static native long IntVector\_data(long jarg1, IntVector jarg1\_);

public final static native long IntVector\_size(long jarg1, IntVector jarg1\_);

public final static native int IntVector\_at(long jarg1, IntVector jarg1\_, long jarg2);

public final static native void IntVector\_resize(long jarg1, IntVector jarg1\_, long jarg2);

public final static native void IntVector\_reserve(long jarg1, IntVector jarg1\_, long jarg2);

public final static native void IntVector\_swap(long jarg1, IntVector jarg1\_, long jarg2, IntVector jarg2\_);

public final static native void delete\_IntVector(long jarg1);

public final static native long new\_VectorTransformVector();

public final static native void VectorTransformVector\_push\_back(long jarg1, VectorTransformVector jarg1\_, long jarg2, VectorTransform jarg2\_);

public final static native void VectorTransformVector\_clear(long jarg1, VectorTransformVector jarg1\_);

public final static native long VectorTransformVector\_data(long jarg1, VectorTransformVector jarg1\_);

public final static native long VectorTransformVector\_size(long jarg1, VectorTransformVector jarg1\_);

public final static native long VectorTransformVector\_at(long jarg1, VectorTransformVector jarg1\_, long jarg2);

public final static native void VectorTransformVector\_resize(long jarg1, VectorTransformVector jarg1\_, long jarg2);

public final static native void VectorTransformVector\_reserve(long jarg1, VectorTransformVector jarg1\_, long jarg2);

public final static native void VectorTransformVector\_swap(long jarg1, VectorTransformVector jarg1\_, long jarg2, VectorTransformVector jarg2\_);

public final static native void delete\_VectorTransformVector(long jarg1);

public final static native long new\_OperatingPointVector();

public final static native void OperatingPointVector\_push\_back(long jarg1, OperatingPointVector jarg1\_, long jarg2, OperatingPoint jarg2\_);

public final static native void OperatingPointVector\_clear(long jarg1, OperatingPointVector jarg1\_);

public final static native long OperatingPointVector\_data(long jarg1, OperatingPointVector jarg1\_);

public final static native long OperatingPointVector\_size(long jarg1, OperatingPointVector jarg1\_);

public final static native long OperatingPointVector\_at(long jarg1, OperatingPointVector jarg1\_, long jarg2);

public final static native void OperatingPointVector\_resize(long jarg1, OperatingPointVector jarg1\_, long jarg2);

public final static native void OperatingPointVector\_reserve(long jarg1, OperatingPointVector jarg1\_, long jarg2);

public final static native void OperatingPointVector\_swap(long jarg1, OperatingPointVector jarg1\_, long jarg2, OperatingPointVector jarg2\_);

public final static native void delete\_OperatingPointVector(long jarg1);

public final static native long new\_InvertedListsPtrVector();

public final static native void InvertedListsPtrVector\_push\_back(long jarg1, InvertedListsPtrVector jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native void InvertedListsPtrVector\_clear(long jarg1, InvertedListsPtrVector jarg1\_);

public final static native long InvertedListsPtrVector\_data(long jarg1, InvertedListsPtrVector jarg1\_);

public final static native long InvertedListsPtrVector\_size(long jarg1, InvertedListsPtrVector jarg1\_);

public final static native long InvertedListsPtrVector\_at(long jarg1, InvertedListsPtrVector jarg1\_, long jarg2);

public final static native void InvertedListsPtrVector\_resize(long jarg1, InvertedListsPtrVector jarg1\_, long jarg2);

public final static native void InvertedListsPtrVector\_reserve(long jarg1, InvertedListsPtrVector jarg1\_, long jarg2);

public final static native void InvertedListsPtrVector\_swap(long jarg1, InvertedListsPtrVector jarg1\_, long jarg2, InvertedListsPtrVector jarg2\_);

public final static native void delete\_InvertedListsPtrVector(long jarg1);

public final static native long new\_FloatVectorVector();

public final static native void FloatVectorVector\_push\_back(long jarg1, FloatVectorVector jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native void FloatVectorVector\_clear(long jarg1, FloatVectorVector jarg1\_);

public final static native long FloatVectorVector\_data(long jarg1, FloatVectorVector jarg1\_);

public final static native long FloatVectorVector\_size(long jarg1, FloatVectorVector jarg1\_);

public final static native long FloatVectorVector\_at(long jarg1, FloatVectorVector jarg1\_, long jarg2);

public final static native void FloatVectorVector\_resize(long jarg1, FloatVectorVector jarg1\_, long jarg2);

public final static native void FloatVectorVector\_reserve(long jarg1, FloatVectorVector jarg1\_, long jarg2);

public final static native void FloatVectorVector\_swap(long jarg1, FloatVectorVector jarg1\_, long jarg2, FloatVectorVector jarg2\_);

public final static native void delete\_FloatVectorVector(long jarg1);

public final static native long new\_ByteVectorVector();

public final static native void ByteVectorVector\_push\_back(long jarg1, ByteVectorVector jarg1\_, long jarg2, ByteVector jarg2\_);

public final static native void ByteVectorVector\_clear(long jarg1, ByteVectorVector jarg1\_);

public final static native long ByteVectorVector\_data(long jarg1, ByteVectorVector jarg1\_);

public final static native long ByteVectorVector\_size(long jarg1, ByteVectorVector jarg1\_);

public final static native long ByteVectorVector\_at(long jarg1, ByteVectorVector jarg1\_, long jarg2);

public final static native void ByteVectorVector\_resize(long jarg1, ByteVectorVector jarg1\_, long jarg2);

public final static native void ByteVectorVector\_reserve(long jarg1, ByteVectorVector jarg1\_, long jarg2);

public final static native void ByteVectorVector\_swap(long jarg1, ByteVectorVector jarg1\_, long jarg2, ByteVectorVector jarg2\_);

public final static native void delete\_ByteVectorVector(long jarg1);

public final static native long new\_LongVectorVector();

public final static native void LongVectorVector\_push\_back(long jarg1, LongVectorVector jarg1\_, long jarg2);

public final static native void LongVectorVector\_clear(long jarg1, LongVectorVector jarg1\_);

public final static native long LongVectorVector\_data(long jarg1, LongVectorVector jarg1\_);

public final static native long LongVectorVector\_size(long jarg1, LongVectorVector jarg1\_);

public final static native long LongVectorVector\_at(long jarg1, LongVectorVector jarg1\_, long jarg2);

public final static native void LongVectorVector\_resize(long jarg1, LongVectorVector jarg1\_, long jarg2);

public final static native void LongVectorVector\_reserve(long jarg1, LongVectorVector jarg1\_, long jarg2);

public final static native void LongVectorVector\_swap(long jarg1, LongVectorVector jarg1\_, long jarg2, LongVectorVector jarg2\_);

public final static native void delete\_LongVectorVector(long jarg1);

public final static native void bitvec\_print(long jarg1, long jarg2);

public final static native void fvecs2bitvecs(long jarg1, long jarg2, long jarg3, long jarg4);

public final static native void bitvecs2fvecs(long jarg1, long jarg2, long jarg3, long jarg4);

public final static native void fvec2bitvec(long jarg1, long jarg2, long jarg3);

public final static native void bitvec\_shuffle(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void BitstringWriter\_code\_set(long jarg1, BitstringWriter jarg1\_, long jarg2);

public final static native long BitstringWriter\_code\_get(long jarg1, BitstringWriter jarg1\_);

public final static native void BitstringWriter\_code\_size\_set(long jarg1, BitstringWriter jarg1\_, long jarg2);

public final static native long BitstringWriter\_code\_size\_get(long jarg1, BitstringWriter jarg1\_);

public final static native void BitstringWriter\_i\_set(long jarg1, BitstringWriter jarg1\_, long jarg2);

public final static native long BitstringWriter\_i\_get(long jarg1, BitstringWriter jarg1\_);

public final static native long new\_BitstringWriter(long jarg1, long jarg2);

public final static native void BitstringWriter\_write(long jarg1, BitstringWriter jarg1\_, long jarg2, int jarg3);

public final static native void delete\_BitstringWriter(long jarg1);

public final static native void BitstringReader\_code\_set(long jarg1, BitstringReader jarg1\_, long jarg2);

public final static native long BitstringReader\_code\_get(long jarg1, BitstringReader jarg1\_);

public final static native void BitstringReader\_code\_size\_set(long jarg1, BitstringReader jarg1\_, long jarg2);

public final static native long BitstringReader\_code\_size\_get(long jarg1, BitstringReader jarg1\_);

public final static native void BitstringReader\_i\_set(long jarg1, BitstringReader jarg1\_, long jarg2);

public final static native long BitstringReader\_i\_get(long jarg1, BitstringReader jarg1\_);

public final static native long new\_BitstringReader(long jarg1, long jarg2);

public final static native long BitstringReader\_read(long jarg1, BitstringReader jarg1\_, int jarg2);

public final static native void delete\_BitstringReader(long jarg1);

public final static native void hamming\_batch\_size\_set(long jarg1);

public final static native long hamming\_batch\_size\_get();

public final static native int popcount64(long jarg1);

public final static native void hammings(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void hammings\_knn\_hc(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5, int jarg6);

public final static native void hammings\_knn(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5, int jarg6);

public final static native void hammings\_knn\_mc(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, long jarg7, long jarg8, LongVector jarg8\_);

public final static native void hamming\_range\_search(long jarg1, long jarg2, long jarg3, long jarg4, int jarg5, long jarg6, long jarg7, RangeSearchResult jarg7\_);

public final static native void hamming\_count\_thres(long jarg1, long jarg2, long jarg3, long jarg4, int jarg5, long jarg6, long jarg7);

public final static native long match\_hamming\_thres(long jarg1, long jarg2, long jarg3, long jarg4, int jarg5, long jarg6, long jarg7, LongVector jarg7\_, long jarg8);

public final static native void crosshamming\_count\_thres(long jarg1, long jarg2, int jarg3, long jarg4, long jarg5);

public final static native int get\_num\_gpus();

public final static native int METRIC\_INNER\_PRODUCT\_get();

public final static native int METRIC\_L2\_get();

public final static native int METRIC\_Canberra\_get();

public final static native String get\_compile\_options();

public final static native double getmillisecs();

public final static native long get\_mem\_usage\_kb();

public final static native long get\_cycles();

public final static native void fvec\_madd(long jarg1, long jarg2, float jarg3, long jarg4, long jarg5);

public final static native int fvec\_madd\_and\_argmin(long jarg1, long jarg2, float jarg3, long jarg4, long jarg5);

public final static native void reflection(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native void matrix\_qr(int jarg1, int jarg2, long jarg3);

public final static native void ranklist\_handle\_ties(int jarg1, long jarg2, LongVector jarg2\_, long jarg3);

public final static native long ranklist\_intersection\_size(long jarg1, long jarg2, LongVector jarg2\_, long jarg3, long jarg4, LongVector jarg4\_);

public final static native long merge\_result\_table\_with\_\_SWIG\_0(long jarg1, long jarg2, long jarg3, LongVector jarg3\_, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, boolean jarg7, long jarg8);

public final static native long merge\_result\_table\_with\_\_SWIG\_1(long jarg1, long jarg2, long jarg3, LongVector jarg3\_, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, boolean jarg7);

public final static native long merge\_result\_table\_with\_\_SWIG\_2(long jarg1, long jarg2, long jarg3, LongVector jarg3\_, long jarg4, long jarg5, LongVector jarg5\_, long jarg6);

public final static native double imbalance\_factor\_\_SWIG\_0(int jarg1, int jarg2, long jarg3, LongVector jarg3\_);

public final static native double imbalance\_factor\_\_SWIG\_1(int jarg1, long jarg2);

public final static native void fvec\_argsort(long jarg1, long jarg2, long jarg3);

public final static native void fvec\_argsort\_parallel(long jarg1, long jarg2, long jarg3);

public final static native int ivec\_hist(long jarg1, long jarg2, int jarg3, long jarg4);

public final static native void bincode\_hist(long jarg1, long jarg2, long jarg3, long jarg4);

public final static native long ivec\_checksum(long jarg1, long jarg2);

public final static native long fvecs\_maybe\_subsample\_\_SWIG\_0(long jarg1, long jarg2, long jarg3, long jarg4, boolean jarg5, long jarg6);

public final static native long fvecs\_maybe\_subsample\_\_SWIG\_1(long jarg1, long jarg2, long jarg3, long jarg4, boolean jarg5);

public final static native long fvecs\_maybe\_subsample\_\_SWIG\_2(long jarg1, long jarg2, long jarg3, long jarg4);

public final static native void binary\_to\_real(long jarg1, long jarg2, long jarg3);

public final static native void real\_to\_binary(long jarg1, long jarg2, long jarg3);

public final static native long hash\_bytes(long jarg1, long jarg2);

public final static native boolean check\_openmp();

public final static native int FAISS\_VERSION\_MAJOR\_get();

public final static native int FAISS\_VERSION\_MINOR\_get();

public final static native int FAISS\_VERSION\_PATCH\_get();

public final static native void Index\_d\_set(long jarg1, Index jarg1\_, int jarg2);

public final static native int Index\_d\_get(long jarg1, Index jarg1\_);

public final static native void Index\_ntotal\_set(long jarg1, Index jarg1\_, long jarg2);

public final static native long Index\_ntotal\_get(long jarg1, Index jarg1\_);

public final static native void Index\_verbose\_set(long jarg1, Index jarg1\_, boolean jarg2);

public final static native boolean Index\_verbose\_get(long jarg1, Index jarg1\_);

public final static native void Index\_is\_trained\_set(long jarg1, Index jarg1\_, boolean jarg2);

public final static native boolean Index\_is\_trained\_get(long jarg1, Index jarg1\_);

public final static native void Index\_metric\_type\_set(long jarg1, Index jarg1\_, int jarg2);

public final static native int Index\_metric\_type\_get(long jarg1, Index jarg1\_);

public final static native void Index\_metric\_arg\_set(long jarg1, Index jarg1\_, float jarg2);

public final static native float Index\_metric\_arg\_get(long jarg1, Index jarg1\_);

public final static native void delete\_Index(long jarg1);

public final static native void Index\_train(long jarg1, Index jarg1\_, long jarg2, long jarg3);

public final static native void Index\_add(long jarg1, Index jarg1\_, long jarg2, long jarg3);

public final static native void Index\_add\_with\_ids(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void Index\_search(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void Index\_range\_search(long jarg1, Index jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, RangeSearchResult jarg5\_);

public final static native void Index\_assign\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void Index\_assign\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void Index\_reset(long jarg1, Index jarg1\_);

public final static native long Index\_remove\_ids(long jarg1, Index jarg1\_, long jarg2, IDSelector jarg2\_);

public final static native void Index\_reconstruct(long jarg1, Index jarg1\_, long jarg2, long jarg3);

public final static native void Index\_reconstruct\_n(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void Index\_search\_and\_reconstruct(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_, long jarg7);

public final static native void Index\_compute\_residual(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void Index\_compute\_residual\_n(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_);

public final static native long Index\_get\_distance\_computer(long jarg1, Index jarg1\_);

public final static native long Index\_sa\_code\_size(long jarg1, Index jarg1\_);

public final static native void Index\_sa\_encode(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void Index\_sa\_decode(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long Index\_toIVF(long jarg1, Index jarg1\_);

public final static native void ClusteringParameters\_niter\_set(long jarg1, ClusteringParameters jarg1\_, int jarg2);

public final static native int ClusteringParameters\_niter\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_nredo\_set(long jarg1, ClusteringParameters jarg1\_, int jarg2);

public final static native int ClusteringParameters\_nredo\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_verbose\_set(long jarg1, ClusteringParameters jarg1\_, boolean jarg2);

public final static native boolean ClusteringParameters\_verbose\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_spherical\_set(long jarg1, ClusteringParameters jarg1\_, boolean jarg2);

public final static native boolean ClusteringParameters\_spherical\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_int\_centroids\_set(long jarg1, ClusteringParameters jarg1\_, boolean jarg2);

public final static native boolean ClusteringParameters\_int\_centroids\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_update\_index\_set(long jarg1, ClusteringParameters jarg1\_, boolean jarg2);

public final static native boolean ClusteringParameters\_update\_index\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_frozen\_centroids\_set(long jarg1, ClusteringParameters jarg1\_, boolean jarg2);

public final static native boolean ClusteringParameters\_frozen\_centroids\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_min\_points\_per\_centroid\_set(long jarg1, ClusteringParameters jarg1\_, int jarg2);

public final static native int ClusteringParameters\_min\_points\_per\_centroid\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_max\_points\_per\_centroid\_set(long jarg1, ClusteringParameters jarg1\_, int jarg2);

public final static native int ClusteringParameters\_max\_points\_per\_centroid\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_seed\_set(long jarg1, ClusteringParameters jarg1\_, int jarg2);

public final static native int ClusteringParameters\_seed\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native void ClusteringParameters\_decode\_block\_size\_set(long jarg1, ClusteringParameters jarg1\_, long jarg2);

public final static native long ClusteringParameters\_decode\_block\_size\_get(long jarg1, ClusteringParameters jarg1\_);

public final static native long new\_ClusteringParameters();

public final static native void delete\_ClusteringParameters(long jarg1);

public final static native void ClusteringIterationStats\_obj\_set(long jarg1, ClusteringIterationStats jarg1\_, float jarg2);

public final static native float ClusteringIterationStats\_obj\_get(long jarg1, ClusteringIterationStats jarg1\_);

public final static native void ClusteringIterationStats\_time\_set(long jarg1, ClusteringIterationStats jarg1\_, double jarg2);

public final static native double ClusteringIterationStats\_time\_get(long jarg1, ClusteringIterationStats jarg1\_);

public final static native void ClusteringIterationStats\_time\_search\_set(long jarg1, ClusteringIterationStats jarg1\_, double jarg2);

public final static native double ClusteringIterationStats\_time\_search\_get(long jarg1, ClusteringIterationStats jarg1\_);

public final static native void ClusteringIterationStats\_imbalance\_factor\_set(long jarg1, ClusteringIterationStats jarg1\_, double jarg2);

public final static native double ClusteringIterationStats\_imbalance\_factor\_get(long jarg1, ClusteringIterationStats jarg1\_);

public final static native void ClusteringIterationStats\_nsplit\_set(long jarg1, ClusteringIterationStats jarg1\_, int jarg2);

public final static native int ClusteringIterationStats\_nsplit\_get(long jarg1, ClusteringIterationStats jarg1\_);

public final static native long new\_ClusteringIterationStats();

public final static native void delete\_ClusteringIterationStats(long jarg1);

public final static native void Clustering\_d\_set(long jarg1, Clustering jarg1\_, long jarg2);

public final static native long Clustering\_d\_get(long jarg1, Clustering jarg1\_);

public final static native void Clustering\_k\_set(long jarg1, Clustering jarg1\_, long jarg2);

public final static native long Clustering\_k\_get(long jarg1, Clustering jarg1\_);

public final static native void Clustering\_centroids\_set(long jarg1, Clustering jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long Clustering\_centroids\_get(long jarg1, Clustering jarg1\_);

public final static native void Clustering\_iteration\_stats\_set(long jarg1, Clustering jarg1\_, long jarg2);

public final static native long Clustering\_iteration\_stats\_get(long jarg1, Clustering jarg1\_);

public final static native long new\_Clustering\_\_SWIG\_0(int jarg1, int jarg2);

public final static native long new\_Clustering\_\_SWIG\_1(int jarg1, int jarg2, long jarg3, ClusteringParameters jarg3\_);

public final static native void Clustering\_train\_\_SWIG\_0(long jarg1, Clustering jarg1\_, long jarg2, long jarg3, long jarg4, Index jarg4\_, long jarg5);

public final static native void Clustering\_train\_\_SWIG\_1(long jarg1, Clustering jarg1\_, long jarg2, long jarg3, long jarg4, Index jarg4\_);

public final static native void Clustering\_train\_encoded\_\_SWIG\_0(long jarg1, Clustering jarg1\_, long jarg2, long jarg3, long jarg4, Index jarg4\_, long jarg5, Index jarg5\_, long jarg6);

public final static native void Clustering\_train\_encoded\_\_SWIG\_1(long jarg1, Clustering jarg1\_, long jarg2, long jarg3, long jarg4, Index jarg4\_, long jarg5, Index jarg5\_);

public final static native void Clustering\_post\_process\_centroids(long jarg1, Clustering jarg1\_);

public final static native void delete\_Clustering(long jarg1);

public final static native long new\_Clustering1D\_\_SWIG\_0(int jarg1);

public final static native long new\_Clustering1D\_\_SWIG\_1(int jarg1, long jarg2, ClusteringParameters jarg2\_);

public final static native void Clustering1D\_train\_exact(long jarg1, Clustering1D jarg1\_, long jarg2, long jarg3);

public final static native void delete\_Clustering1D(long jarg1);

public final static native void ProgressiveDimClusteringParameters\_progressive\_dim\_steps\_set(long jarg1, ProgressiveDimClusteringParameters jarg1\_, int jarg2);

public final static native int ProgressiveDimClusteringParameters\_progressive\_dim\_steps\_get(long jarg1, ProgressiveDimClusteringParameters jarg1\_);

public final static native void ProgressiveDimClusteringParameters\_apply\_pca\_set(long jarg1, ProgressiveDimClusteringParameters jarg1\_, boolean jarg2);

public final static native boolean ProgressiveDimClusteringParameters\_apply\_pca\_get(long jarg1, ProgressiveDimClusteringParameters jarg1\_);

public final static native long new\_ProgressiveDimClusteringParameters();

public final static native void delete\_ProgressiveDimClusteringParameters(long jarg1);

public final static native void delete\_ProgressiveDimIndexFactory(long jarg1);

public final static native long new\_ProgressiveDimIndexFactory();

public final static native void ProgressiveDimClustering\_d\_set(long jarg1, ProgressiveDimClustering jarg1\_, long jarg2);

public final static native long ProgressiveDimClustering\_d\_get(long jarg1, ProgressiveDimClustering jarg1\_);

public final static native void ProgressiveDimClustering\_k\_set(long jarg1, ProgressiveDimClustering jarg1\_, long jarg2);

public final static native long ProgressiveDimClustering\_k\_get(long jarg1, ProgressiveDimClustering jarg1\_);

public final static native void ProgressiveDimClustering\_centroids\_set(long jarg1, ProgressiveDimClustering jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long ProgressiveDimClustering\_centroids\_get(long jarg1, ProgressiveDimClustering jarg1\_);

public final static native void ProgressiveDimClustering\_iteration\_stats\_set(long jarg1, ProgressiveDimClustering jarg1\_, long jarg2);

public final static native long ProgressiveDimClustering\_iteration\_stats\_get(long jarg1, ProgressiveDimClustering jarg1\_);

public final static native long new\_ProgressiveDimClustering\_\_SWIG\_0(int jarg1, int jarg2);

public final static native long new\_ProgressiveDimClustering\_\_SWIG\_1(int jarg1, int jarg2, long jarg3, ProgressiveDimClusteringParameters jarg3\_);

public final static native void ProgressiveDimClustering\_train(long jarg1, ProgressiveDimClustering jarg1\_, long jarg2, long jarg3, long jarg4, ProgressiveDimIndexFactory jarg4\_);

public final static native void delete\_ProgressiveDimClustering(long jarg1);

public final static native float kmeans\_clustering(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native void ProductQuantizer\_d\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2);

public final static native long ProductQuantizer\_d\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_M\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2);

public final static native long ProductQuantizer\_M\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_nbits\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2);

public final static native long ProductQuantizer\_nbits\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_dsub\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2);

public final static native long ProductQuantizer\_dsub\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_code\_size\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2);

public final static native long ProductQuantizer\_code\_size\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_ksub\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2);

public final static native long ProductQuantizer\_ksub\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_verbose\_set(long jarg1, ProductQuantizer jarg1\_, boolean jarg2);

public final static native boolean ProductQuantizer\_verbose\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_train\_type\_set(long jarg1, ProductQuantizer jarg1\_, int jarg2);

public final static native int ProductQuantizer\_train\_type\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_cp\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2, ClusteringParameters jarg2\_);

public final static native long ProductQuantizer\_cp\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_assign\_index\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2, Index jarg2\_);

public final static native long ProductQuantizer\_assign\_index\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_centroids\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long ProductQuantizer\_centroids\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native long ProductQuantizer\_get\_centroids(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void ProductQuantizer\_train(long jarg1, ProductQuantizer jarg1\_, int jarg2, long jarg3);

public final static native long new\_ProductQuantizer\_\_SWIG\_0(long jarg1, long jarg2, long jarg3);

public final static native long new\_ProductQuantizer\_\_SWIG\_1();

public final static native void ProductQuantizer\_set\_derived\_values(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_set\_params(long jarg1, ProductQuantizer jarg1\_, long jarg2, int jarg3);

public final static native void ProductQuantizer\_compute\_code(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void ProductQuantizer\_compute\_codes(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void ProductQuantizer\_compute\_codes\_with\_assign\_index(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void ProductQuantizer\_decode\_\_SWIG\_0(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void ProductQuantizer\_decode\_\_SWIG\_1(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void ProductQuantizer\_compute\_code\_from\_distance\_table(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void ProductQuantizer\_compute\_distance\_table(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void ProductQuantizer\_compute\_inner\_prod\_table(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void ProductQuantizer\_compute\_distance\_tables(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void ProductQuantizer\_compute\_inner\_prod\_tables(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void ProductQuantizer\_search\_\_SWIG\_0(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, boolean jarg7);

