הטכניון - מכון טכנולוגי לישראל **044101 מבוא למערכות תוכנה**
הפקולטה להנדסת חשמל **22.2.11**

מבחן סוף סמסטר
מועד א'

משך הבחינה: שלוש שעות

נא לרשום את השם ומספר תעודת הזהות במקום המיועד במחברת הבחינה.

עליכם לענות על כל השאלות **אך ורק במחברת הבחינה**.

יש להגיש רק את מחברת הבחינה (אני אינני מחברת הבחינה) !

במבחן זה 10 עמודים (לא כולל עמוד זה) ו-3 שאלות.

מומלץ לקרוא כל שאלה בעיון רב לפני שניגשים לפתור אותה.

השאלות בבחינה מסודרות לפי נושאי הלימוד ולוא דווקא לפי רמת הקושי !

כל חומר עזר מותר.

בהצלחה!!!

שאלה 1 - ADT (30 נק')

נתון ADT הממש קבוצת ערכים. מספר הערכים המקסימאלי שהקבוצה יכולה להכיל נתון עם אתחולה (maxNum).
זיהוי חד-חד ערכי לערכים נתון ע"י אינדקס שלהם, והוא בתחום 0..MaxNum

```
========= set.h =============
#ifndef __SET_H_
#define __SET_H_

typedef enum {SUCCESS, FAILURE} result;
//element used inside the set
typedef void* SetElementP;
//pointer to the set
typedef struct SetStructure* SetStructureP;
//function to get the element's key
typedef int (* GetKeyFuncP) (SetElementP elem);
//function to delete an element
typedef void (* DeleteElementFuncP) (SetElementP elem);

//initializes a new set
SetStructureP createSet(int maxElement, GetKeyFuncP keyFunc, DeleteElementFuncP deleteFunc);
//destroys the set, and all it's elements
void destroySet(SetStructureP set);
//adds an element to the set
result addElement(SetStructureP set, SetElementP element);
//gets the element from the set (doesn't remove it)
SetElementP getElement(SetStructureP set, int key);
//number of elements inside the set
int getElementCount(SetStructureP set);
#endif

============= set.c ==========================
#include <stdlib.h>
#include "set.h"

typedef struct SetStructure{
	SetElementP* elements;
	int numOfElements;
	GetKeyFuncP getKey;
	DeleteElementFuncP deleteElement;
}SetStructure;

SetElementP getElement(SetStructureP set, int key){
	if (!set)
		return NULL;
	return set->elements[key];
}
```

```
SetStructureP createSet(int maxElement, GetKeyFuncP keyFunc, DeleteElementFuncP deleteFunc){
	int i;
	SetElementP* elements = malloc(sizeof(SetElementP) * maxElement);
	SetStructureP newStruct;
	if (! elements)  return NULL;
	newStruct = malloc(sizeof(SetStructure));
	if (! newStruct) {
		free(elements);
		return NULL;
	}
	for (i=0; i<maxElement; i++) {
		elements[i] = NULL;
	}
	newStruct->elements = elements;
	newStruct->numOfElements = maxElement;
	newStruct->getKey = keyFunc;
	newStruct->deleteElement = deleteFunc;
	return newStruct;
}

result addElement(SetStructureP set, SetElementP element){
	int elKey;
	if (!set || !element)
		return FAILURE;
	elKey = set->getKey(element);
	if (set->elements[elKey])
		return FAILURE;
	set->elements[elKey] = element;
	return SUCCESS;
}

int getElementCount(SetStructureP set){
	int i, count=0;
	if (!set)
		return -1;
	for (i=0;i<set->numOfElements; i++)
		if (set->elements[i]) count++;
	return count;
}

void destroySet(SetStructureP set){
	int i;
	if (! set)
		return;
	for (i=0; i<set->numOfElements; i++)
		set->deleteElement(set->elements[i]);
	free(set->elements);
	free(set);
```

```
}
```

ה- ADT הוצע לחברת בר-קמצא בכדי לאפשר מעקב אחר הגעת עובדיה באיחור. החברה סיפקה את האלמנט שהיא מעוניינת לשמור עם פונקציה שיוצרת אלמנט כזה:

```
typedef struct DayLate {
	int day;		//day in the year the employee was late, 1-365
	int minutes;	//how many minutes the employee was late on that day
	char* excuse;	//employee's excuse for being late
} DayLate;

DayLate* createLate(int day, int minutes, char* excuse){
	DayLate* res = malloc(sizeof(DayLate));
	int excuseLen = strlen(excuse);
	if (! res) return NULL;
	res->excuse= malloc(excuseLen +1);
	if (! excuse) {
		free(res);
		return NULL;
	}
	strcpy(res->excuse, excuse);
	res->day = day;
	res->minutes = minutes;
	return res;
}
```

