**אביב2015—מועד-א- פתרון**

**שאלה 1**

**חלק א'**

סעיף 1

typedef struct Point\_\* PPoint;

/\*Interface functions\*/

PPoint PointCreate(int n);

void PointDestroy(PPoint);

int PointnGetLength(PPoint);

Result PointAddCoordinate(PPoint, float);

float PointGetFirstCoordinate(PPoint);

float PointGetNextCoordinate(PPoint);

סעיף 2

typedef struct Point\_

{

int n;

int size; // Must exist to check correctness when adding coordinates

PList coordinates;

} Point;

סעיף 3

float\* cloneCoordinate(float\* coordinate)

{

float\* newCoordinate = (float\*)malloc( sizeof(float) );

\*newCoordinate = \*coordinate;

return newCoordinate;

}

void deleteCoordinate(float\* coordinate)

{

free(coordinate);

}

PPoint PointCreate(int n)

{

PPoint pPoint = (PPoint)malloc( sizeof(Point) );

pPoint->n = n;

pPoint->size = 0;

pPoint->coordinates = ListCreate( cloneCoordinate, deleteCoordinate );

return pPoint;

}

**חלק ב'**

סעיף 1

typedef struct Cluster\_\* PCluster;

/\*Interface functions\*/

PCluster ClusterCreate(int n);

void ClusterDestroy(PCluster);

Result ClusterAddPoint(PCluster, PPoint);

float ClusterGetMinDistance(PCluster, PPoint);

סעיף 2

typedef struct Cluster\_

{

int n;

PList Points;

} Cluster;

**חלק ג'**

סעיף 1

float calcDist( PPoint p1, PPoint p2 )

{

int i;

float dist = 0.f;

float f1 = PointGetFirstCoordinate( p1 );

float f2 = PointGetFirstCoordinate( p2 );

int n = PointnGetLength( p1 );

for( i = 0; i < n; ++i )

{

dist += (f1 - f2) \* (f1 - f2);

f1 = PointGetNextCoordinate( p1 );

f2 = PointGetNextCoordinate( p2 );

}

return sqrt( dist );

}

סעיף 2

float ClusterGetMinDistance(PCluster pCluster, PPoint pPoint)

{

float minDist = 1000.f;

int n = PointnGetLength( pPoint );

PPoint pCurrentPoint = (PPoint)ListGetFirst( pCluster->Points );

while( NULL != pCurrentPoint )

{

float dist = calcDist( pPoint, pCurrentPoint );

if( dist < minDist ) minDist = dist;

pCurrentPoint = (PPoint)ListGetNext( pCluster->Points );

}

return minDist;

**שאלה 2**

**סעיף א'**

void push\_back(T el);

unsigned int size() const;

Vec operator+(const Vec& rhs) const;

T& operator[](unsigned int ind);

template <class T>

Vec<T> Vec<T>::operator+(const Vec<T>& rhs) const {

if (size() != rhs.size())

throw "vector dimensions must agree";

Vec res(\*this);

list<T>::iterator it\_res = res.vals.begin();

list<T>::const\_iterator it\_rhs = rhs.vals.begin();

while ( it\_res != res.vals.end() )

{

\*it\_res = \*it\_res + \*it\_rhs;

++it\_res;

++it\_rhs;

}

return res;

}

**סעיף ב'**

template <class T>

Vec<T> Vec<T>::operator,(const Vec<T>& rhs) const {

Vec<T> res(\*this);

res.vals.insert(res.vals.end(), rhs.vals.begin(), rhs.vals.end());

return res;

}

template <class T>

Vec<T> Vec<T>::operator,(const T& val) const {

Vec<T> res(\*this);

res.push\_back(val);

return res;

}

**סעיף ג'**

template <class T>

Vec<T> Vec<T>::operator[](Vec<unsigned int>& ind) {

Vec<T> res;

for (unsigned int i = 0; i < ind.size(); ++i)

{

if ( ind[i] < size() )

res.push\_back( (\*this)[ ind[i] ] );

else

throw "index exeeds vector dimensions";

}

return res;

}

**סעיף ד' (לא כלול בשאלה)**

template <class T>

class norm\_comp {

public:

bool operator()(const Vec<T>& lhs, const Vec<T>& rhs) const {

T lhs\_norm(0);

T rhs\_norm(0);

for (unsigned int i = 0; i < lhs.size(); ++i)

lhs\_norm = lhs\_norm + lhs[i]\*lhs[i];

for (unsigned int i = 0; i < rhs.size(); ++i)

rhs\_norm = rhs\_norm + rhs[i]\*rhs[i];

return lhs\_norm < rhs\_norm;

}

};**שאלה 3 – פתרון:**

stage 1:

X::X()

stage 2:

X::X()

X::X()

Y::Y()

X::X()

Z::Z()

stage 3:

X::X()

X::X(X&)

Y::Y(X)

X::X(X&)

Z::Z(X)

stage 4:

X::X()

X::X(X&)

Y::Y(X)

stage 5:

X::f()

Y::f()

X::X()

X::X(X&)

Caught X

X::~X()

X::~X()

stage 6:

X::~X()

stage 7:

X::g(int)

Y::g(int)

end of the question.

**שאלה 4 - BASH, שאלות הבנה** (20 נק')

**חלק א': BASH (10 נק')**

1. פתרון:

find\_string:

grep $2 $1 | head -$3 > output1.txt

1. פתרון:

print\_time:

cat `grep .log $1 |head -$2|tail -$3` | calc\_time | sort -n -r -k2

calc\_time:

while read line; do

line\_split=( $line )

(( total\_time=${line\_split[2]}-${line\_split[1]} ))

echo ${line\_split[0]} $total\_time

done

**חלק ב': שאלות הבנה (10 נק')**

1. פתרון:
   1. אם היה מועבר by-value הייתה קריאה נוספת לקונסטרקטור העתקה והיינו נכנסים ללולאה אינסופית
   2. כדי שיהיה אפשר להעביר אובייקטים שהם const
2. פתרון:
   1. פונקציה ידידה – יכולה לגשת למשתני private של A
   2. כי מימוש כפונקציה חברה יהיה בתוך מחלקת ostream שהיא לא בשליטתנו.
3. פתרון:
   1. אופציות :
      1. method() עלול לזרוק חריגה ונאבד את המצביע לאוביקט A שנוצר (זליגת זכרון)
      2. אחרי  delete pa2  אז pa1  נשאר  dangling   ואם ניגש איתו למתודה כלשהי התוכנית תיפול.
   2. אופציות:
      * 1. להשתמש בsmart pointer אשר משחרר זכרון לאובייקט המוצבע כאשר הוא בעצמו נמחק .
        2. לעטוף את הקוד בבלוקים של try ושל catch . בcatch לדאוג לשחרור הזכרון ולזרוק מחדש( (rethrow (עבור בעיה i).
        3. פוינטר חכם שחופף את operator =