public final static native void ProductQuantizer\_search\_\_SWIG\_1(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void ProductQuantizer\_search\_ip\_\_SWIG\_0(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, boolean jarg7);

public final static native void ProductQuantizer\_search\_ip\_\_SWIG\_1(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void ProductQuantizer\_sdc\_table\_set(long jarg1, ProductQuantizer jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long ProductQuantizer\_sdc\_table\_get(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_compute\_sdc\_table(long jarg1, ProductQuantizer jarg1\_);

public final static native void ProductQuantizer\_search\_sdc\_\_SWIG\_0(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, boolean jarg7);

public final static native void ProductQuantizer\_search\_sdc\_\_SWIG\_1(long jarg1, ProductQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void delete\_ProductQuantizer(long jarg1);

public final static native void PQEncoderGeneric\_code\_set(long jarg1, PQEncoderGeneric jarg1\_, long jarg2);

public final static native long PQEncoderGeneric\_code\_get(long jarg1, PQEncoderGeneric jarg1\_);

public final static native void PQEncoderGeneric\_offset\_set(long jarg1, PQEncoderGeneric jarg1\_, short jarg2);

public final static native short PQEncoderGeneric\_offset\_get(long jarg1, PQEncoderGeneric jarg1\_);

public final static native int PQEncoderGeneric\_nbits\_get(long jarg1, PQEncoderGeneric jarg1\_);

public final static native void PQEncoderGeneric\_reg\_set(long jarg1, PQEncoderGeneric jarg1\_, short jarg2);

public final static native short PQEncoderGeneric\_reg\_get(long jarg1, PQEncoderGeneric jarg1\_);

public final static native long new\_PQEncoderGeneric\_\_SWIG\_0(long jarg1, int jarg2, short jarg3);

public final static native long new\_PQEncoderGeneric\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void PQEncoderGeneric\_encode(long jarg1, PQEncoderGeneric jarg1\_, long jarg2);

public final static native void delete\_PQEncoderGeneric(long jarg1);

public final static native void PQEncoder8\_code\_set(long jarg1, PQEncoder8 jarg1\_, long jarg2);

public final static native long PQEncoder8\_code\_get(long jarg1, PQEncoder8 jarg1\_);

public final static native long new\_PQEncoder8(long jarg1, int jarg2);

public final static native void PQEncoder8\_encode(long jarg1, PQEncoder8 jarg1\_, long jarg2);

public final static native void delete\_PQEncoder8(long jarg1);

public final static native void PQEncoder16\_code\_set(long jarg1, PQEncoder16 jarg1\_, long jarg2);

public final static native long PQEncoder16\_code\_get(long jarg1, PQEncoder16 jarg1\_);

public final static native long new\_PQEncoder16(long jarg1, int jarg2);

public final static native void PQEncoder16\_encode(long jarg1, PQEncoder16 jarg1\_, long jarg2);

public final static native void delete\_PQEncoder16(long jarg1);

public final static native void PQDecoderGeneric\_code\_set(long jarg1, PQDecoderGeneric jarg1\_, long jarg2);

public final static native long PQDecoderGeneric\_code\_get(long jarg1, PQDecoderGeneric jarg1\_);

public final static native void PQDecoderGeneric\_offset\_set(long jarg1, PQDecoderGeneric jarg1\_, short jarg2);

public final static native short PQDecoderGeneric\_offset\_get(long jarg1, PQDecoderGeneric jarg1\_);

public final static native int PQDecoderGeneric\_nbits\_get(long jarg1, PQDecoderGeneric jarg1\_);

public final static native long PQDecoderGeneric\_mask\_get(long jarg1, PQDecoderGeneric jarg1\_);

public final static native void PQDecoderGeneric\_reg\_set(long jarg1, PQDecoderGeneric jarg1\_, short jarg2);

public final static native short PQDecoderGeneric\_reg\_get(long jarg1, PQDecoderGeneric jarg1\_);

public final static native long new\_PQDecoderGeneric(long jarg1, int jarg2);

public final static native long PQDecoderGeneric\_decode(long jarg1, PQDecoderGeneric jarg1\_);

public final static native void delete\_PQDecoderGeneric(long jarg1);

public final static native int PQDecoder8\_nbits\_get();

public final static native void PQDecoder8\_code\_set(long jarg1, PQDecoder8 jarg1\_, long jarg2);

public final static native long PQDecoder8\_code\_get(long jarg1, PQDecoder8 jarg1\_);

public final static native long new\_PQDecoder8(long jarg1, int jarg2);

public final static native long PQDecoder8\_decode(long jarg1, PQDecoder8 jarg1\_);

public final static native void delete\_PQDecoder8(long jarg1);

public final static native int PQDecoder16\_nbits\_get();

public final static native void PQDecoder16\_code\_set(long jarg1, PQDecoder16 jarg1\_, long jarg2);

public final static native long PQDecoder16\_code\_get(long jarg1, PQDecoder16 jarg1\_);

public final static native long new\_PQDecoder16(long jarg1, int jarg2);

public final static native long PQDecoder16\_decode(long jarg1, PQDecoder16 jarg1\_);

public final static native void delete\_PQDecoder16(long jarg1);

public final static native void VectorTransform\_d\_in\_set(long jarg1, VectorTransform jarg1\_, int jarg2);

public final static native int VectorTransform\_d\_in\_get(long jarg1, VectorTransform jarg1\_);

public final static native void VectorTransform\_d\_out\_set(long jarg1, VectorTransform jarg1\_, int jarg2);

public final static native int VectorTransform\_d\_out\_get(long jarg1, VectorTransform jarg1\_);

public final static native void VectorTransform\_is\_trained\_set(long jarg1, VectorTransform jarg1\_, boolean jarg2);

public final static native boolean VectorTransform\_is\_trained\_get(long jarg1, VectorTransform jarg1\_);

public final static native void VectorTransform\_train(long jarg1, VectorTransform jarg1\_, long jarg2, long jarg3);

public final static native long VectorTransform\_apply(long jarg1, VectorTransform jarg1\_, long jarg2, long jarg3);

public final static native void VectorTransform\_apply\_noalloc(long jarg1, VectorTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void VectorTransform\_reverse\_transform(long jarg1, VectorTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void delete\_VectorTransform(long jarg1);

public final static native void LinearTransform\_have\_bias\_set(long jarg1, LinearTransform jarg1\_, boolean jarg2);

public final static native boolean LinearTransform\_have\_bias\_get(long jarg1, LinearTransform jarg1\_);

public final static native void LinearTransform\_is\_orthonormal\_set(long jarg1, LinearTransform jarg1\_, boolean jarg2);

public final static native boolean LinearTransform\_is\_orthonormal\_get(long jarg1, LinearTransform jarg1\_);

public final static native void LinearTransform\_A\_set(long jarg1, LinearTransform jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long LinearTransform\_A\_get(long jarg1, LinearTransform jarg1\_);

public final static native void LinearTransform\_b\_set(long jarg1, LinearTransform jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long LinearTransform\_b\_get(long jarg1, LinearTransform jarg1\_);

public final static native long new\_LinearTransform\_\_SWIG\_0(int jarg1, int jarg2, boolean jarg3);

public final static native long new\_LinearTransform\_\_SWIG\_1(int jarg1, int jarg2);

public final static native long new\_LinearTransform\_\_SWIG\_2(int jarg1);

public final static native long new\_LinearTransform\_\_SWIG\_3();

public final static native void LinearTransform\_apply\_noalloc(long jarg1, LinearTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void LinearTransform\_transform\_transpose(long jarg1, LinearTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void LinearTransform\_reverse\_transform(long jarg1, LinearTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void LinearTransform\_set\_is\_orthonormal(long jarg1, LinearTransform jarg1\_);

public final static native void LinearTransform\_verbose\_set(long jarg1, LinearTransform jarg1\_, boolean jarg2);

public final static native boolean LinearTransform\_verbose\_get(long jarg1, LinearTransform jarg1\_);

public final static native void LinearTransform\_print\_if\_verbose(long jarg1, LinearTransform jarg1\_, String jarg2, long jarg3, DoubleVector jarg3\_, int jarg4, int jarg5);

public final static native void delete\_LinearTransform(long jarg1);

public final static native long new\_RandomRotationMatrix\_\_SWIG\_0(int jarg1, int jarg2);

public final static native void RandomRotationMatrix\_init(long jarg1, RandomRotationMatrix jarg1\_, int jarg2);

public final static native void RandomRotationMatrix\_train(long jarg1, RandomRotationMatrix jarg1\_, long jarg2, long jarg3);

public final static native long new\_RandomRotationMatrix\_\_SWIG\_1();

public final static native void delete\_RandomRotationMatrix(long jarg1);

public final static native void PCAMatrix\_eigen\_power\_set(long jarg1, PCAMatrix jarg1\_, float jarg2);

public final static native float PCAMatrix\_eigen\_power\_get(long jarg1, PCAMatrix jarg1\_);

public final static native void PCAMatrix\_epsilon\_set(long jarg1, PCAMatrix jarg1\_, float jarg2);

public final static native float PCAMatrix\_epsilon\_get(long jarg1, PCAMatrix jarg1\_);

public final static native void PCAMatrix\_random\_rotation\_set(long jarg1, PCAMatrix jarg1\_, boolean jarg2);

public final static native boolean PCAMatrix\_random\_rotation\_get(long jarg1, PCAMatrix jarg1\_);

public final static native void PCAMatrix\_max\_points\_per\_d\_set(long jarg1, PCAMatrix jarg1\_, long jarg2);

public final static native long PCAMatrix\_max\_points\_per\_d\_get(long jarg1, PCAMatrix jarg1\_);

public final static native void PCAMatrix\_balanced\_bins\_set(long jarg1, PCAMatrix jarg1\_, int jarg2);

public final static native int PCAMatrix\_balanced\_bins\_get(long jarg1, PCAMatrix jarg1\_);

public final static native void PCAMatrix\_mean\_set(long jarg1, PCAMatrix jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long PCAMatrix\_mean\_get(long jarg1, PCAMatrix jarg1\_);

public final static native void PCAMatrix\_eigenvalues\_set(long jarg1, PCAMatrix jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long PCAMatrix\_eigenvalues\_get(long jarg1, PCAMatrix jarg1\_);

public final static native void PCAMatrix\_PCAMat\_set(long jarg1, PCAMatrix jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long PCAMatrix\_PCAMat\_get(long jarg1, PCAMatrix jarg1\_);

public final static native long new\_PCAMatrix\_\_SWIG\_0(int jarg1, int jarg2, float jarg3, boolean jarg4);

public final static native long new\_PCAMatrix\_\_SWIG\_1(int jarg1, int jarg2, float jarg3);

public final static native long new\_PCAMatrix\_\_SWIG\_2(int jarg1, int jarg2);

public final static native long new\_PCAMatrix\_\_SWIG\_3(int jarg1);

public final static native long new\_PCAMatrix\_\_SWIG\_4();

public final static native void PCAMatrix\_train(long jarg1, PCAMatrix jarg1\_, long jarg2, long jarg3);

public final static native void PCAMatrix\_copy\_from(long jarg1, PCAMatrix jarg1\_, long jarg2, PCAMatrix jarg2\_);

public final static native void PCAMatrix\_prepare\_Ab(long jarg1, PCAMatrix jarg1\_);

public final static native void delete\_PCAMatrix(long jarg1);

public final static native void ITQMatrix\_max\_iter\_set(long jarg1, ITQMatrix jarg1\_, int jarg2);

public final static native int ITQMatrix\_max\_iter\_get(long jarg1, ITQMatrix jarg1\_);

public final static native void ITQMatrix\_seed\_set(long jarg1, ITQMatrix jarg1\_, int jarg2);

public final static native int ITQMatrix\_seed\_get(long jarg1, ITQMatrix jarg1\_);

public final static native void ITQMatrix\_init\_rotation\_set(long jarg1, ITQMatrix jarg1\_, long jarg2, DoubleVector jarg2\_);

public final static native long ITQMatrix\_init\_rotation\_get(long jarg1, ITQMatrix jarg1\_);

public final static native long new\_ITQMatrix\_\_SWIG\_0(int jarg1);

public final static native long new\_ITQMatrix\_\_SWIG\_1();

public final static native void ITQMatrix\_train(long jarg1, ITQMatrix jarg1\_, long jarg2, long jarg3);

public final static native void delete\_ITQMatrix(long jarg1);

public final static native void ITQTransform\_mean\_set(long jarg1, ITQTransform jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long ITQTransform\_mean\_get(long jarg1, ITQTransform jarg1\_);

public final static native void ITQTransform\_do\_pca\_set(long jarg1, ITQTransform jarg1\_, boolean jarg2);

public final static native boolean ITQTransform\_do\_pca\_get(long jarg1, ITQTransform jarg1\_);

public final static native void ITQTransform\_itq\_set(long jarg1, ITQTransform jarg1\_, long jarg2, ITQMatrix jarg2\_);

public final static native long ITQTransform\_itq\_get(long jarg1, ITQTransform jarg1\_);

public final static native void ITQTransform\_max\_train\_per\_dim\_set(long jarg1, ITQTransform jarg1\_, int jarg2);

public final static native int ITQTransform\_max\_train\_per\_dim\_get(long jarg1, ITQTransform jarg1\_);

public final static native void ITQTransform\_pca\_then\_itq\_set(long jarg1, ITQTransform jarg1\_, long jarg2, LinearTransform jarg2\_);

public final static native long ITQTransform\_pca\_then\_itq\_get(long jarg1, ITQTransform jarg1\_);

public final static native long new\_ITQTransform\_\_SWIG\_0(int jarg1, int jarg2, boolean jarg3);

public final static native long new\_ITQTransform\_\_SWIG\_1(int jarg1, int jarg2);

public final static native long new\_ITQTransform\_\_SWIG\_2(int jarg1);

public final static native long new\_ITQTransform\_\_SWIG\_3();

public final static native void ITQTransform\_train(long jarg1, ITQTransform jarg1\_, long jarg2, long jarg3);

public final static native void ITQTransform\_apply\_noalloc(long jarg1, ITQTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void delete\_ITQTransform(long jarg1);

public final static native void OPQMatrix\_M\_set(long jarg1, OPQMatrix jarg1\_, int jarg2);

public final static native int OPQMatrix\_M\_get(long jarg1, OPQMatrix jarg1\_);

public final static native void OPQMatrix\_niter\_set(long jarg1, OPQMatrix jarg1\_, int jarg2);

public final static native int OPQMatrix\_niter\_get(long jarg1, OPQMatrix jarg1\_);

public final static native void OPQMatrix\_niter\_pq\_set(long jarg1, OPQMatrix jarg1\_, int jarg2);

public final static native int OPQMatrix\_niter\_pq\_get(long jarg1, OPQMatrix jarg1\_);

public final static native void OPQMatrix\_niter\_pq\_0\_set(long jarg1, OPQMatrix jarg1\_, int jarg2);

public final static native int OPQMatrix\_niter\_pq\_0\_get(long jarg1, OPQMatrix jarg1\_);

public final static native void OPQMatrix\_max\_train\_points\_set(long jarg1, OPQMatrix jarg1\_, long jarg2);

public final static native long OPQMatrix\_max\_train\_points\_get(long jarg1, OPQMatrix jarg1\_);

public final static native void OPQMatrix\_verbose\_set(long jarg1, OPQMatrix jarg1\_, boolean jarg2);

public final static native boolean OPQMatrix\_verbose\_get(long jarg1, OPQMatrix jarg1\_);

public final static native void OPQMatrix\_pq\_set(long jarg1, OPQMatrix jarg1\_, long jarg2, ProductQuantizer jarg2\_);

public final static native long OPQMatrix\_pq\_get(long jarg1, OPQMatrix jarg1\_);

public final static native long new\_OPQMatrix\_\_SWIG\_0(int jarg1, int jarg2, int jarg3);

public final static native long new\_OPQMatrix\_\_SWIG\_1(int jarg1, int jarg2);

public final static native long new\_OPQMatrix\_\_SWIG\_2(int jarg1);

public final static native long new\_OPQMatrix\_\_SWIG\_3();

public final static native void OPQMatrix\_train(long jarg1, OPQMatrix jarg1\_, long jarg2, long jarg3);

public final static native void delete\_OPQMatrix(long jarg1);

public final static native void RemapDimensionsTransform\_map\_set(long jarg1, RemapDimensionsTransform jarg1\_, long jarg2, IntVector jarg2\_);

public final static native long RemapDimensionsTransform\_map\_get(long jarg1, RemapDimensionsTransform jarg1\_);

public final static native long new\_RemapDimensionsTransform\_\_SWIG\_0(int jarg1, int jarg2, long jarg3);

public final static native long new\_RemapDimensionsTransform\_\_SWIG\_1(int jarg1, int jarg2, boolean jarg3);

public final static native long new\_RemapDimensionsTransform\_\_SWIG\_2(int jarg1, int jarg2);

public final static native void RemapDimensionsTransform\_apply\_noalloc(long jarg1, RemapDimensionsTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void RemapDimensionsTransform\_reverse\_transform(long jarg1, RemapDimensionsTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long new\_RemapDimensionsTransform\_\_SWIG\_3();

public final static native void delete\_RemapDimensionsTransform(long jarg1);

public final static native void NormalizationTransform\_norm\_set(long jarg1, NormalizationTransform jarg1\_, float jarg2);

public final static native float NormalizationTransform\_norm\_get(long jarg1, NormalizationTransform jarg1\_);

public final static native long new\_NormalizationTransform\_\_SWIG\_0(int jarg1, float jarg2);

public final static native long new\_NormalizationTransform\_\_SWIG\_1(int jarg1);

public final static native long new\_NormalizationTransform\_\_SWIG\_2();

public final static native void NormalizationTransform\_apply\_noalloc(long jarg1, NormalizationTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void NormalizationTransform\_reverse\_transform(long jarg1, NormalizationTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void delete\_NormalizationTransform(long jarg1);

public final static native void CenteringTransform\_mean\_set(long jarg1, CenteringTransform jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long CenteringTransform\_mean\_get(long jarg1, CenteringTransform jarg1\_);

public final static native long new\_CenteringTransform\_\_SWIG\_0(int jarg1);

public final static native long new\_CenteringTransform\_\_SWIG\_1();

public final static native void CenteringTransform\_train(long jarg1, CenteringTransform jarg1\_, long jarg2, long jarg3);

public final static native void CenteringTransform\_apply\_noalloc(long jarg1, CenteringTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void CenteringTransform\_reverse\_transform(long jarg1, CenteringTransform jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void delete\_CenteringTransform(long jarg1);

public final static native void IndexFlatCodes\_code\_size\_set(long jarg1, IndexFlatCodes jarg1\_, long jarg2);

public final static native long IndexFlatCodes\_code\_size\_get(long jarg1, IndexFlatCodes jarg1\_);

public final static native void IndexFlatCodes\_codes\_set(long jarg1, IndexFlatCodes jarg1\_, long jarg2, ByteVector jarg2\_);

public final static native long IndexFlatCodes\_codes\_get(long jarg1, IndexFlatCodes jarg1\_);

public final static native void IndexFlatCodes\_add(long jarg1, IndexFlatCodes jarg1\_, long jarg2, long jarg3);

public final static native void IndexFlatCodes\_reset(long jarg1, IndexFlatCodes jarg1\_);

public final static native void IndexFlatCodes\_reconstruct\_n(long jarg1, IndexFlatCodes jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexFlatCodes\_reconstruct(long jarg1, IndexFlatCodes jarg1\_, long jarg2, long jarg3);

public final static native long IndexFlatCodes\_sa\_code\_size(long jarg1, IndexFlatCodes jarg1\_);

public final static native long IndexFlatCodes\_remove\_ids(long jarg1, IndexFlatCodes jarg1\_, long jarg2, IDSelector jarg2\_);

public final static native void delete\_IndexFlatCodes(long jarg1);

public final static native long new\_IndexFlat\_\_SWIG\_0(long jarg1, int jarg2);

public final static native long new\_IndexFlat\_\_SWIG\_1(long jarg1);

public final static native void IndexFlat\_search(long jarg1, IndexFlat jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexFlat\_range\_search(long jarg1, IndexFlat jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, RangeSearchResult jarg5\_);

public final static native void IndexFlat\_reconstruct(long jarg1, IndexFlat jarg1\_, long jarg2, long jarg3);

public final static native void IndexFlat\_compute\_distance\_subset(long jarg1, IndexFlat jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native long IndexFlat\_get\_xb\_\_SWIG\_0(long jarg1, IndexFlat jarg1\_);

public final static native long new\_IndexFlat\_\_SWIG\_2();

public final static native long IndexFlat\_get\_distance\_computer(long jarg1, IndexFlat jarg1\_);

public final static native void IndexFlat\_sa\_encode(long jarg1, IndexFlat jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexFlat\_sa\_decode(long jarg1, IndexFlat jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void delete\_IndexFlat(long jarg1);

public final static native long new\_IndexFlatIP\_\_SWIG\_0(long jarg1);

public final static native long new\_IndexFlatIP\_\_SWIG\_1();

public final static native void delete\_IndexFlatIP(long jarg1);

public final static native long new\_IndexFlatL2\_\_SWIG\_0(long jarg1);

public final static native long new\_IndexFlatL2\_\_SWIG\_1();

public final static native void delete\_IndexFlatL2(long jarg1);

public final static native void IndexFlat1D\_continuous\_update\_set(long jarg1, IndexFlat1D jarg1\_, boolean jarg2);

public final static native boolean IndexFlat1D\_continuous\_update\_get(long jarg1, IndexFlat1D jarg1\_);

public final static native void IndexFlat1D\_perm\_set(long jarg1, IndexFlat1D jarg1\_, long jarg2);

public final static native long IndexFlat1D\_perm\_get(long jarg1, IndexFlat1D jarg1\_);

public final static native long new\_IndexFlat1D\_\_SWIG\_0(boolean jarg1);

public final static native long new\_IndexFlat1D\_\_SWIG\_1();

public final static native void IndexFlat1D\_update\_permutation(long jarg1, IndexFlat1D jarg1\_);

public final static native void IndexFlat1D\_add(long jarg1, IndexFlat1D jarg1\_, long jarg2, long jarg3);

public final static native void IndexFlat1D\_reset(long jarg1, IndexFlat1D jarg1\_);

public final static native void IndexFlat1D\_search(long jarg1, IndexFlat1D jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void delete\_IndexFlat1D(long jarg1);

public final static native void IndexLSH\_nbits\_set(long jarg1, IndexLSH jarg1\_, int jarg2);

public final static native int IndexLSH\_nbits\_get(long jarg1, IndexLSH jarg1\_);

public final static native void IndexLSH\_rotate\_data\_set(long jarg1, IndexLSH jarg1\_, boolean jarg2);

public final static native boolean IndexLSH\_rotate\_data\_get(long jarg1, IndexLSH jarg1\_);

public final static native void IndexLSH\_train\_thresholds\_set(long jarg1, IndexLSH jarg1\_, boolean jarg2);

public final static native boolean IndexLSH\_train\_thresholds\_get(long jarg1, IndexLSH jarg1\_);

public final static native void IndexLSH\_rrot\_set(long jarg1, IndexLSH jarg1\_, long jarg2, RandomRotationMatrix jarg2\_);

public final static native long IndexLSH\_rrot\_get(long jarg1, IndexLSH jarg1\_);

public final static native void IndexLSH\_thresholds\_set(long jarg1, IndexLSH jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long IndexLSH\_thresholds\_get(long jarg1, IndexLSH jarg1\_);

public final static native long new\_IndexLSH\_\_SWIG\_0(long jarg1, int jarg2, boolean jarg3, boolean jarg4);

public final static native long new\_IndexLSH\_\_SWIG\_1(long jarg1, int jarg2, boolean jarg3);

public final static native long new\_IndexLSH\_\_SWIG\_2(long jarg1, int jarg2);

public final static native long IndexLSH\_apply\_preprocess(long jarg1, IndexLSH jarg1\_, long jarg2, long jarg3);

public final static native void IndexLSH\_train(long jarg1, IndexLSH jarg1\_, long jarg2, long jarg3);

public final static native void IndexLSH\_search(long jarg1, IndexLSH jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexLSH\_transfer\_thresholds(long jarg1, IndexLSH jarg1\_, long jarg2, LinearTransform jarg2\_);

public final static native void delete\_IndexLSH(long jarg1);

public final static native long new\_IndexLSH\_\_SWIG\_3();

public final static native void IndexLSH\_sa\_encode(long jarg1, IndexLSH jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexLSH\_sa\_decode(long jarg1, IndexLSH jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void SimulatedAnnealingParameters\_init\_temperature\_set(long jarg1, SimulatedAnnealingParameters jarg1\_, double jarg2);

public final static native double SimulatedAnnealingParameters\_init\_temperature\_get(long jarg1, SimulatedAnnealingParameters jarg1\_);

public final static native void SimulatedAnnealingParameters\_temperature\_decay\_set(long jarg1, SimulatedAnnealingParameters jarg1\_, double jarg2);

public final static native double SimulatedAnnealingParameters\_temperature\_decay\_get(long jarg1, SimulatedAnnealingParameters jarg1\_);

public final static native void SimulatedAnnealingParameters\_n\_iter\_set(long jarg1, SimulatedAnnealingParameters jarg1\_, int jarg2);

public final static native int SimulatedAnnealingParameters\_n\_iter\_get(long jarg1, SimulatedAnnealingParameters jarg1\_);