<u>סעיף א' (12 נקודות)</u>

כתוב דוגמא לחברת בר-קמצא כיצד עליה להשתמש ב-ADT. בדוגמא זו:

- צור משתנה בשם dannyLateSet שיחזיק את ימי האיחור של דני
- צור משתנה בשם yossyLateSet שיחזיק את ימי האיחור של יוסי
- עדכן את הנתונים הבאים בהתאם:
  - דורון איחור בימים ה- 35 וה- 67 של השנה, ב- 5 דקות בכל פעם, בתירוץ "bus was late'
  - יוסי איחר ביום ה- 186 של השנה, ב- 150 דקות, בתירוץ "I thought it was Saturday"
  - מותר להניח, לצורך דוגמא זו בלבד, שאין התנגשויות ב- set (return FAILURE השני בפונקציה addElement לא יקרא)
- הדפס בהודעה כמה ימי איחור היו לדני ויוסי.
- שחרר את כל הזיכרון שהוקצה דינאמית.

עליך לכתוב את כל פונקציות העזר הנדרשות לתוכנה זו.

<u>סעיף ב' (18 נקודות)</u>

חברת בר קמצא עובדת עם ה-ADT המוצע מספר שנים, ופתאום הבינה שהמימוש שלו אינו יעיל. עפ"י מנהל החברה:

- עובדים מגיעים באיחור לעבודה לעיתים נדירות מאוד, אבל הבדיקה כמה ימים עובד איחר מבוצעת הרבה מאוד פעמים.
- עובד שאיחר מעל 9 ימים בשנה, מכל סיבה שהיא, מפוטר מיידית. עובד שלא פוטר לאחר 10 ימי איחור הוא כנראה קרוב משפחה ולא יפוטר בשום מקרה (האיחורים הנוספים שלו אינם חשובים). לכן אין צורך לשמור יותר מ-10 ימי איחור לעובד.

נדרש לשנות את המימוש הפנימי של מבנה הנתונים. על המימוש החדש:

- לא לדרוש שום שינוי בקוד של חברת בר-קמצא שעובדת עם מבנה הנתונים הישן (בפרט, על הפונקציות לערוך בדיקות תקינות שקולות לאלו של הגרסאות הקודמות שלהן)
- להקצות מקום בזכרון ל- 10 איברים בלבד
- לממש בצורה יעילה ככך הניתן את הקריאה ל getSetSize
- אין דרישות יעילות לגבי שאר הפונקציות

נתון המימוש החדש לפונקציה createSet:

```
#define MAX_ELEMENTS 10

SetStructureP createSet(int maxElement, GetKeyFuncP keyFunc, DeleteElementFuncP deleteFunc){
	SetStructureP set = createSet_old(MAX_ELEMENTS, keyFunc, deleteFunc);
	if (set) {
		set->numOfElements=0;
	}
	return set;
}
```

הפונקציה createSet_old, מכילה את המימוש הישן ל- createSet (שמוצג בעמודים הקודמים).

כתוב את המימוש החדש לפונקציות addElement, getElement, getSetSize, destroySet.מותר להשתמש בפונקציות addElement_old, getElement_old, getSetSize_old, destroySet_old, שמכילות את המימושים הקודמים לפונקציות הללו.

שאלה 2 – C++

חלק א' (30 נק')

ברצוננו לתכנן מבנה אשר ינהל את רישום הסטודנטים המתגוררים במעונות בטכניון. לצורך התרגיל נבחין בשני סוגים שונים של מעונות: מעונות לרווקים, בהם מתגוררים מספר דיירים, ומעונות למשפחות, בהן מתגוררות משפחות. מכאן נגזרים שינויים בפונקציונאליות של המבנים המטפלים בסוגי המעונות השונים, והנה מספר דוגמאות: במקרה של הרווקים, כל אחד שמתגורר במעון רשום בנפרד במערכת. במקרה של משפחות, רק אחד מבני המשפחה רשום במערכת. דירה תהיה פנויה במקרה של רווקים אם יש בה לפחות חדר אחד פנוי. לעומת זאת, דירה תהיה פנויה במקרה של משפחות רק בתנאי ולא מתגורר אף דייר בדירה.
נפתח את המבנה בשלבים.