public final static native void SimulatedAnnealingParameters\_n\_redo\_set(long jarg1, SimulatedAnnealingParameters jarg1\_, int jarg2);

public final static native int SimulatedAnnealingParameters\_n\_redo\_get(long jarg1, SimulatedAnnealingParameters jarg1\_);

public final static native void SimulatedAnnealingParameters\_seed\_set(long jarg1, SimulatedAnnealingParameters jarg1\_, int jarg2);

public final static native int SimulatedAnnealingParameters\_seed\_get(long jarg1, SimulatedAnnealingParameters jarg1\_);

public final static native void SimulatedAnnealingParameters\_verbose\_set(long jarg1, SimulatedAnnealingParameters jarg1\_, int jarg2);

public final static native int SimulatedAnnealingParameters\_verbose\_get(long jarg1, SimulatedAnnealingParameters jarg1\_);

public final static native void SimulatedAnnealingParameters\_only\_bit\_flips\_set(long jarg1, SimulatedAnnealingParameters jarg1\_, boolean jarg2);

public final static native boolean SimulatedAnnealingParameters\_only\_bit\_flips\_get(long jarg1, SimulatedAnnealingParameters jarg1\_);

public final static native void SimulatedAnnealingParameters\_init\_random\_set(long jarg1, SimulatedAnnealingParameters jarg1\_, boolean jarg2);

public final static native boolean SimulatedAnnealingParameters\_init\_random\_get(long jarg1, SimulatedAnnealingParameters jarg1\_);

public final static native long new\_SimulatedAnnealingParameters();

public final static native void delete\_SimulatedAnnealingParameters(long jarg1);

public final static native void PermutationObjective\_n\_set(long jarg1, PermutationObjective jarg1\_, int jarg2);

public final static native int PermutationObjective\_n\_get(long jarg1, PermutationObjective jarg1\_);

public final static native double PermutationObjective\_compute\_cost(long jarg1, PermutationObjective jarg1\_, long jarg2);

public final static native double PermutationObjective\_cost\_update(long jarg1, PermutationObjective jarg1\_, long jarg2, int jarg3, int jarg4);

public final static native void delete\_PermutationObjective(long jarg1);

public final static native void ReproduceDistancesObjective\_dis\_weight\_factor\_set(long jarg1, ReproduceDistancesObjective jarg1\_, double jarg2);

public final static native double ReproduceDistancesObjective\_dis\_weight\_factor\_get(long jarg1, ReproduceDistancesObjective jarg1\_);

public final static native double ReproduceDistancesObjective\_sqr(double jarg1);

public final static native double ReproduceDistancesObjective\_dis\_weight(long jarg1, ReproduceDistancesObjective jarg1\_, double jarg2);

public final static native void ReproduceDistancesObjective\_source\_dis\_set(long jarg1, ReproduceDistancesObjective jarg1\_, long jarg2, DoubleVector jarg2\_);

public final static native long ReproduceDistancesObjective\_source\_dis\_get(long jarg1, ReproduceDistancesObjective jarg1\_);

public final static native void ReproduceDistancesObjective\_target\_dis\_set(long jarg1, ReproduceDistancesObjective jarg1\_, long jarg2);

public final static native long ReproduceDistancesObjective\_target\_dis\_get(long jarg1, ReproduceDistancesObjective jarg1\_);

public final static native void ReproduceDistancesObjective\_weights\_set(long jarg1, ReproduceDistancesObjective jarg1\_, long jarg2, DoubleVector jarg2\_);

public final static native long ReproduceDistancesObjective\_weights\_get(long jarg1, ReproduceDistancesObjective jarg1\_);

public final static native double ReproduceDistancesObjective\_get\_source\_dis(long jarg1, ReproduceDistancesObjective jarg1\_, int jarg2, int jarg3);

public final static native double ReproduceDistancesObjective\_compute\_cost(long jarg1, ReproduceDistancesObjective jarg1\_, long jarg2);

public final static native double ReproduceDistancesObjective\_cost\_update(long jarg1, ReproduceDistancesObjective jarg1\_, long jarg2, int jarg3, int jarg4);

public final static native long new\_ReproduceDistancesObjective(int jarg1, long jarg2, long jarg3, double jarg4);

public final static native void ReproduceDistancesObjective\_compute\_mean\_stdev(long jarg1, long jarg2, long jarg3, long jarg4);

public final static native void ReproduceDistancesObjective\_set\_affine\_target\_dis(long jarg1, ReproduceDistancesObjective jarg1\_, long jarg2);

public final static native void delete\_ReproduceDistancesObjective(long jarg1);

public final static native void SimulatedAnnealingOptimizer\_obj\_set(long jarg1, SimulatedAnnealingOptimizer jarg1\_, long jarg2, PermutationObjective jarg2\_);

public final static native long SimulatedAnnealingOptimizer\_obj\_get(long jarg1, SimulatedAnnealingOptimizer jarg1\_);

public final static native void SimulatedAnnealingOptimizer\_n\_set(long jarg1, SimulatedAnnealingOptimizer jarg1\_, int jarg2);

public final static native int SimulatedAnnealingOptimizer\_n\_get(long jarg1, SimulatedAnnealingOptimizer jarg1\_);

public final static native void SimulatedAnnealingOptimizer\_logfile\_set(long jarg1, SimulatedAnnealingOptimizer jarg1\_, long jarg2);

public final static native long SimulatedAnnealingOptimizer\_logfile\_get(long jarg1, SimulatedAnnealingOptimizer jarg1\_);

public final static native long new\_SimulatedAnnealingOptimizer(long jarg1, PermutationObjective jarg1\_, long jarg2, SimulatedAnnealingParameters jarg2\_);

public final static native void SimulatedAnnealingOptimizer\_rnd\_set(long jarg1, SimulatedAnnealingOptimizer jarg1\_, long jarg2);

public final static native long SimulatedAnnealingOptimizer\_rnd\_get(long jarg1, SimulatedAnnealingOptimizer jarg1\_);

public final static native void SimulatedAnnealingOptimizer\_init\_cost\_set(long jarg1, SimulatedAnnealingOptimizer jarg1\_, double jarg2);

public final static native double SimulatedAnnealingOptimizer\_init\_cost\_get(long jarg1, SimulatedAnnealingOptimizer jarg1\_);

public final static native double SimulatedAnnealingOptimizer\_optimize(long jarg1, SimulatedAnnealingOptimizer jarg1\_, long jarg2);

public final static native double SimulatedAnnealingOptimizer\_run\_optimization(long jarg1, SimulatedAnnealingOptimizer jarg1\_, long jarg2);

public final static native void delete\_SimulatedAnnealingOptimizer(long jarg1);

public final static native void PolysemousTraining\_optimization\_type\_set(long jarg1, PolysemousTraining jarg1\_, int jarg2);

public final static native int PolysemousTraining\_optimization\_type\_get(long jarg1, PolysemousTraining jarg1\_);

public final static native void PolysemousTraining\_ntrain\_permutation\_set(long jarg1, PolysemousTraining jarg1\_, int jarg2);

public final static native int PolysemousTraining\_ntrain\_permutation\_get(long jarg1, PolysemousTraining jarg1\_);

public final static native void PolysemousTraining\_dis\_weight\_factor\_set(long jarg1, PolysemousTraining jarg1\_, double jarg2);

public final static native double PolysemousTraining\_dis\_weight\_factor\_get(long jarg1, PolysemousTraining jarg1\_);

public final static native void PolysemousTraining\_max\_memory\_set(long jarg1, PolysemousTraining jarg1\_, long jarg2);

public final static native long PolysemousTraining\_max\_memory\_get(long jarg1, PolysemousTraining jarg1\_);

public final static native void PolysemousTraining\_log\_pattern\_set(long jarg1, PolysemousTraining jarg1\_, String jarg2);

public final static native String PolysemousTraining\_log\_pattern\_get(long jarg1, PolysemousTraining jarg1\_);

public final static native long new\_PolysemousTraining();

public final static native void PolysemousTraining\_optimize\_pq\_for\_hamming(long jarg1, PolysemousTraining jarg1\_, long jarg2, ProductQuantizer jarg2\_, long jarg3, long jarg4);

public final static native void PolysemousTraining\_optimize\_ranking(long jarg1, PolysemousTraining jarg1\_, long jarg2, ProductQuantizer jarg2\_, long jarg3, long jarg4);

public final static native void PolysemousTraining\_optimize\_reproduce\_distances(long jarg1, PolysemousTraining jarg1\_, long jarg2, ProductQuantizer jarg2\_);

public final static native long PolysemousTraining\_memory\_usage\_per\_thread(long jarg1, PolysemousTraining jarg1\_, long jarg2, ProductQuantizer jarg2\_);

public final static native void delete\_PolysemousTraining(long jarg1);

public final static native void IndexPQ\_pq\_set(long jarg1, IndexPQ jarg1\_, long jarg2, ProductQuantizer jarg2\_);

public final static native long IndexPQ\_pq\_get(long jarg1, IndexPQ jarg1\_);

public final static native long new\_IndexPQ\_\_SWIG\_0(int jarg1, long jarg2, long jarg3, int jarg4);

public final static native long new\_IndexPQ\_\_SWIG\_1(int jarg1, long jarg2, long jarg3);

public final static native long new\_IndexPQ\_\_SWIG\_2();

public final static native void IndexPQ\_train(long jarg1, IndexPQ jarg1\_, long jarg2, long jarg3);

public final static native void IndexPQ\_search(long jarg1, IndexPQ jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexPQ\_sa\_encode(long jarg1, IndexPQ jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexPQ\_sa\_decode(long jarg1, IndexPQ jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long IndexPQ\_get\_distance\_computer(long jarg1, IndexPQ jarg1\_);

public final static native void IndexPQ\_do\_polysemous\_training\_set(long jarg1, IndexPQ jarg1\_, boolean jarg2);

public final static native boolean IndexPQ\_do\_polysemous\_training\_get(long jarg1, IndexPQ jarg1\_);

public final static native void IndexPQ\_polysemous\_training\_set(long jarg1, IndexPQ jarg1\_, long jarg2, PolysemousTraining jarg2\_);

public final static native long IndexPQ\_polysemous\_training\_get(long jarg1, IndexPQ jarg1\_);

public final static native void IndexPQ\_search\_type\_set(long jarg1, IndexPQ jarg1\_, int jarg2);

public final static native int IndexPQ\_search\_type\_get(long jarg1, IndexPQ jarg1\_);

public final static native void IndexPQ\_encode\_signs\_set(long jarg1, IndexPQ jarg1\_, boolean jarg2);

public final static native boolean IndexPQ\_encode\_signs\_get(long jarg1, IndexPQ jarg1\_);

public final static native void IndexPQ\_polysemous\_ht\_set(long jarg1, IndexPQ jarg1\_, int jarg2);

public final static native int IndexPQ\_polysemous\_ht\_get(long jarg1, IndexPQ jarg1\_);

public final static native void IndexPQ\_search\_core\_polysemous(long jarg1, IndexPQ jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexPQ\_hamming\_distance\_histogram(long jarg1, IndexPQ jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexPQ\_hamming\_distance\_table(long jarg1, IndexPQ jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void delete\_IndexPQ(long jarg1);

public final static native void IndexPQStats\_nq\_set(long jarg1, IndexPQStats jarg1\_, long jarg2);

public final static native long IndexPQStats\_nq\_get(long jarg1, IndexPQStats jarg1\_);

public final static native void IndexPQStats\_ncode\_set(long jarg1, IndexPQStats jarg1\_, long jarg2);

public final static native long IndexPQStats\_ncode\_get(long jarg1, IndexPQStats jarg1\_);

public final static native void IndexPQStats\_n\_hamming\_pass\_set(long jarg1, IndexPQStats jarg1\_, long jarg2);

public final static native long IndexPQStats\_n\_hamming\_pass\_get(long jarg1, IndexPQStats jarg1\_);

public final static native long new\_IndexPQStats();

public final static native void IndexPQStats\_reset(long jarg1, IndexPQStats jarg1\_);

public final static native void delete\_IndexPQStats(long jarg1);

public final static native void indexPQ\_stats\_set(long jarg1, IndexPQStats jarg1\_);

public final static native long indexPQ\_stats\_get();

public final static native void MultiIndexQuantizer\_pq\_set(long jarg1, MultiIndexQuantizer jarg1\_, long jarg2, ProductQuantizer jarg2\_);

public final static native long MultiIndexQuantizer\_pq\_get(long jarg1, MultiIndexQuantizer jarg1\_);

public final static native long new\_MultiIndexQuantizer\_\_SWIG\_0(int jarg1, long jarg2, long jarg3);

public final static native void MultiIndexQuantizer\_train(long jarg1, MultiIndexQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void MultiIndexQuantizer\_search(long jarg1, MultiIndexQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void MultiIndexQuantizer\_add(long jarg1, MultiIndexQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void MultiIndexQuantizer\_reset(long jarg1, MultiIndexQuantizer jarg1\_);

public final static native long new\_MultiIndexQuantizer\_\_SWIG\_1();

public final static native void MultiIndexQuantizer\_reconstruct(long jarg1, MultiIndexQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void delete\_MultiIndexQuantizer(long jarg1);

public final static native void MultiIndexQuantizer2\_assign\_indexes\_set(long jarg1, MultiIndexQuantizer2 jarg1\_, long jarg2);

public final static native long MultiIndexQuantizer2\_assign\_indexes\_get(long jarg1, MultiIndexQuantizer2 jarg1\_);

public final static native void MultiIndexQuantizer2\_own\_fields\_set(long jarg1, MultiIndexQuantizer2 jarg1\_, boolean jarg2);

public final static native boolean MultiIndexQuantizer2\_own\_fields\_get(long jarg1, MultiIndexQuantizer2 jarg1\_);

public final static native long new\_MultiIndexQuantizer2\_\_SWIG\_0(int jarg1, long jarg2, long jarg3, long jarg4);

public final static native long new\_MultiIndexQuantizer2\_\_SWIG\_1(int jarg1, long jarg2, long jarg3, Index jarg3\_, long jarg4, Index jarg4\_);

public final static native void MultiIndexQuantizer2\_train(long jarg1, MultiIndexQuantizer2 jarg1\_, long jarg2, long jarg3);

public final static native void MultiIndexQuantizer2\_search(long jarg1, MultiIndexQuantizer2 jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void delete\_MultiIndexQuantizer2(long jarg1);

public final static native void InvertedLists\_nlist\_set(long jarg1, InvertedLists jarg1\_, long jarg2);

public final static native long InvertedLists\_nlist\_get(long jarg1, InvertedLists jarg1\_);

public final static native void InvertedLists\_code\_size\_set(long jarg1, InvertedLists jarg1\_, long jarg2);

public final static native long InvertedLists\_code\_size\_get(long jarg1, InvertedLists jarg1\_);

public final static native long InvertedLists\_INVALID\_CODE\_SIZE\_get();

public final static native long InvertedLists\_list\_size(long jarg1, InvertedLists jarg1\_, long jarg2);

public final static native long InvertedLists\_get\_codes(long jarg1, InvertedLists jarg1\_, long jarg2);

public final static native long InvertedLists\_get\_ids(long jarg1, InvertedLists jarg1\_, long jarg2);

public final static native void InvertedLists\_release\_codes(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void InvertedLists\_release\_ids(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long InvertedLists\_get\_single\_id(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long InvertedLists\_get\_single\_code(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void InvertedLists\_prefetch\_lists(long jarg1, InvertedLists jarg1\_, long jarg2, LongVector jarg2\_, int jarg3);

public final static native long InvertedLists\_add\_entry(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long InvertedLists\_add\_entries(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void InvertedLists\_update\_entry(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native void InvertedLists\_update\_entries(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6);

public final static native void InvertedLists\_resize(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void InvertedLists\_reset(long jarg1, InvertedLists jarg1\_);

public final static native void InvertedLists\_merge\_from(long jarg1, InvertedLists jarg1\_, long jarg2, InvertedLists jarg2\_, long jarg3);

public final static native void delete\_InvertedLists(long jarg1);

public final static native double InvertedLists\_imbalance\_factor(long jarg1, InvertedLists jarg1\_);

public final static native void InvertedLists\_print\_stats(long jarg1, InvertedLists jarg1\_);

public final static native long InvertedLists\_compute\_ntotal(long jarg1, InvertedLists jarg1\_);

public final static native void InvertedLists\_ScopedIds\_il\_set(long jarg1, InvertedLists.ScopedIds jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long InvertedLists\_ScopedIds\_il\_get(long jarg1, InvertedLists.ScopedIds jarg1\_);

public final static native void InvertedLists\_ScopedIds\_ids\_set(long jarg1, InvertedLists.ScopedIds jarg1\_, long jarg2, LongVector jarg2\_);

public final static native long InvertedLists\_ScopedIds\_ids\_get(long jarg1, InvertedLists.ScopedIds jarg1\_);

public final static native void InvertedLists\_ScopedIds\_list\_no\_set(long jarg1, InvertedLists.ScopedIds jarg1\_, long jarg2);

public final static native long InvertedLists\_ScopedIds\_list\_no\_get(long jarg1, InvertedLists.ScopedIds jarg1\_);

public final static native long new\_InvertedLists\_ScopedIds(long jarg1, InvertedLists jarg1\_, long jarg2);

public final static native long InvertedLists\_ScopedIds\_get(long jarg1, InvertedLists.ScopedIds jarg1\_);

public final static native void delete\_InvertedLists\_ScopedIds(long jarg1);

public final static native void InvertedLists\_ScopedCodes\_il\_set(long jarg1, InvertedLists.ScopedCodes jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long InvertedLists\_ScopedCodes\_il\_get(long jarg1, InvertedLists.ScopedCodes jarg1\_);

public final static native void InvertedLists\_ScopedCodes\_codes\_set(long jarg1, InvertedLists.ScopedCodes jarg1\_, long jarg2);

public final static native long InvertedLists\_ScopedCodes\_codes\_get(long jarg1, InvertedLists.ScopedCodes jarg1\_);

public final static native void InvertedLists\_ScopedCodes\_list\_no\_set(long jarg1, InvertedLists.ScopedCodes jarg1\_, long jarg2);

public final static native long InvertedLists\_ScopedCodes\_list\_no\_get(long jarg1, InvertedLists.ScopedCodes jarg1\_);

public final static native long new\_InvertedLists\_ScopedCodes\_\_SWIG\_0(long jarg1, InvertedLists jarg1\_, long jarg2);

public final static native long new\_InvertedLists\_ScopedCodes\_\_SWIG\_1(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long InvertedLists\_ScopedCodes\_get(long jarg1, InvertedLists.ScopedCodes jarg1\_);

public final static native void delete\_InvertedLists\_ScopedCodes(long jarg1);

public final static native void ArrayInvertedLists\_codes\_set(long jarg1, ArrayInvertedLists jarg1\_, long jarg2, ByteVectorVector jarg2\_);

public final static native long ArrayInvertedLists\_codes\_get(long jarg1, ArrayInvertedLists jarg1\_);

public final static native void ArrayInvertedLists\_ids\_set(long jarg1, ArrayInvertedLists jarg1\_, long jarg2);

public final static native long ArrayInvertedLists\_ids\_get(long jarg1, ArrayInvertedLists jarg1\_);

public final static native long new\_ArrayInvertedLists(long jarg1, long jarg2);

public final static native long ArrayInvertedLists\_list\_size(long jarg1, ArrayInvertedLists jarg1\_, long jarg2);

public final static native long ArrayInvertedLists\_get\_codes(long jarg1, ArrayInvertedLists jarg1\_, long jarg2);

public final static native long ArrayInvertedLists\_get\_ids(long jarg1, ArrayInvertedLists jarg1\_, long jarg2);

public final static native long ArrayInvertedLists\_add\_entries(long jarg1, ArrayInvertedLists jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void ArrayInvertedLists\_update\_entries(long jarg1, ArrayInvertedLists jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6);

public final static native void ArrayInvertedLists\_resize(long jarg1, ArrayInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void delete\_ArrayInvertedLists(long jarg1);

public final static native long ReadOnlyInvertedLists\_add\_entries(long jarg1, ReadOnlyInvertedLists jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void ReadOnlyInvertedLists\_update\_entries(long jarg1, ReadOnlyInvertedLists jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6);

public final static native void ReadOnlyInvertedLists\_resize(long jarg1, ReadOnlyInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void delete\_ReadOnlyInvertedLists(long jarg1);

public final static native void HStackInvertedLists\_ils\_set(long jarg1, HStackInvertedLists jarg1\_, long jarg2);

public final static native long HStackInvertedLists\_ils\_get(long jarg1, HStackInvertedLists jarg1\_);

public final static native long new\_HStackInvertedLists(int jarg1, long jarg2);

public final static native long HStackInvertedLists\_list\_size(long jarg1, HStackInvertedLists jarg1\_, long jarg2);

public final static native long HStackInvertedLists\_get\_codes(long jarg1, HStackInvertedLists jarg1\_, long jarg2);

public final static native long HStackInvertedLists\_get\_ids(long jarg1, HStackInvertedLists jarg1\_, long jarg2);

public final static native void HStackInvertedLists\_prefetch\_lists(long jarg1, HStackInvertedLists jarg1\_, long jarg2, LongVector jarg2\_, int jarg3);

public final static native void HStackInvertedLists\_release\_codes(long jarg1, HStackInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void HStackInvertedLists\_release\_ids(long jarg1, HStackInvertedLists jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long HStackInvertedLists\_get\_single\_id(long jarg1, HStackInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long HStackInvertedLists\_get\_single\_code(long jarg1, HStackInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void delete\_HStackInvertedLists(long jarg1);

public final static native void SliceInvertedLists\_il\_set(long jarg1, SliceInvertedLists jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long SliceInvertedLists\_il\_get(long jarg1, SliceInvertedLists jarg1\_);

public final static native void SliceInvertedLists\_i0\_set(long jarg1, SliceInvertedLists jarg1\_, long jarg2);

public final static native long SliceInvertedLists\_i0\_get(long jarg1, SliceInvertedLists jarg1\_);

public final static native void SliceInvertedLists\_i1\_set(long jarg1, SliceInvertedLists jarg1\_, long jarg2);

public final static native long SliceInvertedLists\_i1\_get(long jarg1, SliceInvertedLists jarg1\_);

public final static native long new\_SliceInvertedLists(long jarg1, InvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long SliceInvertedLists\_list\_size(long jarg1, SliceInvertedLists jarg1\_, long jarg2);

public final static native long SliceInvertedLists\_get\_codes(long jarg1, SliceInvertedLists jarg1\_, long jarg2);

public final static native long SliceInvertedLists\_get\_ids(long jarg1, SliceInvertedLists jarg1\_, long jarg2);

public final static native void SliceInvertedLists\_release\_codes(long jarg1, SliceInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void SliceInvertedLists\_release\_ids(long jarg1, SliceInvertedLists jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long SliceInvertedLists\_get\_single\_id(long jarg1, SliceInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long SliceInvertedLists\_get\_single\_code(long jarg1, SliceInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void SliceInvertedLists\_prefetch\_lists(long jarg1, SliceInvertedLists jarg1\_, long jarg2, LongVector jarg2\_, int jarg3);

public final static native void delete\_SliceInvertedLists(long jarg1);

public final static native void VStackInvertedLists\_ils\_set(long jarg1, VStackInvertedLists jarg1\_, long jarg2);

public final static native long VStackInvertedLists\_ils\_get(long jarg1, VStackInvertedLists jarg1\_);

public final static native void VStackInvertedLists\_cumsz\_set(long jarg1, VStackInvertedLists jarg1\_, long jarg2);

public final static native long VStackInvertedLists\_cumsz\_get(long jarg1, VStackInvertedLists jarg1\_);

public final static native long new\_VStackInvertedLists(int jarg1, long jarg2);

public final static native long VStackInvertedLists\_list\_size(long jarg1, VStackInvertedLists jarg1\_, long jarg2);

public final static native long VStackInvertedLists\_get\_codes(long jarg1, VStackInvertedLists jarg1\_, long jarg2);

public final static native long VStackInvertedLists\_get\_ids(long jarg1, VStackInvertedLists jarg1\_, long jarg2);

public final static native void VStackInvertedLists\_release\_codes(long jarg1, VStackInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void VStackInvertedLists\_release\_ids(long jarg1, VStackInvertedLists jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long VStackInvertedLists\_get\_single\_id(long jarg1, VStackInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long VStackInvertedLists\_get\_single\_code(long jarg1, VStackInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void VStackInvertedLists\_prefetch\_lists(long jarg1, VStackInvertedLists jarg1\_, long jarg2, LongVector jarg2\_, int jarg3);

public final static native void delete\_VStackInvertedLists(long jarg1);

public final static native void MaskedInvertedLists\_il0\_set(long jarg1, MaskedInvertedLists jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long MaskedInvertedLists\_il0\_get(long jarg1, MaskedInvertedLists jarg1\_);

public final static native void MaskedInvertedLists\_il1\_set(long jarg1, MaskedInvertedLists jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long MaskedInvertedLists\_il1\_get(long jarg1, MaskedInvertedLists jarg1\_);

public final static native long new\_MaskedInvertedLists(long jarg1, InvertedLists jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long MaskedInvertedLists\_list\_size(long jarg1, MaskedInvertedLists jarg1\_, long jarg2);

public final static native long MaskedInvertedLists\_get\_codes(long jarg1, MaskedInvertedLists jarg1\_, long jarg2);

public final static native long MaskedInvertedLists\_get\_ids(long jarg1, MaskedInvertedLists jarg1\_, long jarg2);

public final static native void MaskedInvertedLists\_release\_codes(long jarg1, MaskedInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void MaskedInvertedLists\_release\_ids(long jarg1, MaskedInvertedLists jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long MaskedInvertedLists\_get\_single\_id(long jarg1, MaskedInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long MaskedInvertedLists\_get\_single\_code(long jarg1, MaskedInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void MaskedInvertedLists\_prefetch\_lists(long jarg1, MaskedInvertedLists jarg1\_, long jarg2, LongVector jarg2\_, int jarg3);

public final static native void delete\_MaskedInvertedLists(long jarg1);

public final static native void StopWordsInvertedLists\_il0\_set(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long StopWordsInvertedLists\_il0\_get(long jarg1, StopWordsInvertedLists jarg1\_);

public final static native void StopWordsInvertedLists\_maxsize\_set(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2);

public final static native long StopWordsInvertedLists\_maxsize\_get(long jarg1, StopWordsInvertedLists jarg1\_);

public final static native long new\_StopWordsInvertedLists(long jarg1, InvertedLists jarg1\_, long jarg2);

public final static native long StopWordsInvertedLists\_list\_size(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2);

public final static native long StopWordsInvertedLists\_get\_codes(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2);

public final static native long StopWordsInvertedLists\_get\_ids(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2);

public final static native void StopWordsInvertedLists\_release\_codes(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void StopWordsInvertedLists\_release\_ids(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long StopWordsInvertedLists\_get\_single\_id(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long StopWordsInvertedLists\_get\_single\_code(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void StopWordsInvertedLists\_prefetch\_lists(long jarg1, StopWordsInvertedLists jarg1\_, long jarg2, LongVector jarg2\_, int jarg3);

public final static native void delete\_StopWordsInvertedLists(long jarg1);

public final static native void Level1Quantizer\_quantizer\_set(long jarg1, Level1Quantizer jarg1\_, long jarg2, Index jarg2\_);

public final static native long Level1Quantizer\_quantizer\_get(long jarg1, Level1Quantizer jarg1\_);

public final static native void Level1Quantizer\_nlist\_set(long jarg1, Level1Quantizer jarg1\_, long jarg2);

public final static native long Level1Quantizer\_nlist\_get(long jarg1, Level1Quantizer jarg1\_);

public final static native void Level1Quantizer\_quantizer\_trains\_alone\_set(long jarg1, Level1Quantizer jarg1\_, char jarg2);

public final static native char Level1Quantizer\_quantizer\_trains\_alone\_get(long jarg1, Level1Quantizer jarg1\_);

public final static native void Level1Quantizer\_own\_fields\_set(long jarg1, Level1Quantizer jarg1\_, boolean jarg2);

public final static native boolean Level1Quantizer\_own\_fields\_get(long jarg1, Level1Quantizer jarg1\_);

public final static native void Level1Quantizer\_cp\_set(long jarg1, Level1Quantizer jarg1\_, long jarg2, ClusteringParameters jarg2\_);

public final static native long Level1Quantizer\_cp\_get(long jarg1, Level1Quantizer jarg1\_);

public final static native void Level1Quantizer\_clustering\_index\_set(long jarg1, Level1Quantizer jarg1\_, long jarg2, Index jarg2\_);

public final static native long Level1Quantizer\_clustering\_index\_get(long jarg1, Level1Quantizer jarg1\_);

public final static native void Level1Quantizer\_train\_q1(long jarg1, Level1Quantizer jarg1\_, long jarg2, long jarg3, boolean jarg4, int jarg5);

public final static native long Level1Quantizer\_coarse\_code\_size(long jarg1, Level1Quantizer jarg1\_);

public final static native void Level1Quantizer\_encode\_listno(long jarg1, Level1Quantizer jarg1\_, long jarg2, long jarg3);

public final static native long Level1Quantizer\_decode\_listno(long jarg1, Level1Quantizer jarg1\_, long jarg2);

public final static native long new\_Level1Quantizer\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2);

public final static native long new\_Level1Quantizer\_\_SWIG\_1();

public final static native void delete\_Level1Quantizer(long jarg1);

public final static native void IVFSearchParameters\_nprobe\_set(long jarg1, IVFSearchParameters jarg1\_, long jarg2);

public final static native long IVFSearchParameters\_nprobe\_get(long jarg1, IVFSearchParameters jarg1\_);

public final static native void IVFSearchParameters\_max\_codes\_set(long jarg1, IVFSearchParameters jarg1\_, long jarg2);

public final static native long IVFSearchParameters\_max\_codes\_get(long jarg1, IVFSearchParameters jarg1\_);

public final static native long new\_IVFSearchParameters();

public final static native void delete\_IVFSearchParameters(long jarg1);

public final static native void IndexIVF\_invlists\_set(long jarg1, IndexIVF jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long IndexIVF\_invlists\_get(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_own\_invlists\_set(long jarg1, IndexIVF jarg1\_, boolean jarg2);

public final static native boolean IndexIVF\_own\_invlists\_get(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_code\_size\_set(long jarg1, IndexIVF jarg1\_, long jarg2);

public final static native long IndexIVF\_code\_size\_get(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_nprobe\_set(long jarg1, IndexIVF jarg1\_, long jarg2);

public final static native long IndexIVF\_nprobe\_get(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_max\_codes\_set(long jarg1, IndexIVF jarg1\_, long jarg2);

public final static native long IndexIVF\_max\_codes\_get(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_parallel\_mode\_set(long jarg1, IndexIVF jarg1\_, int jarg2);

public final static native int IndexIVF\_parallel\_mode\_get(long jarg1, IndexIVF jarg1\_);

public final static native int IndexIVF\_PARALLEL\_MODE\_NO\_HEAP\_INIT\_get(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_direct\_map\_set(long jarg1, IndexIVF jarg1\_, long jarg2);

public final static native long IndexIVF\_direct\_map\_get(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_reset(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_train(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3);

public final static native void IndexIVF\_add(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3);

public final static native void IndexIVF\_add\_with\_ids(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void IndexIVF\_add\_core(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, LongVector jarg5\_);

public final static native void IndexIVF\_encode\_vectors\_\_SWIG\_0(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, boolean jarg6);

public final static native void IndexIVF\_encode\_vectors\_\_SWIG\_1(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void IndexIVF\_add\_sa\_codes(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void IndexIVF\_train\_residual(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3);

public final static native void IndexIVF\_search\_preassigned\_\_SWIG\_0(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, boolean jarg9, long jarg10, IVFSearchParameters jarg10\_, long jarg11, IndexIVFStats jarg11\_);

public final static native void IndexIVF\_search\_preassigned\_\_SWIG\_1(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, boolean jarg9, long jarg10, IVFSearchParameters jarg10\_);

public final static native void IndexIVF\_search\_preassigned\_\_SWIG\_2(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, boolean jarg9);

public final static native void IndexIVF\_search(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexIVF\_range\_search(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, RangeSearchResult jarg5\_);

public final static native void IndexIVF\_range\_search\_preassigned\_\_SWIG\_0(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, RangeSearchResult jarg7\_, boolean jarg8, long jarg9, IVFSearchParameters jarg9\_, long jarg10, IndexIVFStats jarg10\_);

public final static native void IndexIVF\_range\_search\_preassigned\_\_SWIG\_1(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, RangeSearchResult jarg7\_, boolean jarg8, long jarg9, IVFSearchParameters jarg9\_);

public final static native void IndexIVF\_range\_search\_preassigned\_\_SWIG\_2(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, RangeSearchResult jarg7\_, boolean jarg8);

public final static native void IndexIVF\_range\_search\_preassigned\_\_SWIG\_3(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, RangeSearchResult jarg7\_);

public final static native long IndexIVF\_get\_InvertedListScanner\_\_SWIG\_0(long jarg1, IndexIVF jarg1\_, boolean jarg2);

public final static native long IndexIVF\_get\_InvertedListScanner\_\_SWIG\_1(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_reconstruct(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3);

public final static native void IndexIVF\_update\_vectors(long jarg1, IndexIVF jarg1\_, int jarg2, long jarg3, LongVector jarg3\_, long jarg4);

public final static native void IndexIVF\_reconstruct\_n(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexIVF\_search\_and\_reconstruct(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_, long jarg7);

public final static native void IndexIVF\_reconstruct\_from\_offset(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long IndexIVF\_remove\_ids(long jarg1, IndexIVF jarg1\_, long jarg2, IDSelector jarg2\_);

public final static native void IndexIVF\_check\_compatible\_for\_merge(long jarg1, IndexIVF jarg1\_, long jarg2, IndexIVF jarg2\_);

public final static native void IndexIVF\_merge\_from(long jarg1, IndexIVF jarg1\_, long jarg2, IndexIVF jarg2\_, long jarg3);

public final static native void IndexIVF\_copy\_subset\_to(long jarg1, IndexIVF jarg1\_, long jarg2, IndexIVF jarg2\_, int jarg3, long jarg4, long jarg5);

public final static native void delete\_IndexIVF(long jarg1);

public final static native long IndexIVF\_get\_list\_size(long jarg1, IndexIVF jarg1\_, long jarg2);

public final static native void IndexIVF\_make\_direct\_map\_\_SWIG\_0(long jarg1, IndexIVF jarg1\_, boolean jarg2);

public final static native void IndexIVF\_make\_direct\_map\_\_SWIG\_1(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_set\_direct\_map\_type(long jarg1, IndexIVF jarg1\_, long jarg2);

public final static native void IndexIVF\_replace\_invlists\_\_SWIG\_0(long jarg1, IndexIVF jarg1\_, long jarg2, InvertedLists jarg2\_, boolean jarg3);

public final static native void IndexIVF\_replace\_invlists\_\_SWIG\_1(long jarg1, IndexIVF jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long IndexIVF\_sa\_code\_size(long jarg1, IndexIVF jarg1\_);

public final static native void IndexIVF\_sa\_encode(long jarg1, IndexIVF jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexIVFStats\_nq\_set(long jarg1, IndexIVFStats jarg1\_, long jarg2);

public final static native long IndexIVFStats\_nq\_get(long jarg1, IndexIVFStats jarg1\_);

public final static native void IndexIVFStats\_nlist\_set(long jarg1, IndexIVFStats jarg1\_, long jarg2);

public final static native long IndexIVFStats\_nlist\_get(long jarg1, IndexIVFStats jarg1\_);

public final static native void IndexIVFStats\_ndis\_set(long jarg1, IndexIVFStats jarg1\_, long jarg2);

public final static native long IndexIVFStats\_ndis\_get(long jarg1, IndexIVFStats jarg1\_);

public final static native void IndexIVFStats\_nheap\_updates\_set(long jarg1, IndexIVFStats jarg1\_, long jarg2);

public final static native long IndexIVFStats\_nheap\_updates\_get(long jarg1, IndexIVFStats jarg1\_);

public final static native void IndexIVFStats\_quantization\_time\_set(long jarg1, IndexIVFStats jarg1\_, double jarg2);

public final static native double IndexIVFStats\_quantization\_time\_get(long jarg1, IndexIVFStats jarg1\_);

public final static native void IndexIVFStats\_search\_time\_set(long jarg1, IndexIVFStats jarg1\_, double jarg2);

public final static native double IndexIVFStats\_search\_time\_get(long jarg1, IndexIVFStats jarg1\_);

public final static native long new\_IndexIVFStats();

public final static native void IndexIVFStats\_reset(long jarg1, IndexIVFStats jarg1\_);

public final static native void IndexIVFStats\_add(long jarg1, IndexIVFStats jarg1\_, long jarg2, IndexIVFStats jarg2\_);

public final static native void delete\_IndexIVFStats(long jarg1);

public final static native void indexIVF\_stats\_set(long jarg1, IndexIVFStats jarg1\_);

public final static native long indexIVF\_stats\_get();

public final static native short[] hamdis\_tab\_ham\_bytes\_get();

public final static native void HammingComputer4\_a0\_set(long jarg1, HammingComputer4 jarg1\_, long jarg2);

public final static native long HammingComputer4\_a0\_get(long jarg1, HammingComputer4 jarg1\_);

public final static native long new\_HammingComputer4\_\_SWIG\_0();

public final static native long new\_HammingComputer4\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void HammingComputer4\_set(long jarg1, HammingComputer4 jarg1\_, long jarg2, int jarg3);

public final static native int HammingComputer4\_hamming(long jarg1, HammingComputer4 jarg1\_, long jarg2);

public final static native void delete\_HammingComputer4(long jarg1);

public final static native void HammingComputer8\_a0\_set(long jarg1, HammingComputer8 jarg1\_, long jarg2);

public final static native long HammingComputer8\_a0\_get(long jarg1, HammingComputer8 jarg1\_);

public final static native long new\_HammingComputer8\_\_SWIG\_0();

public final static native long new\_HammingComputer8\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void HammingComputer8\_set(long jarg1, HammingComputer8 jarg1\_, long jarg2, int jarg3);

public final static native int HammingComputer8\_hamming(long jarg1, HammingComputer8 jarg1\_, long jarg2);

public final static native void delete\_HammingComputer8(long jarg1);

public final static native void HammingComputer16\_a0\_set(long jarg1, HammingComputer16 jarg1\_, long jarg2);

public final static native long HammingComputer16\_a0\_get(long jarg1, HammingComputer16 jarg1\_);

public final static native void HammingComputer16\_a1\_set(long jarg1, HammingComputer16 jarg1\_, long jarg2);

public final static native long HammingComputer16\_a1\_get(long jarg1, HammingComputer16 jarg1\_);

public final static native long new\_HammingComputer16\_\_SWIG\_0();

public final static native long new\_HammingComputer16\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void HammingComputer16\_set(long jarg1, HammingComputer16 jarg1\_, long jarg2, int jarg3);

public final static native int HammingComputer16\_hamming(long jarg1, HammingComputer16 jarg1\_, long jarg2);

public final static native void delete\_HammingComputer16(long jarg1);

public final static native void HammingComputer20\_a0\_set(long jarg1, HammingComputer20 jarg1\_, long jarg2);

public final static native long HammingComputer20\_a0\_get(long jarg1, HammingComputer20 jarg1\_);

public final static native void HammingComputer20\_a1\_set(long jarg1, HammingComputer20 jarg1\_, long jarg2);

public final static native long HammingComputer20\_a1\_get(long jarg1, HammingComputer20 jarg1\_);

public final static native void HammingComputer20\_a2\_set(long jarg1, HammingComputer20 jarg1\_, long jarg2);

public final static native long HammingComputer20\_a2\_get(long jarg1, HammingComputer20 jarg1\_);

public final static native long new\_HammingComputer20\_\_SWIG\_0();

public final static native long new\_HammingComputer20\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void HammingComputer20\_set(long jarg1, HammingComputer20 jarg1\_, long jarg2, int jarg3);

public final static native int HammingComputer20\_hamming(long jarg1, HammingComputer20 jarg1\_, long jarg2);

public final static native void delete\_HammingComputer20(long jarg1);

public final static native void HammingComputer32\_a0\_set(long jarg1, HammingComputer32 jarg1\_, long jarg2);

public final static native long HammingComputer32\_a0\_get(long jarg1, HammingComputer32 jarg1\_);

public final static native void HammingComputer32\_a1\_set(long jarg1, HammingComputer32 jarg1\_, long jarg2);

public final static native long HammingComputer32\_a1\_get(long jarg1, HammingComputer32 jarg1\_);

public final static native void HammingComputer32\_a2\_set(long jarg1, HammingComputer32 jarg1\_, long jarg2);

public final static native long HammingComputer32\_a2\_get(long jarg1, HammingComputer32 jarg1\_);

public final static native void HammingComputer32\_a3\_set(long jarg1, HammingComputer32 jarg1\_, long jarg2);

public final static native long HammingComputer32\_a3\_get(long jarg1, HammingComputer32 jarg1\_);

public final static native long new\_HammingComputer32\_\_SWIG\_0();

public final static native long new\_HammingComputer32\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void HammingComputer32\_set(long jarg1, HammingComputer32 jarg1\_, long jarg2, int jarg3);

public final static native int HammingComputer32\_hamming(long jarg1, HammingComputer32 jarg1\_, long jarg2);

public final static native void delete\_HammingComputer32(long jarg1);

public final static native void HammingComputer64\_a0\_set(long jarg1, HammingComputer64 jarg1\_, long jarg2);

public final static native long HammingComputer64\_a0\_get(long jarg1, HammingComputer64 jarg1\_);

public final static native void HammingComputer64\_a1\_set(long jarg1, HammingComputer64 jarg1\_, long jarg2);

public final static native long HammingComputer64\_a1\_get(long jarg1, HammingComputer64 jarg1\_);

public final static native void HammingComputer64\_a2\_set(long jarg1, HammingComputer64 jarg1\_, long jarg2);

public final static native long HammingComputer64\_a2\_get(long jarg1, HammingComputer64 jarg1\_);

public final static native void HammingComputer64\_a3\_set(long jarg1, HammingComputer64 jarg1\_, long jarg2);

public final static native long HammingComputer64\_a3\_get(long jarg1, HammingComputer64 jarg1\_);

public final static native void HammingComputer64\_a4\_set(long jarg1, HammingComputer64 jarg1\_, long jarg2);

public final static native long HammingComputer64\_a4\_get(long jarg1, HammingComputer64 jarg1\_);

public final static native void HammingComputer64\_a5\_set(long jarg1, HammingComputer64 jarg1\_, long jarg2);

public final static native long HammingComputer64\_a5\_get(long jarg1, HammingComputer64 jarg1\_);

public final static native void HammingComputer64\_a6\_set(long jarg1, HammingComputer64 jarg1\_, long jarg2);

public final static native long HammingComputer64\_a6\_get(long jarg1, HammingComputer64 jarg1\_);

public final static native void HammingComputer64\_a7\_set(long jarg1, HammingComputer64 jarg1\_, long jarg2);

public final static native long HammingComputer64\_a7\_get(long jarg1, HammingComputer64 jarg1\_);

public final static native long new\_HammingComputer64\_\_SWIG\_0();

public final static native long new\_HammingComputer64\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void HammingComputer64\_set(long jarg1, HammingComputer64 jarg1\_, long jarg2, int jarg3);

public final static native int HammingComputer64\_hamming(long jarg1, HammingComputer64 jarg1\_, long jarg2);

public final static native void delete\_HammingComputer64(long jarg1);

public final static native void HammingComputerDefault\_a8\_set(long jarg1, HammingComputerDefault jarg1\_, long jarg2);

public final static native long HammingComputerDefault\_a8\_get(long jarg1, HammingComputerDefault jarg1\_);

public final static native void HammingComputerDefault\_quotient8\_set(long jarg1, HammingComputerDefault jarg1\_, int jarg2);

public final static native int HammingComputerDefault\_quotient8\_get(long jarg1, HammingComputerDefault jarg1\_);

public final static native void HammingComputerDefault\_remainder8\_set(long jarg1, HammingComputerDefault jarg1\_, int jarg2);

public final static native int HammingComputerDefault\_remainder8\_get(long jarg1, HammingComputerDefault jarg1\_);

public final static native long new\_HammingComputerDefault\_\_SWIG\_0();

public final static native long new\_HammingComputerDefault\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void HammingComputerDefault\_set(long jarg1, HammingComputerDefault jarg1\_, long jarg2, int jarg3);

public final static native int HammingComputerDefault\_hamming(long jarg1, HammingComputerDefault jarg1\_, long jarg2);

public final static native void delete\_HammingComputerDefault(long jarg1);

public final static native void HammingComputerM8\_a\_set(long jarg1, HammingComputerM8 jarg1\_, long jarg2);

public final static native long HammingComputerM8\_a\_get(long jarg1, HammingComputerM8 jarg1\_);

public final static native void HammingComputerM8\_n\_set(long jarg1, HammingComputerM8 jarg1\_, int jarg2);

public final static native int HammingComputerM8\_n\_get(long jarg1, HammingComputerM8 jarg1\_);

public final static native long new\_HammingComputerM8\_\_SWIG\_0();

public final static native long new\_HammingComputerM8\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void HammingComputerM8\_set(long jarg1, HammingComputerM8 jarg1\_, long jarg2, int jarg3);

public final static native int HammingComputerM8\_hamming(long jarg1, HammingComputerM8 jarg1\_, long jarg2);

public final static native void delete\_HammingComputerM8(long jarg1);

public final static native void HammingComputerM4\_a\_set(long jarg1, HammingComputerM4 jarg1\_, long jarg2);

public final static native long HammingComputerM4\_a\_get(long jarg1, HammingComputerM4 jarg1\_);

public final static native void HammingComputerM4\_n\_set(long jarg1, HammingComputerM4 jarg1\_, int jarg2);

public final static native int HammingComputerM4\_n\_get(long jarg1, HammingComputerM4 jarg1\_);

public final static native long new\_HammingComputerM4\_\_SWIG\_0();

public final static native long new\_HammingComputerM4\_\_SWIG\_1(long jarg1, int jarg2);

public final static native void HammingComputerM4\_set(long jarg1, HammingComputerM4 jarg1\_, long jarg2, int jarg3);

public final static native int HammingComputerM4\_hamming(long jarg1, HammingComputerM4 jarg1\_, long jarg2);

public final static native void delete\_HammingComputerM4(long jarg1);

public final static native int generalized\_hamming\_64(long jarg1);

public final static native void GenHammingComputer8\_a0\_set(long jarg1, GenHammingComputer8 jarg1\_, long jarg2);

public final static native long GenHammingComputer8\_a0\_get(long jarg1, GenHammingComputer8 jarg1\_);

public final static native long new\_GenHammingComputer8(long jarg1, int jarg2);

public final static native int GenHammingComputer8\_hamming(long jarg1, GenHammingComputer8 jarg1\_, long jarg2);

public final static native void delete\_GenHammingComputer8(long jarg1);

public final static native void GenHammingComputer16\_a0\_set(long jarg1, GenHammingComputer16 jarg1\_, long jarg2);

public final static native long GenHammingComputer16\_a0\_get(long jarg1, GenHammingComputer16 jarg1\_);

public final static native void GenHammingComputer16\_a1\_set(long jarg1, GenHammingComputer16 jarg1\_, long jarg2);

public final static native long GenHammingComputer16\_a1\_get(long jarg1, GenHammingComputer16 jarg1\_);

public final static native long new\_GenHammingComputer16(long jarg1, int jarg2);

public final static native int GenHammingComputer16\_hamming(long jarg1, GenHammingComputer16 jarg1\_, long jarg2);

public final static native void delete\_GenHammingComputer16(long jarg1);

public final static native void GenHammingComputer32\_a0\_set(long jarg1, GenHammingComputer32 jarg1\_, long jarg2);

public final static native long GenHammingComputer32\_a0\_get(long jarg1, GenHammingComputer32 jarg1\_);

public final static native void GenHammingComputer32\_a1\_set(long jarg1, GenHammingComputer32 jarg1\_, long jarg2);

public final static native long GenHammingComputer32\_a1\_get(long jarg1, GenHammingComputer32 jarg1\_);

public final static native void GenHammingComputer32\_a2\_set(long jarg1, GenHammingComputer32 jarg1\_, long jarg2);

public final static native long GenHammingComputer32\_a2\_get(long jarg1, GenHammingComputer32 jarg1\_);

public final static native void GenHammingComputer32\_a3\_set(long jarg1, GenHammingComputer32 jarg1\_, long jarg2);

public final static native long GenHammingComputer32\_a3\_get(long jarg1, GenHammingComputer32 jarg1\_);

public final static native long new\_GenHammingComputer32(long jarg1, int jarg2);

public final static native int GenHammingComputer32\_hamming(long jarg1, GenHammingComputer32 jarg1\_, long jarg2);

public final static native void delete\_GenHammingComputer32(long jarg1);

public final static native void GenHammingComputerM8\_a\_set(long jarg1, GenHammingComputerM8 jarg1\_, long jarg2);

public final static native long GenHammingComputerM8\_a\_get(long jarg1, GenHammingComputerM8 jarg1\_);

public final static native void GenHammingComputerM8\_n\_set(long jarg1, GenHammingComputerM8 jarg1\_, int jarg2);

public final static native int GenHammingComputerM8\_n\_get(long jarg1, GenHammingComputerM8 jarg1\_);

public final static native long new\_GenHammingComputerM8(long jarg1, int jarg2);

public final static native int GenHammingComputerM8\_hamming(long jarg1, GenHammingComputerM8 jarg1\_, long jarg2);

public final static native void delete\_GenHammingComputerM8(long jarg1);

public final static native void generalized\_hammings\_knn\_hc\_\_SWIG\_0(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5, int jarg6);

public final static native void generalized\_hammings\_knn\_hc\_\_SWIG\_1(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native void check\_compatible\_for\_merge(long jarg1, Index jarg1\_, long jarg2, Index jarg2\_);

public final static native long extract\_index\_ivf\_\_SWIG\_0(long jarg1, Index jarg1\_);

public final static native long try\_extract\_index\_ivf\_\_SWIG\_0(long jarg1, Index jarg1\_);

public final static native void merge\_into(long jarg1, Index jarg1\_, long jarg2, Index jarg2\_, boolean jarg3);

public final static native void search\_centroid(long jarg1, Index jarg1\_, long jarg2, int jarg3, long jarg4, LongVector jarg4\_);

public final static native void search\_and\_return\_centroids(long jarg1, Index jarg1\_, long jarg2, long jarg3, int jarg4, long jarg5, long jarg6, LongVector jarg6\_, long jarg7, LongVector jarg7\_, long jarg8, LongVector jarg8\_);