.1
א. (10 נק)
ראשית, נגדיר מבנה עזר בשם `Dictionary`. מבנה זה משייך בין מפתח מסוים לאוסף של ערכים. להלן קטע קוד לדוגמא המשתמש במחלקה `Dictionary`:

```
int main()
{
	Dictionary<int, double, 2> dictTest; /*Dictionary in which the
	key is int and the values are double.
		Up to two values can be assigned to the same key.*/
	dictTest.AddVal(1, 1.1); /*Adds value "1.1" under key "1"*/
	dictTest.AddVal(1, 2.2); /*Adds value "2.2" under key "1"*/
	dictTest.AddVal(2, 1.1); /*Adds value "1.1" under key "2"*/
	double* vals = dictTest[1]; /*Gets the array of values under
	key "1". The returned array contains copies of the
		stored values*/
	dictTest.RemoveVal(1, 2.2); /*Removes value "3.3" under key
	"1"*/
	cout << dictTest; /*Prints out the data stored in dictTest*/
	return 0;
}
```

בהתאם לקטע הקוד ולהערות בקוד הנכם נדרשים להגדיר את הממשק של מחלקת `Dictionary`. הנכם נדרשים להגדיר רק את המתודות של המחלקה המוגדרות ב-`public` ופונקציות גלובאליות נדרשות ואין צורך להגדיר את המשתנים הפנימיים של המחלקה. אין צורך לממש את המתודות והפונקציות.
שימו לב: יש להגדיר רק את המתודות והפונקציות ההכרחיות כפי שמשתמע מקטע הקוד לדוגמא. יורד נקוד על הגדרות מיותרות.

ב. (4 נק')
מהן הדרישות לגבי טיפוס הערכים (`values`)? הגדירו סט דרישות הגיוניות ונמקו עפ"י הקוד לדוגמא.

.2
נתונות ההגדרות הבאות:

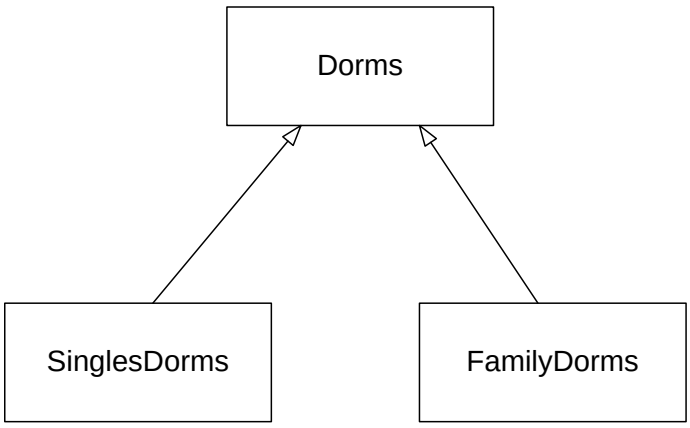
```
#define MAX_STR_LEN     256
typedef unsigned int AddressNum;

class DormAddress
{
public:
      DormAddress(AddressNum house, AddressNum flat): _house(house),
_flat(flat) { }
private:
      AddressNum _house;
      AddressNum _flat;
};

class Resident
{
public:
      Resident(const char* name, unsigned long id): _id(id) {
            strcpy(_name, name); }
private:
      char _name[MAX_STR_LEN];
      unsigned long _id;
};
```

בנוסף נתון קטע קוד לדוגמא:

```
int main()
{
      SinglesDorms<5> sd("Canada", "Dan"); /*Creates a structure for
      managing dorms for sigles,
            up to 5 room-mates in a single flat.
      The constructor contains the name of the dorms and the name of the dorms
     administrator*/

     /*Adds residents: two students in the same flat*/
     sd.AddResident(DormAddress(509, 4), Resident("Haim", 123));
     sd.AddResident(DormAddress(509, 4), Resident("Moshe", 234));

     FamilyDorms fd("Segel Zutar", "Yossi"); /*Creates a structure for
      dorms for families. The constructor contains the name of the dorms and
     the name of the dorms administrator*/

      /*Populates the flat with a family*/
      fd.PopulateFlat(DormAddress(405, 4), Resident("David", 345));

      Dorms *pd[2];
      pd[0] = &sd;
      pd[1] = &fd;

      /*Checks flat vacancy (free room/flat?)*/
      assert(pd[0]->FlatHasVacancy(DormAddress(509, 4)) == true);
      assert(pd[1]->FlatHasVacancy(DormAddress(405, 4)) == false);
      cout << "Total number of residents in dorms:" <<
      Dorms::GetTotalResidents(); /*Prints the total number of
            residents in any dorm created; family is
      counted as a single resident.*/
      /*...*/
```

```
        return 0;
}
```

א. (10 נק')
להלן תרשים המגדים את יחס ההורשה בין שלושת המחלקות Dorms, SinglesDorms ו-FamilyDorms :



הגדירו את המחלקות הנ"ל. ההגדרה חייבת להיות מלאה, ולכלול את כל המתודות, הפונקציות והמשתנים החברים הנדרשים לפי הדוגמא.
רמז: חשבו כיצד להשתמש במחלקה Dictionary בתוך המחלקות של המעונות.
שימו לב: יש להגדיר רק את המתודות , המשתנים והפונקציות ההכרחיות כפי שמשתמע מקטע הקוד לדוגמא. יורד נקוד על הגדרות מיותרות.