public final static native void SlidingIndexWindow\_index\_set(long jarg1, SlidingIndexWindow jarg1\_, long jarg2, Index jarg2\_);

public final static native long SlidingIndexWindow\_index\_get(long jarg1, SlidingIndexWindow jarg1\_);

public final static native void SlidingIndexWindow\_ils\_set(long jarg1, SlidingIndexWindow jarg1\_, long jarg2, ArrayInvertedLists jarg2\_);

public final static native long SlidingIndexWindow\_ils\_get(long jarg1, SlidingIndexWindow jarg1\_);

public final static native void SlidingIndexWindow\_n\_slice\_set(long jarg1, SlidingIndexWindow jarg1\_, int jarg2);

public final static native int SlidingIndexWindow\_n\_slice\_get(long jarg1, SlidingIndexWindow jarg1\_);

public final static native void SlidingIndexWindow\_nlist\_set(long jarg1, SlidingIndexWindow jarg1\_, long jarg2);

public final static native long SlidingIndexWindow\_nlist\_get(long jarg1, SlidingIndexWindow jarg1\_);

public final static native void SlidingIndexWindow\_sizes\_set(long jarg1, SlidingIndexWindow jarg1\_, long jarg2);

public final static native long SlidingIndexWindow\_sizes\_get(long jarg1, SlidingIndexWindow jarg1\_);

public final static native long new\_SlidingIndexWindow(long jarg1, Index jarg1\_);

public final static native void SlidingIndexWindow\_step(long jarg1, SlidingIndexWindow jarg1\_, long jarg2, Index jarg2\_, boolean jarg3);

public final static native void delete\_SlidingIndexWindow(long jarg1);

public final static native long get\_invlist\_range(long jarg1, Index jarg1\_, int jarg2, int jarg3);

public final static native void set\_invlist\_range(long jarg1, Index jarg1\_, int jarg2, int jarg3, long jarg4, ArrayInvertedLists jarg4\_);

public final static native void search\_with\_parameters\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_, long jarg7, IVFSearchParameters jarg7\_, long jarg8, long jarg9);

public final static native void search\_with\_parameters\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_, long jarg7, IVFSearchParameters jarg7\_, long jarg8);

public final static native void search\_with\_parameters\_\_SWIG\_2(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_, long jarg7, IVFSearchParameters jarg7\_);

public final static native void range\_search\_with\_parameters\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, RangeSearchResult jarg5\_, long jarg6, IVFSearchParameters jarg6\_, long jarg7, long jarg8);

public final static native void range\_search\_with\_parameters\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, RangeSearchResult jarg5\_, long jarg6, IVFSearchParameters jarg6\_, long jarg7);

public final static native void range\_search\_with\_parameters\_\_SWIG\_2(long jarg1, Index jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, RangeSearchResult jarg5\_, long jarg6, IVFSearchParameters jarg6\_);

public final static native void IndexScalarQuantizer\_sq\_set(long jarg1, IndexScalarQuantizer jarg1\_, long jarg2);

public final static native long IndexScalarQuantizer\_sq\_get(long jarg1, IndexScalarQuantizer jarg1\_);

public final static native long new\_IndexScalarQuantizer\_\_SWIG\_0(int jarg1, long jarg2, int jarg3);

public final static native long new\_IndexScalarQuantizer\_\_SWIG\_1(int jarg1, long jarg2);

public final static native long new\_IndexScalarQuantizer\_\_SWIG\_2();

public final static native void IndexScalarQuantizer\_train(long jarg1, IndexScalarQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void IndexScalarQuantizer\_search(long jarg1, IndexScalarQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native long IndexScalarQuantizer\_get\_distance\_computer(long jarg1, IndexScalarQuantizer jarg1\_);

public final static native void IndexScalarQuantizer\_sa\_encode(long jarg1, IndexScalarQuantizer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexScalarQuantizer\_sa\_decode(long jarg1, IndexScalarQuantizer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void delete\_IndexScalarQuantizer(long jarg1);

public final static native void IndexIVFScalarQuantizer\_sq\_set(long jarg1, IndexIVFScalarQuantizer jarg1\_, long jarg2);

public final static native long IndexIVFScalarQuantizer\_sq\_get(long jarg1, IndexIVFScalarQuantizer jarg1\_);

public final static native void IndexIVFScalarQuantizer\_by\_residual\_set(long jarg1, IndexIVFScalarQuantizer jarg1\_, boolean jarg2);

public final static native boolean IndexIVFScalarQuantizer\_by\_residual\_get(long jarg1, IndexIVFScalarQuantizer jarg1\_);

public final static native long new\_IndexIVFScalarQuantizer\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, int jarg5, boolean jarg6);

public final static native long new\_IndexIVFScalarQuantizer\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, int jarg5);

public final static native long new\_IndexIVFScalarQuantizer\_\_SWIG\_2(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long new\_IndexIVFScalarQuantizer\_\_SWIG\_3();

public final static native void IndexIVFScalarQuantizer\_train\_residual(long jarg1, IndexIVFScalarQuantizer jarg1\_, long jarg2, long jarg3);

public final static native void IndexIVFScalarQuantizer\_encode\_vectors\_\_SWIG\_0(long jarg1, IndexIVFScalarQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, boolean jarg6);

public final static native void IndexIVFScalarQuantizer\_encode\_vectors\_\_SWIG\_1(long jarg1, IndexIVFScalarQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void IndexIVFScalarQuantizer\_add\_core(long jarg1, IndexIVFScalarQuantizer jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, LongVector jarg5\_);

public final static native long IndexIVFScalarQuantizer\_get\_InvertedListScanner(long jarg1, IndexIVFScalarQuantizer jarg1\_, boolean jarg2);

public final static native void IndexIVFScalarQuantizer\_reconstruct\_from\_offset(long jarg1, IndexIVFScalarQuantizer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexIVFScalarQuantizer\_sa\_decode(long jarg1, IndexIVFScalarQuantizer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void delete\_IndexIVFScalarQuantizer(long jarg1);

public final static native void HNSW\_MinimaxHeap\_n\_set(long jarg1, HNSW.MinimaxHeap jarg1\_, int jarg2);

public final static native int HNSW\_MinimaxHeap\_n\_get(long jarg1, HNSW.MinimaxHeap jarg1\_);

public final static native void HNSW\_MinimaxHeap\_k\_set(long jarg1, HNSW.MinimaxHeap jarg1\_, int jarg2);

public final static native int HNSW\_MinimaxHeap\_k\_get(long jarg1, HNSW.MinimaxHeap jarg1\_);

public final static native void HNSW\_MinimaxHeap\_nvalid\_set(long jarg1, HNSW.MinimaxHeap jarg1\_, int jarg2);

public final static native int HNSW\_MinimaxHeap\_nvalid\_get(long jarg1, HNSW.MinimaxHeap jarg1\_);

public final static native void HNSW\_MinimaxHeap\_ids\_set(long jarg1, HNSW.MinimaxHeap jarg1\_, long jarg2, IntVector jarg2\_);

public final static native long HNSW\_MinimaxHeap\_ids\_get(long jarg1, HNSW.MinimaxHeap jarg1\_);

public final static native void HNSW\_MinimaxHeap\_dis\_set(long jarg1, HNSW.MinimaxHeap jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long HNSW\_MinimaxHeap\_dis\_get(long jarg1, HNSW.MinimaxHeap jarg1\_);

public final static native long new\_HNSW\_MinimaxHeap(int jarg1);

public final static native void HNSW\_MinimaxHeap\_push(long jarg1, HNSW.MinimaxHeap jarg1\_, int jarg2, float jarg3);

public final static native float HNSW\_MinimaxHeap\_max(long jarg1, HNSW.MinimaxHeap jarg1\_);

public final static native int HNSW\_MinimaxHeap\_size(long jarg1, HNSW.MinimaxHeap jarg1\_);

public final static native void HNSW\_MinimaxHeap\_clear(long jarg1, HNSW.MinimaxHeap jarg1\_);

public final static native int HNSW\_MinimaxHeap\_pop\_min\_\_SWIG\_0(long jarg1, HNSW.MinimaxHeap jarg1\_, long jarg2);

public final static native int HNSW\_MinimaxHeap\_pop\_min\_\_SWIG\_1(long jarg1, HNSW.MinimaxHeap jarg1\_);

public final static native int HNSW\_MinimaxHeap\_count\_below(long jarg1, HNSW.MinimaxHeap jarg1\_, float jarg2);

public final static native void delete\_HNSW\_MinimaxHeap(long jarg1);

public final static native void HNSW\_NodeDistCloser\_d\_set(long jarg1, HNSW.NodeDistCloser jarg1\_, float jarg2);

public final static native float HNSW\_NodeDistCloser\_d\_get(long jarg1, HNSW.NodeDistCloser jarg1\_);

public final static native void HNSW\_NodeDistCloser\_id\_set(long jarg1, HNSW.NodeDistCloser jarg1\_, int jarg2);

public final static native int HNSW\_NodeDistCloser\_id\_get(long jarg1, HNSW.NodeDistCloser jarg1\_);

public final static native long new\_HNSW\_NodeDistCloser(float jarg1, int jarg2);

public final static native void delete\_HNSW\_NodeDistCloser(long jarg1);

public final static native void HNSW\_NodeDistFarther\_d\_set(long jarg1, HNSW.NodeDistFarther jarg1\_, float jarg2);

public final static native float HNSW\_NodeDistFarther\_d\_get(long jarg1, HNSW.NodeDistFarther jarg1\_);

public final static native void HNSW\_NodeDistFarther\_id\_set(long jarg1, HNSW.NodeDistFarther jarg1\_, int jarg2);

public final static native int HNSW\_NodeDistFarther\_id\_get(long jarg1, HNSW.NodeDistFarther jarg1\_);

public final static native long new\_HNSW\_NodeDistFarther(float jarg1, int jarg2);

public final static native void delete\_HNSW\_NodeDistFarther(long jarg1);

public final static native void HNSW\_assign\_probas\_set(long jarg1, HNSW jarg1\_, long jarg2, DoubleVector jarg2\_);

public final static native long HNSW\_assign\_probas\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_cum\_nneighbor\_per\_level\_set(long jarg1, HNSW jarg1\_, long jarg2, IntVector jarg2\_);

public final static native long HNSW\_cum\_nneighbor\_per\_level\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_levels\_set(long jarg1, HNSW jarg1\_, long jarg2, IntVector jarg2\_);

public final static native long HNSW\_levels\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_offsets\_set(long jarg1, HNSW jarg1\_, long jarg2, Uint64Vector jarg2\_);

public final static native long HNSW\_offsets\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_neighbors\_set(long jarg1, HNSW jarg1\_, long jarg2, IntVector jarg2\_);

public final static native long HNSW\_neighbors\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_entry\_point\_set(long jarg1, HNSW jarg1\_, int jarg2);

public final static native int HNSW\_entry\_point\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_rng\_set(long jarg1, HNSW jarg1\_, long jarg2);

public final static native long HNSW\_rng\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_max\_level\_set(long jarg1, HNSW jarg1\_, int jarg2);

public final static native int HNSW\_max\_level\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_efConstruction\_set(long jarg1, HNSW jarg1\_, int jarg2);

public final static native int HNSW\_efConstruction\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_efSearch\_set(long jarg1, HNSW jarg1\_, int jarg2);

public final static native int HNSW\_efSearch\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_check\_relative\_distance\_set(long jarg1, HNSW jarg1\_, boolean jarg2);

public final static native boolean HNSW\_check\_relative\_distance\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_upper\_beam\_set(long jarg1, HNSW jarg1\_, int jarg2);

public final static native int HNSW\_upper\_beam\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_search\_bounded\_queue\_set(long jarg1, HNSW jarg1\_, boolean jarg2);

public final static native boolean HNSW\_search\_bounded\_queue\_get(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_set\_default\_probas(long jarg1, HNSW jarg1\_, int jarg2, float jarg3);

public final static native void HNSW\_set\_nb\_neighbors(long jarg1, HNSW jarg1\_, int jarg2, int jarg3);

public final static native int HNSW\_nb\_neighbors(long jarg1, HNSW jarg1\_, int jarg2);

public final static native int HNSW\_cum\_nb\_neighbors(long jarg1, HNSW jarg1\_, int jarg2);

public final static native void HNSW\_neighbor\_range(long jarg1, HNSW jarg1\_, long jarg2, int jarg3, long jarg4, long jarg5);

public final static native long new\_HNSW\_\_SWIG\_0(int jarg1);

public final static native long new\_HNSW\_\_SWIG\_1();

public final static native int HNSW\_random\_level(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_fill\_with\_random\_links(long jarg1, HNSW jarg1\_, long jarg2);

public final static native void HNSW\_add\_links\_starting\_from(long jarg1, HNSW jarg1\_, long jarg2, DistanceComputer jarg2\_, int jarg3, int jarg4, float jarg5, int jarg6, long jarg7, long jarg8, VisitedTable jarg8\_);

public final static native void HNSW\_add\_with\_locks(long jarg1, HNSW jarg1\_, long jarg2, DistanceComputer jarg2\_, int jarg3, int jarg4, long jarg5, long jarg6, VisitedTable jarg6\_);

public final static native int HNSW\_search\_from\_candidates\_\_SWIG\_0(long jarg1, HNSW jarg1\_, long jarg2, DistanceComputer jarg2\_, int jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6, HNSW.MinimaxHeap jarg6\_, long jarg7, VisitedTable jarg7\_, long jarg8, HNSWStats jarg8\_, int jarg9, int jarg10);

public final static native int HNSW\_search\_from\_candidates\_\_SWIG\_1(long jarg1, HNSW jarg1\_, long jarg2, DistanceComputer jarg2\_, int jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6, HNSW.MinimaxHeap jarg6\_, long jarg7, VisitedTable jarg7\_, long jarg8, HNSWStats jarg8\_, int jarg9);

public final static native long HNSW\_search\_from\_candidate\_unbounded(long jarg1, HNSW jarg1\_, long jarg2, long jarg3, DistanceComputer jarg3\_, int jarg4, long jarg5, VisitedTable jarg5\_, long jarg6, HNSWStats jarg6\_);

public final static native long HNSW\_search(long jarg1, HNSW jarg1\_, long jarg2, DistanceComputer jarg2\_, int jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6, VisitedTable jarg6\_);

public final static native void HNSW\_reset(long jarg1, HNSW jarg1\_);

public final static native void HNSW\_clear\_neighbor\_tables(long jarg1, HNSW jarg1\_, int jarg2);

public final static native void HNSW\_print\_neighbor\_stats(long jarg1, HNSW jarg1\_, int jarg2);

public final static native int HNSW\_prepare\_level\_tab\_\_SWIG\_0(long jarg1, HNSW jarg1\_, long jarg2, boolean jarg3);

public final static native int HNSW\_prepare\_level\_tab\_\_SWIG\_1(long jarg1, HNSW jarg1\_, long jarg2);

public final static native void HNSW\_shrink\_neighbor\_list(long jarg1, DistanceComputer jarg1\_, long jarg2, long jarg3, int jarg4);

public final static native void delete\_HNSW(long jarg1);

public final static native void HNSWStats\_n1\_set(long jarg1, HNSWStats jarg1\_, long jarg2);

public final static native long HNSWStats\_n1\_get(long jarg1, HNSWStats jarg1\_);

public final static native void HNSWStats\_n2\_set(long jarg1, HNSWStats jarg1\_, long jarg2);

public final static native long HNSWStats\_n2\_get(long jarg1, HNSWStats jarg1\_);

public final static native void HNSWStats\_n3\_set(long jarg1, HNSWStats jarg1\_, long jarg2);

public final static native long HNSWStats\_n3\_get(long jarg1, HNSWStats jarg1\_);

public final static native void HNSWStats\_ndis\_set(long jarg1, HNSWStats jarg1\_, long jarg2);

public final static native long HNSWStats\_ndis\_get(long jarg1, HNSWStats jarg1\_);

public final static native void HNSWStats\_nreorder\_set(long jarg1, HNSWStats jarg1\_, long jarg2);

public final static native long HNSWStats\_nreorder\_get(long jarg1, HNSWStats jarg1\_);

public final static native long new\_HNSWStats\_\_SWIG\_0(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native long new\_HNSWStats\_\_SWIG\_1(long jarg1, long jarg2, long jarg3, long jarg4);

public final static native long new\_HNSWStats\_\_SWIG\_2(long jarg1, long jarg2, long jarg3);

public final static native long new\_HNSWStats\_\_SWIG\_3(long jarg1, long jarg2);

public final static native long new\_HNSWStats\_\_SWIG\_4(long jarg1);

public final static native long new\_HNSWStats\_\_SWIG\_5();

public final static native void HNSWStats\_reset(long jarg1, HNSWStats jarg1\_);

public final static native void HNSWStats\_combine(long jarg1, HNSWStats jarg1\_, long jarg2, HNSWStats jarg2\_);

public final static native void delete\_HNSWStats(long jarg1);

public final static native void hnsw\_stats\_set(long jarg1, HNSWStats jarg1\_);

public final static native long hnsw\_stats\_get();

public final static native long ReconstructFromNeighbors\_index\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_M\_set(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2);

public final static native long ReconstructFromNeighbors\_M\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_k\_set(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2);

public final static native long ReconstructFromNeighbors\_k\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_nsq\_set(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2);

public final static native long ReconstructFromNeighbors\_nsq\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_code\_size\_set(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2);

public final static native long ReconstructFromNeighbors\_code\_size\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_k\_reorder\_set(long jarg1, ReconstructFromNeighbors jarg1\_, int jarg2);

public final static native int ReconstructFromNeighbors\_k\_reorder\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_codebook\_set(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long ReconstructFromNeighbors\_codebook\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_codes\_set(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2, ByteVector jarg2\_);

public final static native long ReconstructFromNeighbors\_codes\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_ntotal\_set(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2);

public final static native long ReconstructFromNeighbors\_ntotal\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_d\_set(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2);

public final static native long ReconstructFromNeighbors\_d\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native void ReconstructFromNeighbors\_dsub\_set(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2);

public final static native long ReconstructFromNeighbors\_dsub\_get(long jarg1, ReconstructFromNeighbors jarg1\_);

public final static native long new\_ReconstructFromNeighbors\_\_SWIG\_0(long jarg1, IndexHNSW jarg1\_, long jarg2, long jarg3);

public final static native long new\_ReconstructFromNeighbors\_\_SWIG\_1(long jarg1, IndexHNSW jarg1\_, long jarg2);

public final static native long new\_ReconstructFromNeighbors\_\_SWIG\_2(long jarg1, IndexHNSW jarg1\_);

public final static native void ReconstructFromNeighbors\_add\_codes(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2, long jarg3);

public final static native long ReconstructFromNeighbors\_compute\_distances(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2, long jarg3, LongVector jarg3\_, long jarg4, long jarg5);

public final static native void ReconstructFromNeighbors\_estimate\_code(long jarg1, ReconstructFromNeighbors jarg1\_, long jarg2, int jarg3, long jarg4);

public final static native void ReconstructFromNeighbors\_reconstruct(long jarg1, ReconstructFromNeighbors jarg1\_, int jarg2, long jarg3, long jarg4);

public final static native void ReconstructFromNeighbors\_reconstruct\_n(long jarg1, ReconstructFromNeighbors jarg1\_, int jarg2, int jarg3, long jarg4);

public final static native void ReconstructFromNeighbors\_get\_neighbor\_table(long jarg1, ReconstructFromNeighbors jarg1\_, int jarg2, long jarg3);

public final static native void delete\_ReconstructFromNeighbors(long jarg1);

public final static native void IndexHNSW\_hnsw\_set(long jarg1, IndexHNSW jarg1\_, long jarg2, HNSW jarg2\_);

public final static native long IndexHNSW\_hnsw\_get(long jarg1, IndexHNSW jarg1\_);

public final static native void IndexHNSW\_own\_fields\_set(long jarg1, IndexHNSW jarg1\_, boolean jarg2);

public final static native boolean IndexHNSW\_own\_fields\_get(long jarg1, IndexHNSW jarg1\_);

public final static native void IndexHNSW\_storage\_set(long jarg1, IndexHNSW jarg1\_, long jarg2, Index jarg2\_);

public final static native long IndexHNSW\_storage\_get(long jarg1, IndexHNSW jarg1\_);

public final static native void IndexHNSW\_reconstruct\_from\_neighbors\_set(long jarg1, IndexHNSW jarg1\_, long jarg2, ReconstructFromNeighbors jarg2\_);

public final static native long IndexHNSW\_reconstruct\_from\_neighbors\_get(long jarg1, IndexHNSW jarg1\_);

public final static native long new\_IndexHNSW\_\_SWIG\_0(int jarg1, int jarg2, int jarg3);

public final static native long new\_IndexHNSW\_\_SWIG\_1(int jarg1, int jarg2);

public final static native long new\_IndexHNSW\_\_SWIG\_2(int jarg1);

public final static native long new\_IndexHNSW\_\_SWIG\_3();

public final static native long new\_IndexHNSW\_\_SWIG\_4(long jarg1, Index jarg1\_, int jarg2);

public final static native long new\_IndexHNSW\_\_SWIG\_5(long jarg1, Index jarg1\_);

public final static native void delete\_IndexHNSW(long jarg1);

public final static native void IndexHNSW\_add(long jarg1, IndexHNSW jarg1\_, long jarg2, long jarg3);

public final static native void IndexHNSW\_train(long jarg1, IndexHNSW jarg1\_, long jarg2, long jarg3);

public final static native void IndexHNSW\_search(long jarg1, IndexHNSW jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexHNSW\_reconstruct(long jarg1, IndexHNSW jarg1\_, long jarg2, long jarg3);

public final static native void IndexHNSW\_reset(long jarg1, IndexHNSW jarg1\_);

public final static native void IndexHNSW\_shrink\_level\_0\_neighbors(long jarg1, IndexHNSW jarg1\_, int jarg2);

public final static native void IndexHNSW\_search\_level\_0\_\_SWIG\_0(long jarg1, IndexHNSW jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, int jarg9, int jarg10);

public final static native void IndexHNSW\_search\_level\_0\_\_SWIG\_1(long jarg1, IndexHNSW jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, int jarg9);

public final static native void IndexHNSW\_search\_level\_0\_\_SWIG\_2(long jarg1, IndexHNSW jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, long jarg7, long jarg8, LongVector jarg8\_);

public final static native void IndexHNSW\_init\_level\_0\_from\_knngraph(long jarg1, IndexHNSW jarg1\_, int jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void IndexHNSW\_init\_level\_0\_from\_entry\_points(long jarg1, IndexHNSW jarg1\_, int jarg2, long jarg3, long jarg4);

public final static native void IndexHNSW\_reorder\_links(long jarg1, IndexHNSW jarg1\_);

public final static native void IndexHNSW\_link\_singletons(long jarg1, IndexHNSW jarg1\_);

public final static native long new\_IndexHNSWFlat\_\_SWIG\_0();

public final static native long new\_IndexHNSWFlat\_\_SWIG\_1(int jarg1, int jarg2, int jarg3);

public final static native long new\_IndexHNSWFlat\_\_SWIG\_2(int jarg1, int jarg2);

public final static native void delete\_IndexHNSWFlat(long jarg1);

public final static native long new\_IndexHNSWPQ\_\_SWIG\_0();

public final static native long new\_IndexHNSWPQ\_\_SWIG\_1(int jarg1, int jarg2, int jarg3);

public final static native void IndexHNSWPQ\_train(long jarg1, IndexHNSWPQ jarg1\_, long jarg2, long jarg3);

public final static native void delete\_IndexHNSWPQ(long jarg1);

public final static native long new\_IndexHNSWSQ\_\_SWIG\_0();

public final static native long new\_IndexHNSWSQ\_\_SWIG\_1(int jarg1, long jarg2, int jarg3, int jarg4);

public final static native long new\_IndexHNSWSQ\_\_SWIG\_2(int jarg1, long jarg2, int jarg3);

public final static native void delete\_IndexHNSWSQ(long jarg1);

public final static native long new\_IndexHNSW2Level\_\_SWIG\_0();

public final static native long new\_IndexHNSW2Level\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2, int jarg3, int jarg4);

public final static native void IndexHNSW2Level\_flip\_to\_ivf(long jarg1, IndexHNSW2Level jarg1\_);

public final static native void IndexHNSW2Level\_search(long jarg1, IndexHNSW2Level jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void delete\_IndexHNSW2Level(long jarg1);

public final static native long new\_IndexIVFFlat\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2, long jarg3, int jarg4);

public final static native long new\_IndexIVFFlat\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2, long jarg3);

public final static native void IndexIVFFlat\_add\_core(long jarg1, IndexIVFFlat jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, LongVector jarg5\_);

public final static native void IndexIVFFlat\_encode\_vectors\_\_SWIG\_0(long jarg1, IndexIVFFlat jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, boolean jarg6);

public final static native void IndexIVFFlat\_encode\_vectors\_\_SWIG\_1(long jarg1, IndexIVFFlat jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native long IndexIVFFlat\_get\_InvertedListScanner(long jarg1, IndexIVFFlat jarg1\_, boolean jarg2);

public final static native void IndexIVFFlat\_reconstruct\_from\_offset(long jarg1, IndexIVFFlat jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexIVFFlat\_sa\_decode(long jarg1, IndexIVFFlat jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long new\_IndexIVFFlat\_\_SWIG\_2();

public final static native void delete\_IndexIVFFlat(long jarg1);

public final static native void IndexIVFFlatDedup\_instances\_set(long jarg1, IndexIVFFlatDedup jarg1\_, long jarg2);

public final static native long IndexIVFFlatDedup\_instances\_get(long jarg1, IndexIVFFlatDedup jarg1\_);

public final static native long new\_IndexIVFFlatDedup\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2, long jarg3, int jarg4);

public final static native long new\_IndexIVFFlatDedup\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2, long jarg3);

public final static native void IndexIVFFlatDedup\_train(long jarg1, IndexIVFFlatDedup jarg1\_, long jarg2, long jarg3);

public final static native void IndexIVFFlatDedup\_add\_with\_ids(long jarg1, IndexIVFFlatDedup jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void IndexIVFFlatDedup\_search\_preassigned\_\_SWIG\_0(long jarg1, IndexIVFFlatDedup jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, boolean jarg9, long jarg10, IVFSearchParameters jarg10\_, long jarg11, IndexIVFStats jarg11\_);

public final static native void IndexIVFFlatDedup\_search\_preassigned\_\_SWIG\_1(long jarg1, IndexIVFFlatDedup jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, boolean jarg9, long jarg10, IVFSearchParameters jarg10\_);

public final static native void IndexIVFFlatDedup\_search\_preassigned\_\_SWIG\_2(long jarg1, IndexIVFFlatDedup jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, boolean jarg9);

public final static native long IndexIVFFlatDedup\_remove\_ids(long jarg1, IndexIVFFlatDedup jarg1\_, long jarg2, IDSelector jarg2\_);

public final static native void IndexIVFFlatDedup\_range\_search(long jarg1, IndexIVFFlatDedup jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, RangeSearchResult jarg5\_);

public final static native void IndexIVFFlatDedup\_update\_vectors(long jarg1, IndexIVFFlatDedup jarg1\_, int jarg2, long jarg3, LongVector jarg3\_, long jarg4);

public final static native void IndexIVFFlatDedup\_reconstruct\_from\_offset(long jarg1, IndexIVFFlatDedup jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long new\_IndexIVFFlatDedup\_\_SWIG\_2();

public final static native void delete\_IndexIVFFlatDedup(long jarg1);

public final static native void OnDiskOneList\_size\_set(long jarg1, OnDiskOneList jarg1\_, long jarg2);

public final static native long OnDiskOneList\_size\_get(long jarg1, OnDiskOneList jarg1\_);

public final static native void OnDiskOneList\_capacity\_set(long jarg1, OnDiskOneList jarg1\_, long jarg2);

public final static native long OnDiskOneList\_capacity\_get(long jarg1, OnDiskOneList jarg1\_);

public final static native void OnDiskOneList\_offset\_set(long jarg1, OnDiskOneList jarg1\_, long jarg2);

public final static native long OnDiskOneList\_offset\_get(long jarg1, OnDiskOneList jarg1\_);

public final static native long new\_OnDiskOneList();

public final static native void delete\_OnDiskOneList(long jarg1);

public final static native void OnDiskInvertedLists\_lists\_set(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_lists\_get(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native void OnDiskInvertedLists\_Slot\_offset\_set(long jarg1, OnDiskInvertedLists.Slot jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_Slot\_offset\_get(long jarg1, OnDiskInvertedLists.Slot jarg1\_);

public final static native void OnDiskInvertedLists\_Slot\_capacity\_set(long jarg1, OnDiskInvertedLists.Slot jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_Slot\_capacity\_get(long jarg1, OnDiskInvertedLists.Slot jarg1\_);

public final static native long new\_OnDiskInvertedLists\_Slot\_\_SWIG\_0(long jarg1, long jarg2);

public final static native long new\_OnDiskInvertedLists\_Slot\_\_SWIG\_1();

public final static native void delete\_OnDiskInvertedLists\_Slot(long jarg1);

public final static native void OnDiskInvertedLists\_slots\_set(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_slots\_get(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native void OnDiskInvertedLists\_filename\_set(long jarg1, OnDiskInvertedLists jarg1\_, String jarg2);

public final static native String OnDiskInvertedLists\_filename\_get(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native void OnDiskInvertedLists\_totsize\_set(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_totsize\_get(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native void OnDiskInvertedLists\_ptr\_set(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_ptr\_get(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native void OnDiskInvertedLists\_read\_only\_set(long jarg1, OnDiskInvertedLists jarg1\_, boolean jarg2);

public final static native boolean OnDiskInvertedLists\_read\_only\_get(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native long new\_OnDiskInvertedLists\_\_SWIG\_0(long jarg1, long jarg2, String jarg3);

public final static native long OnDiskInvertedLists\_list\_size(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_get\_codes(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_get\_ids(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_add\_entries(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void OnDiskInvertedLists\_update\_entries(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6);

public final static native void OnDiskInvertedLists\_resize(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long OnDiskInvertedLists\_merge\_from\_\_SWIG\_0(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, int jarg3, boolean jarg4);

public final static native long OnDiskInvertedLists\_merge\_from\_\_SWIG\_1(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, int jarg3);

public final static native long OnDiskInvertedLists\_merge\_from\_1\_\_SWIG\_0(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, InvertedLists jarg2\_, boolean jarg3);

public final static native long OnDiskInvertedLists\_merge\_from\_1\_\_SWIG\_1(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native void OnDiskInvertedLists\_crop\_invlists(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void OnDiskInvertedLists\_prefetch\_lists(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, LongVector jarg2\_, int jarg3);

public final static native void delete\_OnDiskInvertedLists(long jarg1);

public final static native void OnDiskInvertedLists\_locks\_set(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_locks\_get(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native void OnDiskInvertedLists\_pf\_set(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long OnDiskInvertedLists\_pf\_get(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native void OnDiskInvertedLists\_prefetch\_nthread\_set(long jarg1, OnDiskInvertedLists jarg1\_, int jarg2);

public final static native int OnDiskInvertedLists\_prefetch\_nthread\_get(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native void OnDiskInvertedLists\_do\_mmap(long jarg1, OnDiskInvertedLists jarg1\_);

public final static native void OnDiskInvertedLists\_update\_totsize(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native void OnDiskInvertedLists\_resize\_locked(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native long OnDiskInvertedLists\_allocate\_slot(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native void OnDiskInvertedLists\_free\_slot(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2, long jarg3);

public final static native void OnDiskInvertedLists\_set\_all\_lists\_sizes(long jarg1, OnDiskInvertedLists jarg1\_, long jarg2);

public final static native long new\_OnDiskInvertedLists\_\_SWIG\_1();

public final static native long new\_OnDiskInvertedListsIOHook();

public final static native void OnDiskInvertedListsIOHook\_write(long jarg1, OnDiskInvertedListsIOHook jarg1\_, long jarg2, InvertedLists jarg2\_, long jarg3);

public final static native long OnDiskInvertedListsIOHook\_read(long jarg1, OnDiskInvertedListsIOHook jarg1\_, long jarg2, int jarg3);

public final static native long OnDiskInvertedListsIOHook\_read\_ArrayInvertedLists(long jarg1, OnDiskInvertedListsIOHook jarg1\_, long jarg2, int jarg3, long jarg4, long jarg5, long jarg6, Uint64Vector jarg6\_);

public final static native void delete\_OnDiskInvertedListsIOHook(long jarg1);

public final static native void IVFPQSearchParameters\_scan\_table\_threshold\_set(long jarg1, IVFPQSearchParameters jarg1\_, long jarg2);

public final static native long IVFPQSearchParameters\_scan\_table\_threshold\_get(long jarg1, IVFPQSearchParameters jarg1\_);

public final static native void IVFPQSearchParameters\_polysemous\_ht\_set(long jarg1, IVFPQSearchParameters jarg1\_, int jarg2);

public final static native int IVFPQSearchParameters\_polysemous\_ht\_get(long jarg1, IVFPQSearchParameters jarg1\_);

public final static native long new\_IVFPQSearchParameters();

public final static native void delete\_IVFPQSearchParameters(long jarg1);

public final static native void precomputed\_table\_max\_bytes\_set(long jarg1);

public final static native long precomputed\_table\_max\_bytes\_get();

public final static native void IndexIVFPQ\_by\_residual\_set(long jarg1, IndexIVFPQ jarg1\_, boolean jarg2);

public final static native boolean IndexIVFPQ\_by\_residual\_get(long jarg1, IndexIVFPQ jarg1\_);

public final static native void IndexIVFPQ\_pq\_set(long jarg1, IndexIVFPQ jarg1\_, long jarg2, ProductQuantizer jarg2\_);

public final static native long IndexIVFPQ\_pq\_get(long jarg1, IndexIVFPQ jarg1\_);

public final static native void IndexIVFPQ\_do\_polysemous\_training\_set(long jarg1, IndexIVFPQ jarg1\_, boolean jarg2);

public final static native boolean IndexIVFPQ\_do\_polysemous\_training\_get(long jarg1, IndexIVFPQ jarg1\_);

public final static native void IndexIVFPQ\_polysemous\_training\_set(long jarg1, IndexIVFPQ jarg1\_, long jarg2, PolysemousTraining jarg2\_);

public final static native long IndexIVFPQ\_polysemous\_training\_get(long jarg1, IndexIVFPQ jarg1\_);

public final static native void IndexIVFPQ\_scan\_table\_threshold\_set(long jarg1, IndexIVFPQ jarg1\_, long jarg2);

public final static native long IndexIVFPQ\_scan\_table\_threshold\_get(long jarg1, IndexIVFPQ jarg1\_);

public final static native void IndexIVFPQ\_polysemous\_ht\_set(long jarg1, IndexIVFPQ jarg1\_, int jarg2);

public final static native int IndexIVFPQ\_polysemous\_ht\_get(long jarg1, IndexIVFPQ jarg1\_);

public final static native void IndexIVFPQ\_use\_precomputed\_table\_set(long jarg1, IndexIVFPQ jarg1\_, int jarg2);

public final static native int IndexIVFPQ\_use\_precomputed\_table\_get(long jarg1, IndexIVFPQ jarg1\_);

public final static native void IndexIVFPQ\_precomputed\_table\_set(long jarg1, IndexIVFPQ jarg1\_, long jarg2);

public final static native long IndexIVFPQ\_precomputed\_table\_get(long jarg1, IndexIVFPQ jarg1\_);

public final static native long new\_IndexIVFPQ\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, int jarg6);

public final static native long new\_IndexIVFPQ\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native void IndexIVFPQ\_encode\_vectors\_\_SWIG\_0(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, boolean jarg6);

public final static native void IndexIVFPQ\_encode\_vectors\_\_SWIG\_1(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void IndexIVFPQ\_sa\_decode(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexIVFPQ\_add\_core(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, LongVector jarg5\_);

public final static native void IndexIVFPQ\_add\_core\_o\_\_SWIG\_0(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexIVFPQ\_add\_core\_o\_\_SWIG\_1(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void IndexIVFPQ\_train\_residual(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3);

public final static native void IndexIVFPQ\_train\_residual\_o(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexIVFPQ\_reconstruct\_from\_offset(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long IndexIVFPQ\_find\_duplicates(long jarg1, IndexIVFPQ jarg1\_, long jarg2, LongVector jarg2\_, long jarg3);

public final static native void IndexIVFPQ\_encode(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexIVFPQ\_encode\_multiple\_\_SWIG\_0(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, LongVector jarg3\_, long jarg4, long jarg5, boolean jarg6);

public final static native void IndexIVFPQ\_encode\_multiple\_\_SWIG\_1(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, LongVector jarg3\_, long jarg4, long jarg5);

public final static native void IndexIVFPQ\_decode\_multiple(long jarg1, IndexIVFPQ jarg1\_, long jarg2, long jarg3, LongVector jarg3\_, long jarg4, long jarg5);

public final static native long IndexIVFPQ\_get\_InvertedListScanner(long jarg1, IndexIVFPQ jarg1\_, boolean jarg2);

public final static native void IndexIVFPQ\_precompute\_table(long jarg1, IndexIVFPQ jarg1\_);

public final static native long new\_IndexIVFPQ\_\_SWIG\_2();

public final static native void delete\_IndexIVFPQ(long jarg1);

public final static native void initialize\_IVFPQ\_precomputed\_table(long jarg1, long jarg2, Index jarg2\_, long jarg3, ProductQuantizer jarg3\_, long jarg4, boolean jarg5);

public final static native void IndexIVFPQStats\_nrefine\_set(long jarg1, IndexIVFPQStats jarg1\_, long jarg2);

public final static native long IndexIVFPQStats\_nrefine\_get(long jarg1, IndexIVFPQStats jarg1\_);

public final static native void IndexIVFPQStats\_n\_hamming\_pass\_set(long jarg1, IndexIVFPQStats jarg1\_, long jarg2);

public final static native long IndexIVFPQStats\_n\_hamming\_pass\_get(long jarg1, IndexIVFPQStats jarg1\_);

public final static native void IndexIVFPQStats\_search\_cycles\_set(long jarg1, IndexIVFPQStats jarg1\_, long jarg2);

public final static native long IndexIVFPQStats\_search\_cycles\_get(long jarg1, IndexIVFPQStats jarg1\_);

public final static native void IndexIVFPQStats\_refine\_cycles\_set(long jarg1, IndexIVFPQStats jarg1\_, long jarg2);

public final static native long IndexIVFPQStats\_refine\_cycles\_get(long jarg1, IndexIVFPQStats jarg1\_);

public final static native long new\_IndexIVFPQStats();

public final static native void IndexIVFPQStats\_reset(long jarg1, IndexIVFPQStats jarg1\_);

public final static native void delete\_IndexIVFPQStats(long jarg1);

public final static native void indexIVFPQ\_stats\_set(long jarg1, IndexIVFPQStats jarg1\_);

public final static native long indexIVFPQ\_stats\_get();

public final static native void IndexBinary\_d\_set(long jarg1, IndexBinary jarg1\_, int jarg2);

public final static native int IndexBinary\_d\_get(long jarg1, IndexBinary jarg1\_);

public final static native void IndexBinary\_code\_size\_set(long jarg1, IndexBinary jarg1\_, int jarg2);

public final static native int IndexBinary\_code\_size\_get(long jarg1, IndexBinary jarg1\_);

public final static native void IndexBinary\_ntotal\_set(long jarg1, IndexBinary jarg1\_, long jarg2);

public final static native long IndexBinary\_ntotal\_get(long jarg1, IndexBinary jarg1\_);

public final static native void IndexBinary\_verbose\_set(long jarg1, IndexBinary jarg1\_, boolean jarg2);

public final static native boolean IndexBinary\_verbose\_get(long jarg1, IndexBinary jarg1\_);

public final static native void IndexBinary\_is\_trained\_set(long jarg1, IndexBinary jarg1\_, boolean jarg2);

public final static native boolean IndexBinary\_is\_trained\_get(long jarg1, IndexBinary jarg1\_);

public final static native void IndexBinary\_metric\_type\_set(long jarg1, IndexBinary jarg1\_, int jarg2);

public final static native int IndexBinary\_metric\_type\_get(long jarg1, IndexBinary jarg1\_);

public final static native void delete\_IndexBinary(long jarg1);

public final static native void IndexBinary\_train(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinary\_add(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinary\_add\_with\_ids(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void IndexBinary\_search(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexBinary\_range\_search(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3, int jarg4, long jarg5, RangeSearchResult jarg5\_);

public final static native void IndexBinary\_assign\_\_SWIG\_0(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void IndexBinary\_assign\_\_SWIG\_1(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void IndexBinary\_reset(long jarg1, IndexBinary jarg1\_);

public final static native long IndexBinary\_remove\_ids(long jarg1, IndexBinary jarg1\_, long jarg2, IDSelector jarg2\_);

public final static native void IndexBinary\_reconstruct(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinary\_reconstruct\_n(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexBinary\_search\_and\_reconstruct(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_, long jarg7);

public final static native void IndexBinary\_display(long jarg1, IndexBinary jarg1\_);

public final static native void Index2Layer\_q1\_set(long jarg1, Index2Layer jarg1\_, long jarg2, Level1Quantizer jarg2\_);

public final static native long Index2Layer\_q1\_get(long jarg1, Index2Layer jarg1\_);

public final static native void Index2Layer\_pq\_set(long jarg1, Index2Layer jarg1\_, long jarg2, ProductQuantizer jarg2\_);

public final static native long Index2Layer\_pq\_get(long jarg1, Index2Layer jarg1\_);

public final static native void Index2Layer\_code\_size\_1\_set(long jarg1, Index2Layer jarg1\_, long jarg2);

public final static native long Index2Layer\_code\_size\_1\_get(long jarg1, Index2Layer jarg1\_);

public final static native void Index2Layer\_code\_size\_2\_set(long jarg1, Index2Layer jarg1\_, long jarg2);

public final static native long Index2Layer\_code\_size\_2\_get(long jarg1, Index2Layer jarg1\_);

public final static native long new\_Index2Layer\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2, int jarg3, int jarg4, int jarg5);

public final static native long new\_Index2Layer\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2, int jarg3, int jarg4);

public final static native long new\_Index2Layer\_\_SWIG\_2(long jarg1, Index jarg1\_, long jarg2, int jarg3);

public final static native long new\_Index2Layer\_\_SWIG\_3();

public final static native void delete\_Index2Layer(long jarg1);

public final static native void Index2Layer\_train(long jarg1, Index2Layer jarg1\_, long jarg2, long jarg3);

public final static native void Index2Layer\_search(long jarg1, Index2Layer jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native long Index2Layer\_get\_distance\_computer(long jarg1, Index2Layer jarg1\_);

public final static native void Index2Layer\_transfer\_to\_IVFPQ(long jarg1, Index2Layer jarg1\_, long jarg2, IndexIVFPQ jarg2\_);

public final static native void Index2Layer\_sa\_encode(long jarg1, Index2Layer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void Index2Layer\_sa\_decode(long jarg1, Index2Layer jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexBinaryFlat\_xb\_set(long jarg1, IndexBinaryFlat jarg1\_, long jarg2, ByteVector jarg2\_);

public final static native long IndexBinaryFlat\_xb\_get(long jarg1, IndexBinaryFlat jarg1\_);

public final static native void IndexBinaryFlat\_use\_heap\_set(long jarg1, IndexBinaryFlat jarg1\_, boolean jarg2);

public final static native boolean IndexBinaryFlat\_use\_heap\_get(long jarg1, IndexBinaryFlat jarg1\_);

public final static native void IndexBinaryFlat\_query\_batch\_size\_set(long jarg1, IndexBinaryFlat jarg1\_, long jarg2);

public final static native long IndexBinaryFlat\_query\_batch\_size\_get(long jarg1, IndexBinaryFlat jarg1\_);

public final static native long new\_IndexBinaryFlat\_\_SWIG\_0(long jarg1);

public final static native void IndexBinaryFlat\_add(long jarg1, IndexBinaryFlat jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinaryFlat\_reset(long jarg1, IndexBinaryFlat jarg1\_);

public final static native void IndexBinaryFlat\_search(long jarg1, IndexBinaryFlat jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexBinaryFlat\_range\_search(long jarg1, IndexBinaryFlat jarg1\_, long jarg2, long jarg3, int jarg4, long jarg5, RangeSearchResult jarg5\_);

public final static native void IndexBinaryFlat\_reconstruct(long jarg1, IndexBinaryFlat jarg1\_, long jarg2, long jarg3);

public final static native long IndexBinaryFlat\_remove\_ids(long jarg1, IndexBinaryFlat jarg1\_, long jarg2, IDSelector jarg2\_);

public final static native long new\_IndexBinaryFlat\_\_SWIG\_1();

public final static native void delete\_IndexBinaryFlat(long jarg1);

public final static native void IndexBinaryIVF\_invlists\_set(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native long IndexBinaryIVF\_invlists\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_own\_invlists\_set(long jarg1, IndexBinaryIVF jarg1\_, boolean jarg2);

public final static native boolean IndexBinaryIVF\_own\_invlists\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_nprobe\_set(long jarg1, IndexBinaryIVF jarg1\_, long jarg2);

public final static native long IndexBinaryIVF\_nprobe\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_max\_codes\_set(long jarg1, IndexBinaryIVF jarg1\_, long jarg2);

public final static native long IndexBinaryIVF\_max\_codes\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_use\_heap\_set(long jarg1, IndexBinaryIVF jarg1\_, boolean jarg2);

public final static native boolean IndexBinaryIVF\_use\_heap\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_direct\_map\_set(long jarg1, IndexBinaryIVF jarg1\_, long jarg2);

public final static native long IndexBinaryIVF\_direct\_map\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_quantizer\_set(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, IndexBinary jarg2\_);

public final static native long IndexBinaryIVF\_quantizer\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_nlist\_set(long jarg1, IndexBinaryIVF jarg1\_, long jarg2);

public final static native long IndexBinaryIVF\_nlist\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_own\_fields\_set(long jarg1, IndexBinaryIVF jarg1\_, boolean jarg2);

public final static native boolean IndexBinaryIVF\_own\_fields\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_cp\_set(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, ClusteringParameters jarg2\_);

public final static native long IndexBinaryIVF\_cp\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_clustering\_index\_set(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, Index jarg2\_);

public final static native long IndexBinaryIVF\_clustering\_index\_get(long jarg1, IndexBinaryIVF jarg1\_);

public final static native long new\_IndexBinaryIVF\_\_SWIG\_0(long jarg1, IndexBinary jarg1\_, long jarg2, long jarg3);

public final static native long new\_IndexBinaryIVF\_\_SWIG\_1();

public final static native void delete\_IndexBinaryIVF(long jarg1);

public final static native void IndexBinaryIVF\_reset(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_train(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinaryIVF\_add(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinaryIVF\_add\_with\_ids(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void IndexBinaryIVF\_add\_core(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, LongVector jarg5\_);

public final static native void IndexBinaryIVF\_search\_preassigned\_\_SWIG\_0(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, boolean jarg9, long jarg10, IVFSearchParameters jarg10\_);

public final static native void IndexBinaryIVF\_search\_preassigned\_\_SWIG\_1(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, long jarg8, LongVector jarg8\_, boolean jarg9);

public final static native long IndexBinaryIVF\_get\_InvertedListScanner\_\_SWIG\_0(long jarg1, IndexBinaryIVF jarg1\_, boolean jarg2);

public final static native long IndexBinaryIVF\_get\_InvertedListScanner\_\_SWIG\_1(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_search(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexBinaryIVF\_range\_search(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, int jarg4, long jarg5, RangeSearchResult jarg5\_);

public final static native void IndexBinaryIVF\_range\_search\_preassigned(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, int jarg4, long jarg5, LongVector jarg5\_, long jarg6, long jarg7, RangeSearchResult jarg7\_);

public final static native void IndexBinaryIVF\_reconstruct(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinaryIVF\_reconstruct\_n(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexBinaryIVF\_search\_and\_reconstruct(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_, long jarg7);

public final static native void IndexBinaryIVF\_reconstruct\_from\_offset(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long IndexBinaryIVF\_remove\_ids(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, IDSelector jarg2\_);

public final static native void IndexBinaryIVF\_merge\_from(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, IndexBinaryIVF jarg2\_, long jarg3);

public final static native long IndexBinaryIVF\_get\_list\_size(long jarg1, IndexBinaryIVF jarg1\_, long jarg2);

public final static native void IndexBinaryIVF\_make\_direct\_map\_\_SWIG\_0(long jarg1, IndexBinaryIVF jarg1\_, boolean jarg2);

public final static native void IndexBinaryIVF\_make\_direct\_map\_\_SWIG\_1(long jarg1, IndexBinaryIVF jarg1\_);

public final static native void IndexBinaryIVF\_set\_direct\_map\_type(long jarg1, IndexBinaryIVF jarg1\_, long jarg2);

public final static native void IndexBinaryIVF\_replace\_invlists\_\_SWIG\_0(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, InvertedLists jarg2\_, boolean jarg3);

public final static native void IndexBinaryIVF\_replace\_invlists\_\_SWIG\_1(long jarg1, IndexBinaryIVF jarg1\_, long jarg2, InvertedLists jarg2\_);

public final static native void IndexBinaryFromFloat\_index\_set(long jarg1, IndexBinaryFromFloat jarg1\_, long jarg2, Index jarg2\_);

public final static native long IndexBinaryFromFloat\_index\_get(long jarg1, IndexBinaryFromFloat jarg1\_);

public final static native void IndexBinaryFromFloat\_own\_fields\_set(long jarg1, IndexBinaryFromFloat jarg1\_, boolean jarg2);

public final static native boolean IndexBinaryFromFloat\_own\_fields\_get(long jarg1, IndexBinaryFromFloat jarg1\_);

public final static native long new\_IndexBinaryFromFloat\_\_SWIG\_0();

public final static native long new\_IndexBinaryFromFloat\_\_SWIG\_1(long jarg1, Index jarg1\_);

public final static native void delete\_IndexBinaryFromFloat(long jarg1);

public final static native void IndexBinaryFromFloat\_add(long jarg1, IndexBinaryFromFloat jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinaryFromFloat\_reset(long jarg1, IndexBinaryFromFloat jarg1\_);

public final static native void IndexBinaryFromFloat\_search(long jarg1, IndexBinaryFromFloat jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexBinaryFromFloat\_train(long jarg1, IndexBinaryFromFloat jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinaryHNSW\_hnsw\_set(long jarg1, IndexBinaryHNSW jarg1\_, long jarg2, HNSW jarg2\_);

public final static native long IndexBinaryHNSW\_hnsw\_get(long jarg1, IndexBinaryHNSW jarg1\_);

public final static native void IndexBinaryHNSW\_own\_fields\_set(long jarg1, IndexBinaryHNSW jarg1\_, boolean jarg2);

public final static native boolean IndexBinaryHNSW\_own\_fields\_get(long jarg1, IndexBinaryHNSW jarg1\_);

public final static native void IndexBinaryHNSW\_storage\_set(long jarg1, IndexBinaryHNSW jarg1\_, long jarg2, IndexBinary jarg2\_);

public final static native long IndexBinaryHNSW\_storage\_get(long jarg1, IndexBinaryHNSW jarg1\_);

public final static native long new\_IndexBinaryHNSW\_\_SWIG\_0();

public final static native long new\_IndexBinaryHNSW\_\_SWIG\_1(int jarg1, int jarg2);

public final static native long new\_IndexBinaryHNSW\_\_SWIG\_2(int jarg1);

public final static native long new\_IndexBinaryHNSW\_\_SWIG\_3(long jarg1, IndexBinary jarg1\_, int jarg2);

public final static native long new\_IndexBinaryHNSW\_\_SWIG\_4(long jarg1, IndexBinary jarg1\_);

public final static native void delete\_IndexBinaryHNSW(long jarg1);

public final static native long IndexBinaryHNSW\_get\_distance\_computer(long jarg1, IndexBinaryHNSW jarg1\_);

public final static native void IndexBinaryHNSW\_add(long jarg1, IndexBinaryHNSW jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinaryHNSW\_train(long jarg1, IndexBinaryHNSW jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinaryHNSW\_search(long jarg1, IndexBinaryHNSW jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexBinaryHNSW\_reconstruct(long jarg1, IndexBinaryHNSW jarg1\_, long jarg2, long jarg3);

public final static native void IndexBinaryHNSW\_reset(long jarg1, IndexBinaryHNSW jarg1\_);

public final static native void IndexRefine\_base\_index\_set(long jarg1, IndexRefine jarg1\_, long jarg2, Index jarg2\_);

public final static native long IndexRefine\_base\_index\_get(long jarg1, IndexRefine jarg1\_);

public final static native void IndexRefine\_refine\_index\_set(long jarg1, IndexRefine jarg1\_, long jarg2, Index jarg2\_);

public final static native long IndexRefine\_refine\_index\_get(long jarg1, IndexRefine jarg1\_);

public final static native void IndexRefine\_own\_fields\_set(long jarg1, IndexRefine jarg1\_, boolean jarg2);

public final static native boolean IndexRefine\_own\_fields\_get(long jarg1, IndexRefine jarg1\_);

public final static native void IndexRefine\_own\_refine\_index\_set(long jarg1, IndexRefine jarg1\_, boolean jarg2);

public final static native boolean IndexRefine\_own\_refine\_index\_get(long jarg1, IndexRefine jarg1\_);

public final static native void IndexRefine\_k\_factor\_set(long jarg1, IndexRefine jarg1\_, float jarg2);

public final static native float IndexRefine\_k\_factor\_get(long jarg1, IndexRefine jarg1\_);

public final static native long new\_IndexRefine\_\_SWIG\_0(long jarg1, Index jarg1\_, long jarg2, Index jarg2\_);

public final static native long new\_IndexRefine\_\_SWIG\_1();

public final static native void IndexRefine\_train(long jarg1, IndexRefine jarg1\_, long jarg2, long jarg3);

public final static native void IndexRefine\_add(long jarg1, IndexRefine jarg1\_, long jarg2, long jarg3);

public final static native void IndexRefine\_reset(long jarg1, IndexRefine jarg1\_);

public final static native void IndexRefine\_search(long jarg1, IndexRefine jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexRefine\_reconstruct(long jarg1, IndexRefine jarg1\_, long jarg2, long jarg3);

public final static native long IndexRefine\_sa\_code\_size(long jarg1, IndexRefine jarg1\_);

public final static native void IndexRefine\_sa\_encode(long jarg1, IndexRefine jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void IndexRefine\_sa\_decode(long jarg1, IndexRefine jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void delete\_IndexRefine(long jarg1);

public final static native long new\_IndexRefineFlat\_\_SWIG\_0(long jarg1, Index jarg1\_);

public final static native long new\_IndexRefineFlat\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2);

public final static native long new\_IndexRefineFlat\_\_SWIG\_2();

public final static native void IndexRefineFlat\_search(long jarg1, IndexRefineFlat jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void delete\_IndexRefineFlat(long jarg1);

public final static native void IndexSplitVectors\_own\_fields\_set(long jarg1, IndexSplitVectors jarg1\_, boolean jarg2);

public final static native boolean IndexSplitVectors\_own\_fields\_get(long jarg1, IndexSplitVectors jarg1\_);

public final static native void IndexSplitVectors\_threaded\_set(long jarg1, IndexSplitVectors jarg1\_, boolean jarg2);

public final static native boolean IndexSplitVectors\_threaded\_get(long jarg1, IndexSplitVectors jarg1\_);

public final static native void IndexSplitVectors\_sub\_indexes\_set(long jarg1, IndexSplitVectors jarg1\_, long jarg2);

public final static native long IndexSplitVectors\_sub\_indexes\_get(long jarg1, IndexSplitVectors jarg1\_);

public final static native void IndexSplitVectors\_sum\_d\_set(long jarg1, IndexSplitVectors jarg1\_, long jarg2);

public final static native long IndexSplitVectors\_sum\_d\_get(long jarg1, IndexSplitVectors jarg1\_);

public final static native void IndexSplitVectors\_add\_sub\_index(long jarg1, IndexSplitVectors jarg1\_, long jarg2, Index jarg2\_);

public final static native void IndexSplitVectors\_sync\_with\_sub\_indexes(long jarg1, IndexSplitVectors jarg1\_);

public final static native void IndexSplitVectors\_add(long jarg1, IndexSplitVectors jarg1\_, long jarg2, long jarg3);

public final static native void IndexSplitVectors\_search(long jarg1, IndexSplitVectors jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexSplitVectors\_train(long jarg1, IndexSplitVectors jarg1\_, long jarg2, long jarg3);

public final static native void IndexSplitVectors\_reset(long jarg1, IndexSplitVectors jarg1\_);

public final static native void delete\_IndexSplitVectors(long jarg1);

public final static native void IndexIDMap\_index\_set(long jarg1, IndexIDMap jarg1\_, long jarg2, Index jarg2\_);

public final static native long IndexIDMap\_index\_get(long jarg1, IndexIDMap jarg1\_);

public final static native void IndexIDMap\_own\_fields\_set(long jarg1, IndexIDMap jarg1\_, boolean jarg2);

public final static native boolean IndexIDMap\_own\_fields\_get(long jarg1, IndexIDMap jarg1\_);

public final static native void IndexIDMap\_id\_map\_set(long jarg1, IndexIDMap jarg1\_, long jarg2);

public final static native long IndexIDMap\_id\_map\_get(long jarg1, IndexIDMap jarg1\_);

public final static native long new\_IndexIDMap\_\_SWIG\_0(long jarg1, Index jarg1\_);

public final static native void IndexIDMap\_add\_with\_ids(long jarg1, IndexIDMap jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void IndexIDMap\_add(long jarg1, IndexIDMap jarg1\_, long jarg2, long jarg3);

public final static native void IndexIDMap\_search(long jarg1, IndexIDMap jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexIDMap\_train(long jarg1, IndexIDMap jarg1\_, long jarg2, long jarg3);

public final static native void IndexIDMap\_reset(long jarg1, IndexIDMap jarg1\_);

public final static native long IndexIDMap\_remove\_ids(long jarg1, IndexIDMap jarg1\_, long jarg2, IDSelector jarg2\_);

public final static native void IndexIDMap\_range\_search(long jarg1, IndexIDMap jarg1\_, long jarg2, long jarg3, float jarg4, long jarg5, RangeSearchResult jarg5\_);

public final static native void delete\_IndexIDMap(long jarg1);

public final static native long new\_IndexIDMap\_\_SWIG\_1();

public final static native long new\_IndexShards\_\_SWIG\_0(boolean jarg1, boolean jarg2);

public final static native long new\_IndexShards\_\_SWIG\_1(boolean jarg1);

public final static native long new\_IndexShards\_\_SWIG\_2();

public final static native long new\_IndexShards\_\_SWIG\_3(int jarg1, boolean jarg2, boolean jarg3);

public final static native long new\_IndexShards\_\_SWIG\_4(int jarg1, boolean jarg2);

public final static native long new\_IndexShards\_\_SWIG\_5(int jarg1);

public final static native void IndexShards\_add\_shard(long jarg1, IndexShards jarg1\_, long jarg2, Index jarg2\_);

public final static native void IndexShards\_remove\_shard(long jarg1, IndexShards jarg1\_, long jarg2, Index jarg2\_);

public final static native void IndexShards\_add(long jarg1, IndexShards jarg1\_, long jarg2, long jarg3);

public final static native void IndexShards\_add\_with\_ids(long jarg1, IndexShards jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void IndexShards\_search(long jarg1, IndexShards jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6, LongVector jarg6\_);

public final static native void IndexShards\_train(long jarg1, IndexShards jarg1\_, long jarg2, long jarg3);

public final static native void IndexShards\_successive\_ids\_set(long jarg1, IndexShards jarg1\_, boolean jarg2);

public final static native boolean IndexShards\_successive\_ids\_get(long jarg1, IndexShards jarg1\_);

public final static native void IndexShards\_syncWithSubIndexes(long jarg1, IndexShards jarg1\_);

public final static native void delete\_IndexShards(long jarg1);

public final static native long downcast\_index(long jarg1, Index jarg1\_);

public final static native long downcast\_VectorTransform(long jarg1, VectorTransform jarg1\_);

public final static native long downcast\_IndexBinary(long jarg1, IndexBinary jarg1\_);

public final static native long upcast\_IndexShards(long jarg1, IndexShards jarg1\_);

public final static native void write\_index\_\_SWIG\_0(long jarg1, Index jarg1\_, String jarg2);

public final static native void write\_index\_\_SWIG\_1(long jarg1, Index jarg1\_, long jarg2);

public final static native void write\_index\_\_SWIG\_2(long jarg1, Index jarg1\_, long jarg2);

public final static native void write\_index\_binary\_\_SWIG\_0(long jarg1, IndexBinary jarg1\_, String jarg2);

public final static native void write\_index\_binary\_\_SWIG\_1(long jarg1, IndexBinary jarg1\_, long jarg2);

public final static native void write\_index\_binary\_\_SWIG\_2(long jarg1, IndexBinary jarg1\_, long jarg2);

public final static native int IO\_FLAG\_READ\_ONLY\_get();

public final static native int IO\_FLAG\_ONDISK\_SAME\_DIR\_get();

public final static native int IO\_FLAG\_SKIP\_IVF\_DATA\_get();

public final static native int IO\_FLAG\_MMAP\_get();

public final static native long read\_index\_\_SWIG\_0(String jarg1, int jarg2);

public final static native long read\_index\_\_SWIG\_1(String jarg1);

public final static native long read\_index\_\_SWIG\_2(long jarg1, int jarg2);

public final static native long read\_index\_\_SWIG\_3(long jarg1);

public final static native long read\_index\_\_SWIG\_4(long jarg1, int jarg2);

public final static native long read\_index\_\_SWIG\_5(long jarg1);

public final static native long read\_index\_binary\_\_SWIG\_0(String jarg1, int jarg2);

public final static native long read\_index\_binary\_\_SWIG\_1(String jarg1);

public final static native long read\_index\_binary\_\_SWIG\_2(long jarg1, int jarg2);

public final static native long read\_index\_binary\_\_SWIG\_3(long jarg1);

public final static native long read\_index\_binary\_\_SWIG\_4(long jarg1, int jarg2);

public final static native long read\_index\_binary\_\_SWIG\_5(long jarg1);

public final static native void write\_VectorTransform(long jarg1, VectorTransform jarg1\_, String jarg2);

public final static native long read\_VectorTransform(String jarg1);

public final static native long read\_ProductQuantizer\_\_SWIG\_0(String jarg1);

public final static native long read\_ProductQuantizer\_\_SWIG\_1(long jarg1);

public final static native void write\_ProductQuantizer\_\_SWIG\_0(long jarg1, ProductQuantizer jarg1\_, String jarg2);

public final static native void write\_ProductQuantizer\_\_SWIG\_1(long jarg1, ProductQuantizer jarg1\_, long jarg2);

public final static native void write\_InvertedLists(long jarg1, InvertedLists jarg1\_, long jarg2);

public final static native long read\_InvertedLists\_\_SWIG\_0(long jarg1, int jarg2);

public final static native long read\_InvertedLists\_\_SWIG\_1(long jarg1);

public final static native void AutoTuneCriterion\_nq\_set(long jarg1, AutoTuneCriterion jarg1\_, long jarg2);

public final static native long AutoTuneCriterion\_nq\_get(long jarg1, AutoTuneCriterion jarg1\_);

public final static native void AutoTuneCriterion\_nnn\_set(long jarg1, AutoTuneCriterion jarg1\_, long jarg2);

public final static native long AutoTuneCriterion\_nnn\_get(long jarg1, AutoTuneCriterion jarg1\_);

public final static native void AutoTuneCriterion\_gt\_nnn\_set(long jarg1, AutoTuneCriterion jarg1\_, long jarg2);

public final static native long AutoTuneCriterion\_gt\_nnn\_get(long jarg1, AutoTuneCriterion jarg1\_);

public final static native void AutoTuneCriterion\_gt\_D\_set(long jarg1, AutoTuneCriterion jarg1\_, long jarg2, FloatVector jarg2\_);

public final static native long AutoTuneCriterion\_gt\_D\_get(long jarg1, AutoTuneCriterion jarg1\_);

public final static native void AutoTuneCriterion\_gt\_I\_set(long jarg1, AutoTuneCriterion jarg1\_, long jarg2);

public final static native long AutoTuneCriterion\_gt\_I\_get(long jarg1, AutoTuneCriterion jarg1\_);

public final static native void AutoTuneCriterion\_set\_groundtruth(long jarg1, AutoTuneCriterion jarg1\_, int jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native double AutoTuneCriterion\_evaluate(long jarg1, AutoTuneCriterion jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native void delete\_AutoTuneCriterion(long jarg1);

public final static native void OneRecallAtRCriterion\_R\_set(long jarg1, OneRecallAtRCriterion jarg1\_, long jarg2);

public final static native long OneRecallAtRCriterion\_R\_get(long jarg1, OneRecallAtRCriterion jarg1\_);

public final static native long new\_OneRecallAtRCriterion(long jarg1, long jarg2);

public final static native double OneRecallAtRCriterion\_evaluate(long jarg1, OneRecallAtRCriterion jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native void delete\_OneRecallAtRCriterion(long jarg1);

public final static native void IntersectionCriterion\_R\_set(long jarg1, IntersectionCriterion jarg1\_, long jarg2);

public final static native long IntersectionCriterion\_R\_get(long jarg1, IntersectionCriterion jarg1\_);

public final static native long new\_IntersectionCriterion(long jarg1, long jarg2);

public final static native double IntersectionCriterion\_evaluate(long jarg1, IntersectionCriterion jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native void delete\_IntersectionCriterion(long jarg1);

public final static native void OperatingPoint\_perf\_set(long jarg1, OperatingPoint jarg1\_, double jarg2);

public final static native double OperatingPoint\_perf\_get(long jarg1, OperatingPoint jarg1\_);

public final static native void OperatingPoint\_t\_set(long jarg1, OperatingPoint jarg1\_, double jarg2);

public final static native double OperatingPoint\_t\_get(long jarg1, OperatingPoint jarg1\_);

public final static native void OperatingPoint\_key\_set(long jarg1, OperatingPoint jarg1\_, String jarg2);

public final static native String OperatingPoint\_key\_get(long jarg1, OperatingPoint jarg1\_);

public final static native void OperatingPoint\_cno\_set(long jarg1, OperatingPoint jarg1\_, long jarg2);

public final static native long OperatingPoint\_cno\_get(long jarg1, OperatingPoint jarg1\_);

public final static native long new\_OperatingPoint();

public final static native void delete\_OperatingPoint(long jarg1);

public final static native void OperatingPoints\_all\_pts\_set(long jarg1, OperatingPoints jarg1\_, long jarg2, OperatingPointVector jarg2\_);

public final static native long OperatingPoints\_all\_pts\_get(long jarg1, OperatingPoints jarg1\_);

public final static native void OperatingPoints\_optimal\_pts\_set(long jarg1, OperatingPoints jarg1\_, long jarg2, OperatingPointVector jarg2\_);

public final static native long OperatingPoints\_optimal\_pts\_get(long jarg1, OperatingPoints jarg1\_);

public final static native long new\_OperatingPoints();

public final static native int OperatingPoints\_merge\_with\_\_SWIG\_0(long jarg1, OperatingPoints jarg1\_, long jarg2, OperatingPoints jarg2\_, String jarg3);

public final static native int OperatingPoints\_merge\_with\_\_SWIG\_1(long jarg1, OperatingPoints jarg1\_, long jarg2, OperatingPoints jarg2\_);

public final static native void OperatingPoints\_clear(long jarg1, OperatingPoints jarg1\_);

public final static native boolean OperatingPoints\_add\_\_SWIG\_0(long jarg1, OperatingPoints jarg1\_, double jarg2, double jarg3, String jarg4, long jarg5);

public final static native boolean OperatingPoints\_add\_\_SWIG\_1(long jarg1, OperatingPoints jarg1\_, double jarg2, double jarg3, String jarg4);

public final static native double OperatingPoints\_t\_for\_perf(long jarg1, OperatingPoints jarg1\_, double jarg2);

public final static native void OperatingPoints\_display\_\_SWIG\_0(long jarg1, OperatingPoints jarg1\_, boolean jarg2);

public final static native void OperatingPoints\_display\_\_SWIG\_1(long jarg1, OperatingPoints jarg1\_);

public final static native void OperatingPoints\_all\_to\_gnuplot(long jarg1, OperatingPoints jarg1\_, String jarg2);

public final static native void OperatingPoints\_optimal\_to\_gnuplot(long jarg1, OperatingPoints jarg1\_, String jarg2);

public final static native void delete\_OperatingPoints(long jarg1);

public final static native void ParameterRange\_name\_set(long jarg1, ParameterRange jarg1\_, String jarg2);

public final static native String ParameterRange\_name\_get(long jarg1, ParameterRange jarg1\_);

public final static native void ParameterRange\_values\_set(long jarg1, ParameterRange jarg1\_, long jarg2, DoubleVector jarg2\_);

public final static native long ParameterRange\_values\_get(long jarg1, ParameterRange jarg1\_);

public final static native long new\_ParameterRange();

public final static native void delete\_ParameterRange(long jarg1);

public final static native void ParameterSpace\_parameter\_ranges\_set(long jarg1, ParameterSpace jarg1\_, long jarg2);

public final static native long ParameterSpace\_parameter\_ranges\_get(long jarg1, ParameterSpace jarg1\_);

public final static native void ParameterSpace\_verbose\_set(long jarg1, ParameterSpace jarg1\_, int jarg2);

public final static native int ParameterSpace\_verbose\_get(long jarg1, ParameterSpace jarg1\_);

public final static native void ParameterSpace\_n\_experiments\_set(long jarg1, ParameterSpace jarg1\_, int jarg2);

public final static native int ParameterSpace\_n\_experiments\_get(long jarg1, ParameterSpace jarg1\_);

public final static native void ParameterSpace\_batchsize\_set(long jarg1, ParameterSpace jarg1\_, long jarg2);

public final static native long ParameterSpace\_batchsize\_get(long jarg1, ParameterSpace jarg1\_);

public final static native void ParameterSpace\_thread\_over\_batches\_set(long jarg1, ParameterSpace jarg1\_, boolean jarg2);

public final static native boolean ParameterSpace\_thread\_over\_batches\_get(long jarg1, ParameterSpace jarg1\_);

public final static native void ParameterSpace\_min\_test\_duration\_set(long jarg1, ParameterSpace jarg1\_, double jarg2);

public final static native double ParameterSpace\_min\_test\_duration\_get(long jarg1, ParameterSpace jarg1\_);

public final static native long new\_ParameterSpace();

public final static native long ParameterSpace\_n\_combinations(long jarg1, ParameterSpace jarg1\_);

public final static native boolean ParameterSpace\_combination\_ge(long jarg1, ParameterSpace jarg1\_, long jarg2, long jarg3);

public final static native String ParameterSpace\_combination\_name(long jarg1, ParameterSpace jarg1\_, long jarg2);

public final static native void ParameterSpace\_display(long jarg1, ParameterSpace jarg1\_);

public final static native long ParameterSpace\_add\_range(long jarg1, ParameterSpace jarg1\_, String jarg2);

public final static native void ParameterSpace\_initialize(long jarg1, ParameterSpace jarg1\_, long jarg2, Index jarg2\_);

public final static native void ParameterSpace\_set\_index\_parameters\_\_SWIG\_0(long jarg1, ParameterSpace jarg1\_, long jarg2, Index jarg2\_, long jarg3);

public final static native void ParameterSpace\_set\_index\_parameters\_\_SWIG\_1(long jarg1, ParameterSpace jarg1\_, long jarg2, Index jarg2\_, String jarg3);

public final static native void ParameterSpace\_set\_index\_parameter(long jarg1, ParameterSpace jarg1\_, long jarg2, Index jarg2\_, String jarg3, double jarg4);

public final static native void ParameterSpace\_update\_bounds(long jarg1, ParameterSpace jarg1\_, long jarg2, long jarg3, OperatingPoint jarg3\_, long jarg4, long jarg5);

public final static native void ParameterSpace\_explore(long jarg1, ParameterSpace jarg1\_, long jarg2, Index jarg2\_, long jarg3, long jarg4, long jarg5, AutoTuneCriterion jarg5\_, long jarg6, OperatingPoints jarg6\_);

public final static native void delete\_ParameterSpace(long jarg1);

public final static native long index\_factory\_\_SWIG\_0(int jarg1, String jarg2, int jarg3);

public final static native long index\_factory\_\_SWIG\_1(int jarg1, String jarg2);

public final static native void index\_factory\_verbose\_set(int jarg1);

public final static native int index\_factory\_verbose\_get();

public final static native long index\_binary\_factory(int jarg1, String jarg2);

public final static native void simd\_histogram\_8(long jarg1, int jarg2, long jarg3, int jarg4, long jarg5);

public final static native void simd\_histogram\_16(long jarg1, int jarg2, long jarg3, int jarg4, long jarg5);

public final static native void PartitionStats\_bissect\_cycles\_set(long jarg1, PartitionStats jarg1\_, long jarg2);

public final static native long PartitionStats\_bissect\_cycles\_get(long jarg1, PartitionStats jarg1\_);

public final static native void PartitionStats\_compress\_cycles\_set(long jarg1, PartitionStats jarg1\_, long jarg2);

public final static native long PartitionStats\_compress\_cycles\_get(long jarg1, PartitionStats jarg1\_);

public final static native long new\_PartitionStats();

public final static native void PartitionStats\_reset(long jarg1, PartitionStats jarg1\_);

public final static native void delete\_PartitionStats(long jarg1);

public final static native void partition\_stats\_set(long jarg1, PartitionStats jarg1\_);

public final static native long partition\_stats\_get();

public final static native void float\_minheap\_array\_t\_nh\_set(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2);

public final static native long float\_minheap\_array\_t\_nh\_get(long jarg1, float\_minheap\_array\_t jarg1\_);

public final static native void float\_minheap\_array\_t\_k\_set(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2);

public final static native long float\_minheap\_array\_t\_k\_get(long jarg1, float\_minheap\_array\_t jarg1\_);

public final static native void float\_minheap\_array\_t\_ids\_set(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, LongVector jarg2\_);

public final static native long float\_minheap\_array\_t\_ids\_get(long jarg1, float\_minheap\_array\_t jarg1\_);

public final static native void float\_minheap\_array\_t\_val\_set(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2);

public final static native long float\_minheap\_array\_t\_val\_get(long jarg1, float\_minheap\_array\_t jarg1\_);

public final static native long float\_minheap\_array\_t\_get\_val(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2);

public final static native long float\_minheap\_array\_t\_get\_ids(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2);

public final static native void float\_minheap\_array\_t\_heapify(long jarg1, float\_minheap\_array\_t jarg1\_);

public final static native void float\_minheap\_array\_t\_addn\_\_SWIG\_0(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void float\_minheap\_array\_t\_addn\_\_SWIG\_1(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native void float\_minheap\_array\_t\_addn\_\_SWIG\_2(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void float\_minheap\_array\_t\_addn\_\_SWIG\_3(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3);

public final static native void float\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_0(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6, long jarg7);

public final static native void float\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_1(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6);

public final static native void float\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_2(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void float\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_3(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void float\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_4(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3);

public final static native void float\_minheap\_array\_t\_reorder(long jarg1, float\_minheap\_array\_t jarg1\_);

public final static native void float\_minheap\_array\_t\_per\_line\_extrema(long jarg1, float\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long new\_float\_minheap\_array\_t();

public final static native void delete\_float\_minheap\_array\_t(long jarg1);

public final static native void int\_minheap\_array\_t\_nh\_set(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2);

public final static native long int\_minheap\_array\_t\_nh\_get(long jarg1, int\_minheap\_array\_t jarg1\_);

public final static native void int\_minheap\_array\_t\_k\_set(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2);

public final static native long int\_minheap\_array\_t\_k\_get(long jarg1, int\_minheap\_array\_t jarg1\_);

public final static native void int\_minheap\_array\_t\_ids\_set(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, LongVector jarg2\_);

public final static native long int\_minheap\_array\_t\_ids\_get(long jarg1, int\_minheap\_array\_t jarg1\_);

public final static native void int\_minheap\_array\_t\_val\_set(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2);

public final static native long int\_minheap\_array\_t\_val\_get(long jarg1, int\_minheap\_array\_t jarg1\_);

public final static native long int\_minheap\_array\_t\_get\_val(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2);

public final static native long int\_minheap\_array\_t\_get\_ids(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2);

public final static native void int\_minheap\_array\_t\_heapify(long jarg1, int\_minheap\_array\_t jarg1\_);

public final static native void int\_minheap\_array\_t\_addn\_\_SWIG\_0(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void int\_minheap\_array\_t\_addn\_\_SWIG\_1(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native void int\_minheap\_array\_t\_addn\_\_SWIG\_2(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void int\_minheap\_array\_t\_addn\_\_SWIG\_3(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3);

public final static native void int\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_0(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6, long jarg7);

public final static native void int\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_1(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6);

public final static native void int\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_2(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void int\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_3(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void int\_minheap\_array\_t\_addn\_with\_ids\_\_SWIG\_4(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3);

public final static native void int\_minheap\_array\_t\_reorder(long jarg1, int\_minheap\_array\_t jarg1\_);

public final static native void int\_minheap\_array\_t\_per\_line\_extrema(long jarg1, int\_minheap\_array\_t jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long new\_int\_minheap\_array\_t();

public final static native void delete\_int\_minheap\_array\_t(long jarg1);

public final static native void float\_maxheap\_array\_t\_nh\_set(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native long float\_maxheap\_array\_t\_nh\_get(long jarg1, float\_maxheap\_array\_t jarg1\_);

public final static native void float\_maxheap\_array\_t\_k\_set(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native long float\_maxheap\_array\_t\_k\_get(long jarg1, float\_maxheap\_array\_t jarg1\_);

public final static native void float\_maxheap\_array\_t\_ids\_set(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, LongVector jarg2\_);

public final static native long float\_maxheap\_array\_t\_ids\_get(long jarg1, float\_maxheap\_array\_t jarg1\_);

public final static native void float\_maxheap\_array\_t\_val\_set(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native long float\_maxheap\_array\_t\_val\_get(long jarg1, float\_maxheap\_array\_t jarg1\_);

public final static native long float\_maxheap\_array\_t\_get\_val(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native long float\_maxheap\_array\_t\_get\_ids(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native void float\_maxheap\_array\_t\_heapify(long jarg1, float\_maxheap\_array\_t jarg1\_);

public final static native void float\_maxheap\_array\_t\_addn\_\_SWIG\_0(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void float\_maxheap\_array\_t\_addn\_\_SWIG\_1(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native void float\_maxheap\_array\_t\_addn\_\_SWIG\_2(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void float\_maxheap\_array\_t\_addn\_\_SWIG\_3(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3);

public final static native void float\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_0(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6, long jarg7);

public final static native void float\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_1(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6);

public final static native void float\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_2(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void float\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_3(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void float\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_4(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3);

public final static native void float\_maxheap\_array\_t\_reorder(long jarg1, float\_maxheap\_array\_t jarg1\_);

public final static native void float\_maxheap\_array\_t\_per\_line\_extrema(long jarg1, float\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long new\_float\_maxheap\_array\_t();

public final static native void delete\_float\_maxheap\_array\_t(long jarg1);

public final static native void int\_maxheap\_array\_t\_nh\_set(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native long int\_maxheap\_array\_t\_nh\_get(long jarg1, int\_maxheap\_array\_t jarg1\_);

public final static native void int\_maxheap\_array\_t\_k\_set(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native long int\_maxheap\_array\_t\_k\_get(long jarg1, int\_maxheap\_array\_t jarg1\_);

public final static native void int\_maxheap\_array\_t\_ids\_set(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, LongVector jarg2\_);

public final static native long int\_maxheap\_array\_t\_ids\_get(long jarg1, int\_maxheap\_array\_t jarg1\_);

public final static native void int\_maxheap\_array\_t\_val\_set(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native long int\_maxheap\_array\_t\_val\_get(long jarg1, int\_maxheap\_array\_t jarg1\_);

public final static native long int\_maxheap\_array\_t\_get\_val(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native long int\_maxheap\_array\_t\_get\_ids(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2);

public final static native void int\_maxheap\_array\_t\_heapify(long jarg1, int\_maxheap\_array\_t jarg1\_);

public final static native void int\_maxheap\_array\_t\_addn\_\_SWIG\_0(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void int\_maxheap\_array\_t\_addn\_\_SWIG\_1(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, long jarg5);

public final static native void int\_maxheap\_array\_t\_addn\_\_SWIG\_2(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native void int\_maxheap\_array\_t\_addn\_\_SWIG\_3(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3);

public final static native void int\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_0(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6, long jarg7);

public final static native void int\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_1(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5, long jarg6);

public final static native void int\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_2(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void int\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_3(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_);

public final static native void int\_maxheap\_array\_t\_addn\_with\_ids\_\_SWIG\_4(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3);

public final static native void int\_maxheap\_array\_t\_reorder(long jarg1, int\_maxheap\_array\_t jarg1\_);

public final static native void int\_maxheap\_array\_t\_per\_line\_extrema(long jarg1, int\_maxheap\_array\_t jarg1\_, long jarg2, long jarg3, LongVector jarg3\_);

public final static native long new\_int\_maxheap\_array\_t();

public final static native void delete\_int\_maxheap\_array\_t(long jarg1);

public final static native float CMin\_float\_partition\_fuzzy(long jarg1, long jarg2, LongVector jarg2\_, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native float CMax\_float\_partition\_fuzzy(long jarg1, long jarg2, LongVector jarg2\_, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void AlignedTableUint8\_tab\_set(long jarg1, AlignedTableUint8 jarg1\_, long jarg2);

public final static native long AlignedTableUint8\_tab\_get(long jarg1, AlignedTableUint8 jarg1\_);

public final static native void AlignedTableUint8\_numel\_set(long jarg1, AlignedTableUint8 jarg1\_, long jarg2);

public final static native long AlignedTableUint8\_numel\_get(long jarg1, AlignedTableUint8 jarg1\_);

public final static native long AlignedTableUint8\_round\_capacity(long jarg1);

public final static native long new\_AlignedTableUint8\_\_SWIG\_0();

public final static native long new\_AlignedTableUint8\_\_SWIG\_1(long jarg1);

public final static native long AlignedTableUint8\_itemsize(long jarg1, AlignedTableUint8 jarg1\_);

public final static native void AlignedTableUint8\_resize(long jarg1, AlignedTableUint8 jarg1\_, long jarg2);

public final static native void AlignedTableUint8\_clear(long jarg1, AlignedTableUint8 jarg1\_);

public final static native long AlignedTableUint8\_size(long jarg1, AlignedTableUint8 jarg1\_);

public final static native long AlignedTableUint8\_nbytes(long jarg1, AlignedTableUint8 jarg1\_);

public final static native long AlignedTableUint8\_get\_\_SWIG\_0(long jarg1, AlignedTableUint8 jarg1\_);

public final static native long AlignedTableUint8\_data\_\_SWIG\_0(long jarg1, AlignedTableUint8 jarg1\_);

public final static native void delete\_AlignedTableUint8(long jarg1);

public final static native void AlignedTableUint16\_tab\_set(long jarg1, AlignedTableUint16 jarg1\_, long jarg2);

public final static native long AlignedTableUint16\_tab\_get(long jarg1, AlignedTableUint16 jarg1\_);

public final static native void AlignedTableUint16\_numel\_set(long jarg1, AlignedTableUint16 jarg1\_, long jarg2);

public final static native long AlignedTableUint16\_numel\_get(long jarg1, AlignedTableUint16 jarg1\_);

public final static native long AlignedTableUint16\_round\_capacity(long jarg1);

public final static native long new\_AlignedTableUint16\_\_SWIG\_0();

public final static native long new\_AlignedTableUint16\_\_SWIG\_1(long jarg1);

public final static native long AlignedTableUint16\_itemsize(long jarg1, AlignedTableUint16 jarg1\_);

public final static native void AlignedTableUint16\_resize(long jarg1, AlignedTableUint16 jarg1\_, long jarg2);

public final static native void AlignedTableUint16\_clear(long jarg1, AlignedTableUint16 jarg1\_);

public final static native long AlignedTableUint16\_size(long jarg1, AlignedTableUint16 jarg1\_);

public final static native long AlignedTableUint16\_nbytes(long jarg1, AlignedTableUint16 jarg1\_);

public final static native long AlignedTableUint16\_get\_\_SWIG\_0(long jarg1, AlignedTableUint16 jarg1\_);

public final static native long AlignedTableUint16\_data\_\_SWIG\_0(long jarg1, AlignedTableUint16 jarg1\_);

public final static native void delete\_AlignedTableUint16(long jarg1);

public final static native void AlignedTableFloat32\_tab\_set(long jarg1, AlignedTableFloat32 jarg1\_, long jarg2);

public final static native long AlignedTableFloat32\_tab\_get(long jarg1, AlignedTableFloat32 jarg1\_);

public final static native void AlignedTableFloat32\_numel\_set(long jarg1, AlignedTableFloat32 jarg1\_, long jarg2);

public final static native long AlignedTableFloat32\_numel\_get(long jarg1, AlignedTableFloat32 jarg1\_);

public final static native long AlignedTableFloat32\_round\_capacity(long jarg1);

public final static native long new\_AlignedTableFloat32\_\_SWIG\_0();

public final static native long new\_AlignedTableFloat32\_\_SWIG\_1(long jarg1);

public final static native long AlignedTableFloat32\_itemsize(long jarg1, AlignedTableFloat32 jarg1\_);

public final static native void AlignedTableFloat32\_resize(long jarg1, AlignedTableFloat32 jarg1\_, long jarg2);

public final static native void AlignedTableFloat32\_clear(long jarg1, AlignedTableFloat32 jarg1\_);

public final static native long AlignedTableFloat32\_size(long jarg1, AlignedTableFloat32 jarg1\_);

public final static native long AlignedTableFloat32\_nbytes(long jarg1, AlignedTableFloat32 jarg1\_);

public final static native long AlignedTableFloat32\_get\_\_SWIG\_0(long jarg1, AlignedTableFloat32 jarg1\_);

public final static native long AlignedTableFloat32\_data\_\_SWIG\_0(long jarg1, AlignedTableFloat32 jarg1\_);

public final static native void delete\_AlignedTableFloat32(long jarg1);

public final static native long CMax\_uint16\_partition\_fuzzy\_\_SWIG\_0(long jarg1, long jarg2, LongVector jarg2\_, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native long CMin\_uint16\_partition\_fuzzy\_\_SWIG\_0(long jarg1, long jarg2, LongVector jarg2\_, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native long CMax\_uint16\_partition\_fuzzy\_\_SWIG\_1(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native long CMin\_uint16\_partition\_fuzzy\_\_SWIG\_1(long jarg1, long jarg2, long jarg3, long jarg4, long jarg5, long jarg6);

public final static native void omp\_set\_num\_threads(int jarg1);

public final static native int omp\_get\_max\_threads();

public final static native long memcpy(long jarg1, long jarg2, long jarg3);

public final static native long cast\_integer\_to\_float\_ptr(int jarg1);

public final static native long cast\_integer\_to\_long\_ptr(int jarg1);

public final static native long cast\_integer\_to\_int\_ptr(int jarg1);

public final static native void RangeSearchResult\_nq\_set(long jarg1, RangeSearchResult jarg1\_, long jarg2);

public final static native long RangeSearchResult\_nq\_get(long jarg1, RangeSearchResult jarg1\_);

public final static native void RangeSearchResult\_lims\_set(long jarg1, RangeSearchResult jarg1\_, long jarg2);

public final static native long RangeSearchResult\_lims\_get(long jarg1, RangeSearchResult jarg1\_);

public final static native void RangeSearchResult\_labels\_set(long jarg1, RangeSearchResult jarg1\_, long jarg2, LongVector jarg2\_);

public final static native long RangeSearchResult\_labels\_get(long jarg1, RangeSearchResult jarg1\_);

public final static native void RangeSearchResult\_distances\_set(long jarg1, RangeSearchResult jarg1\_, long jarg2);

public final static native long RangeSearchResult\_distances\_get(long jarg1, RangeSearchResult jarg1\_);

public final static native void RangeSearchResult\_buffer\_size\_set(long jarg1, RangeSearchResult jarg1\_, long jarg2);

public final static native long RangeSearchResult\_buffer\_size\_get(long jarg1, RangeSearchResult jarg1\_);

public final static native void RangeSearchResult\_do\_allocation(long jarg1, RangeSearchResult jarg1\_);

public final static native void delete\_RangeSearchResult(long jarg1);

public final static native boolean IDSelector\_is\_member(long jarg1, IDSelector jarg1\_, long jarg2);

public final static native void delete\_IDSelector(long jarg1);

public final static native void IDSelectorRange\_imin\_set(long jarg1, IDSelectorRange jarg1\_, long jarg2);

public final static native long IDSelectorRange\_imin\_get(long jarg1, IDSelectorRange jarg1\_);

public final static native void IDSelectorRange\_imax\_set(long jarg1, IDSelectorRange jarg1\_, long jarg2);

public final static native long IDSelectorRange\_imax\_get(long jarg1, IDSelectorRange jarg1\_);

public final static native long new\_IDSelectorRange(long jarg1, long jarg2);

public final static native boolean IDSelectorRange\_is\_member(long jarg1, IDSelectorRange jarg1\_, long jarg2);

public final static native void delete\_IDSelectorRange(long jarg1);

public final static native void IDSelectorArray\_n\_set(long jarg1, IDSelectorArray jarg1\_, long jarg2);

public final static native long IDSelectorArray\_n\_get(long jarg1, IDSelectorArray jarg1\_);

public final static native void IDSelectorArray\_ids\_set(long jarg1, IDSelectorArray jarg1\_, long jarg2, LongVector jarg2\_);

public final static native long IDSelectorArray\_ids\_get(long jarg1, IDSelectorArray jarg1\_);

public final static native long new\_IDSelectorArray(long jarg1, long jarg2, LongVector jarg2\_);

public final static native boolean IDSelectorArray\_is\_member(long jarg1, IDSelectorArray jarg1\_, long jarg2);

public final static native void delete\_IDSelectorArray(long jarg1);

public final static native void IDSelectorBatch\_nbits\_set(long jarg1, IDSelectorBatch jarg1\_, int jarg2);

public final static native int IDSelectorBatch\_nbits\_get(long jarg1, IDSelectorBatch jarg1\_);

public final static native void IDSelectorBatch\_mask\_set(long jarg1, IDSelectorBatch jarg1\_, long jarg2);

public final static native long IDSelectorBatch\_mask\_get(long jarg1, IDSelectorBatch jarg1\_);

public final static native long new\_IDSelectorBatch(long jarg1, long jarg2, LongVector jarg2\_);

public final static native boolean IDSelectorBatch\_is\_member(long jarg1, IDSelectorBatch jarg1\_, long jarg2);

public final static native void delete\_IDSelectorBatch(long jarg1);

public final static native void BufferList\_buffer\_size\_set(long jarg1, BufferList jarg1\_, long jarg2);

public final static native long BufferList\_buffer\_size\_get(long jarg1, BufferList jarg1\_);

public final static native void BufferList\_buffers\_set(long jarg1, BufferList jarg1\_, long jarg2);

public final static native long BufferList\_buffers\_get(long jarg1, BufferList jarg1\_);

public final static native void BufferList\_wp\_set(long jarg1, BufferList jarg1\_, long jarg2);

public final static native long BufferList\_wp\_get(long jarg1, BufferList jarg1\_);

public final static native long new\_BufferList(long jarg1);

public final static native void delete\_BufferList(long jarg1);

public final static native void BufferList\_append\_buffer(long jarg1, BufferList jarg1\_);

public final static native void BufferList\_add(long jarg1, BufferList jarg1\_, long jarg2, float jarg3);

public final static native void BufferList\_copy\_range(long jarg1, BufferList jarg1\_, long jarg2, long jarg3, long jarg4, LongVector jarg4\_, long jarg5);

public final static native void RangeQueryResult\_qno\_set(long jarg1, RangeQueryResult jarg1\_, long jarg2);

public final static native long RangeQueryResult\_qno\_get(long jarg1, RangeQueryResult jarg1\_);

public final static native void RangeQueryResult\_nres\_set(long jarg1, RangeQueryResult jarg1\_, long jarg2);

public final static native long RangeQueryResult\_nres\_get(long jarg1, RangeQueryResult jarg1\_);

public final static native void RangeQueryResult\_pres\_set(long jarg1, RangeQueryResult jarg1\_, long jarg2, RangeSearchPartialResult jarg2\_);

public final static native long RangeQueryResult\_pres\_get(long jarg1, RangeQueryResult jarg1\_);

public final static native void RangeQueryResult\_add(long jarg1, RangeQueryResult jarg1\_, float jarg2, long jarg3);

public final static native long new\_RangeQueryResult();

public final static native void delete\_RangeQueryResult(long jarg1);

public final static native void RangeSearchPartialResult\_res\_set(long jarg1, RangeSearchPartialResult jarg1\_, long jarg2, RangeSearchResult jarg2\_);

public final static native long RangeSearchPartialResult\_res\_get(long jarg1, RangeSearchPartialResult jarg1\_);

public final static native void RangeSearchPartialResult\_queries\_set(long jarg1, RangeSearchPartialResult jarg1\_, long jarg2);

public final static native long RangeSearchPartialResult\_queries\_get(long jarg1, RangeSearchPartialResult jarg1\_);

public final static native long RangeSearchPartialResult\_new\_result(long jarg1, RangeSearchPartialResult jarg1\_, long jarg2);

public final static native void RangeSearchPartialResult\_set\_lims(long jarg1, RangeSearchPartialResult jarg1\_);

public final static native void RangeSearchPartialResult\_copy\_result\_\_SWIG\_0(long jarg1, RangeSearchPartialResult jarg1\_, boolean jarg2);

public final static native void RangeSearchPartialResult\_copy\_result\_\_SWIG\_1(long jarg1, RangeSearchPartialResult jarg1\_);

public final static native void RangeSearchPartialResult\_merge\_\_SWIG\_0(long jarg1, boolean jarg2);

public final static native void RangeSearchPartialResult\_merge\_\_SWIG\_1(long jarg1);

public final static native void delete\_RangeSearchPartialResult(long jarg1);

public final static native void DistanceComputer\_set\_query(long jarg1, DistanceComputer jarg1\_, long jarg2);

public final static native float DistanceComputer\_symmetric\_dis(long jarg1, DistanceComputer jarg1\_, long jarg2, long jarg3);

public final static native void delete\_DistanceComputer(long jarg1);

public final static native boolean InterruptCallback\_want\_interrupt(long jarg1, InterruptCallback jarg1\_);

public final static native void delete\_InterruptCallback(long jarg1);

public final static native void InterruptCallback\_clear\_instance();

public final static native void InterruptCallback\_check();

public final static native boolean InterruptCallback\_is\_interrupted();

public final static native long InterruptCallback\_get\_period\_hint(long jarg1);

public final static native void VisitedTable\_visited\_set(long jarg1, VisitedTable jarg1\_, long jarg2, ByteVector jarg2\_);

public final static native long VisitedTable\_visited\_get(long jarg1, VisitedTable jarg1\_);

public final static native void VisitedTable\_visno\_set(long jarg1, VisitedTable jarg1\_, int jarg2);

public final static native int VisitedTable\_visno\_get(long jarg1, VisitedTable jarg1\_);

public final static native long new\_VisitedTable(int jarg1);

public final static native void VisitedTable\_set(long jarg1, VisitedTable jarg1\_, int jarg2);

public final static native boolean VisitedTable\_get(long jarg1, VisitedTable jarg1\_, int jarg2);

public final static native void VisitedTable\_advance(long jarg1, VisitedTable jarg1\_);

public final static native void delete\_VisitedTable(long jarg1);

public final static native void ignore\_SIGTTIN();

public final static native void MapLong2Long\_map\_set(long jarg1, MapLong2Long jarg1\_, long jarg2);

public final static native long MapLong2Long\_map\_get(long jarg1, MapLong2Long jarg1\_);

public final static native void MapLong2Long\_add(long jarg1, MapLong2Long jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native int MapLong2Long\_search(long jarg1, MapLong2Long jarg1\_, int jarg2);

public final static native void MapLong2Long\_search\_multiple(long jarg1, MapLong2Long jarg1\_, long jarg2, long jarg3, long jarg4);

public final static native long new\_MapLong2Long();

public final static native void delete\_MapLong2Long(long jarg1);

public final static native long Clustering\_SWIGUpcast(long jarg1);

public final static native long Clustering1D\_SWIGUpcast(long jarg1);

public final static native long ProgressiveDimClusteringParameters\_SWIGUpcast(long jarg1);

public final static native long ProgressiveDimClustering\_SWIGUpcast(long jarg1);

public final static native long LinearTransform\_SWIGUpcast(long jarg1);

public final static native long RandomRotationMatrix\_SWIGUpcast(long jarg1);

public final static native long PCAMatrix\_SWIGUpcast(long jarg1);

public final static native long ITQMatrix\_SWIGUpcast(long jarg1);

public final static native long ITQTransform\_SWIGUpcast(long jarg1);

public final static native long OPQMatrix\_SWIGUpcast(long jarg1);

public final static native long RemapDimensionsTransform\_SWIGUpcast(long jarg1);

public final static native long NormalizationTransform\_SWIGUpcast(long jarg1);

public final static native long CenteringTransform\_SWIGUpcast(long jarg1);

public final static native long IndexFlatCodes\_SWIGUpcast(long jarg1);

public final static native long IndexFlat\_SWIGUpcast(long jarg1);

public final static native long IndexFlatIP\_SWIGUpcast(long jarg1);

public final static native long IndexFlatL2\_SWIGUpcast(long jarg1);

public final static native long IndexFlat1D\_SWIGUpcast(long jarg1);

public final static native long IndexLSH\_SWIGUpcast(long jarg1);

public final static native long ReproduceDistancesObjective\_SWIGUpcast(long jarg1);

public final static native long SimulatedAnnealingOptimizer\_SWIGUpcast(long jarg1);

public final static native long PolysemousTraining\_SWIGUpcast(long jarg1);

public final static native long IndexPQ\_SWIGUpcast(long jarg1);

public final static native long MultiIndexQuantizer\_SWIGUpcast(long jarg1);

public final static native long MultiIndexQuantizer2\_SWIGUpcast(long jarg1);

public final static native long ArrayInvertedLists\_SWIGUpcast(long jarg1);

public final static native long ReadOnlyInvertedLists\_SWIGUpcast(long jarg1);

public final static native long HStackInvertedLists\_SWIGUpcast(long jarg1);

public final static native long SliceInvertedLists\_SWIGUpcast(long jarg1);

public final static native long VStackInvertedLists\_SWIGUpcast(long jarg1);

public final static native long MaskedInvertedLists\_SWIGUpcast(long jarg1);

public final static native long StopWordsInvertedLists\_SWIGUpcast(long jarg1);

public final static native long IndexIVF\_SWIGUpcast(long jarg1);

public final static native long IndexScalarQuantizer\_SWIGUpcast(long jarg1);

public final static native long IndexIVFScalarQuantizer\_SWIGUpcast(long jarg1);

public final static native long IndexHNSW\_SWIGUpcast(long jarg1);

public final static native long IndexHNSWFlat\_SWIGUpcast(long jarg1);

public final static native long IndexHNSWPQ\_SWIGUpcast(long jarg1);

public final static native long IndexHNSWSQ\_SWIGUpcast(long jarg1);

public final static native long IndexHNSW2Level\_SWIGUpcast(long jarg1);

public final static native long IndexIVFFlat\_SWIGUpcast(long jarg1);

public final static native long IndexIVFFlatDedup\_SWIGUpcast(long jarg1);

public final static native long OnDiskInvertedLists\_SWIGUpcast(long jarg1);

public final static native long IVFPQSearchParameters\_SWIGUpcast(long jarg1);

public final static native long IndexIVFPQ\_SWIGUpcast(long jarg1);

public final static native long Index2Layer\_SWIGUpcast(long jarg1);

public final static native long IndexBinaryFlat\_SWIGUpcast(long jarg1);

public final static native long IndexBinaryIVF\_SWIGUpcast(long jarg1);

public final static native long IndexBinaryFromFloat\_SWIGUpcast(long jarg1);

public final static native long IndexBinaryHNSW\_SWIGUpcast(long jarg1);

public final static native long IndexRefine\_SWIGUpcast(long jarg1);

public final static native long IndexRefineFlat\_SWIGUpcast(long jarg1);

public final static native long IndexSplitVectors\_SWIGUpcast(long jarg1);

public final static native long IndexIDMap\_SWIGUpcast(long jarg1);

public final static native long OneRecallAtRCriterion\_SWIGUpcast(long jarg1);

public final static native long IntersectionCriterion\_SWIGUpcast(long jarg1);

public final static native long IDSelectorRange\_SWIGUpcast(long jarg1);

public final static native long IDSelectorArray\_SWIGUpcast(long jarg1);

public final static native long IDSelectorBatch\_SWIGUpcast(long jarg1);

public final static native long RangeSearchPartialResult\_SWIGUpcast(long jarg1);

}