ב. (6 נק')
כעת מחליפי את השורה /*...*/ בקטע הקוד מסעיף א' בשורות אלה:

```
        cout << pd[0]; /*Prints the data in singles dorms created*/
        cout << pd[1]; /*Prints the data in families dorms created*/
```

הגדירו וממשו את האופרטור הנדרש. שימו לב כי שני סוגי המעונות מדפיסים דברים שונים. הנכם רשאים לשנות את הגדרות המחלקות שהגדרתם בסעיף א' אם הדבר נדרש, ובתנאי שאינכם פוגעים בתקינותה של הדוגמא שהופיעה בסעיף א'.

חלק ב (20 נק')

מה יודפס בהרצת התוכנית הבאה?

```
#include <iostream>
#include <string.h>
using namespace std;
class G {
public:
      G(const char* str){cout << " G::G(char*) ";}
      G(){cout << " G::G() ";}
      G( const G& g){cout << " G::G(G&)";}
      G& operator= (const G& g) {cout << " G::op= "; return *this;}
      ~G(){cout << " G::~G()";}
};
class H {
protected:
        G g;
public:
         H(){ cout << "H()"; }
         H(const H& h) { cout << "H( H&)"; }
         virtual ~H() {cout << " H::~H() ";}
         virtual  G  f1()=0;
         virtual void  f2(G i) const {cout << " H::f2(G) ";}
         G& f3() {cout << " H::f3() "; return g;}
     };
class K : public H {
 protected:
        G b;
public:
         K(const char* s) {cout << " K::K(char*) "; }
          K& operator=(const K& k) {cout<< " K::op=";  return *this;}
          K(const K& k) { cout << "K(K&)";}
          virtual ~K() { cout << " K::~K()" ; }
           virtual G f1(){cout << " K::f1() "; return b;}
           virtual void f2(const G g){cout << " K::f2() "; }
           G& f3() {cout << " K::f3() "; return b;}
};
int main() {
            cout << "stage 1" << endl;
            K k1("1");
            cout << endl<<" stage 2: " << endl;
            K *pk2 = new K("2");
            cout << endl<< "stage 3: " << endl;
            K k3 = *pk2;
            cout <<endl<< "stage 4: " << endl;
            H *pa = &k1;
            G g1;
            g1 =pa->f1();
            cout<<endl << "stage 5: " << endl;
            pa->f2(g1);
            cout<<endl << " stage 6: " << endl ;
            pa->f3();
            cout <<endl<< "stage 7: " << endl;
            return 0;
}
```

שאלה 3 BASH והנדסת תוכנה (20 נק')

חלק ראשון: שאלה ב-BASH (10 נק')

א. (4 נק')
רשום bash script בשם double_lines הלוקח קובץ בשם source-file, ומדפיס למסך את תוכן הקובץ כאשר כל שורה מודפסת פעמיים (רצוף).
הנחיה: חלק לשני סקריפטים, הראשון יקרא double_lines באורך שורה אחת והשני double שאורכו 4 שורות (לא כולל את שורת הקריאה ל-bash)

ב. (6 נק')
כתוב script שאורכו עד כ-10 שורות המקבל כפרמטר שם של ספריה, ומוחק בספריה זו את כל הקבצים הרגילים אשר השורה הראשונה שלהם מתחילה במחרוזת abc (3 אותיות).

הערות לכל הסעיפים
1. אין צורך לבדוק את תקינות הפרמטרים ואין צורך לרשום את השורה הקוראת ל- bash.
2. שימו לב לאורך הקצר של ה-scripts. תשובות ארוכות תחשבנה לטעות.
3. אין להשתמש בקבצי ביניים.

חלק שני: שאלות בנושא הנדסת תוכנה (10 נק')

א. (6 נק')
בהרצאה ראינו מימוש של סינגלטון בעזרת הקצאה דינאמית.

להלן הצעה נוספת למימוש הסינגלטון , הפעם ללא הקצאה דינאמית. השלם את החסר.

```
class Singleton {
private:
  …..
  int i;
.........
........
public:
  ….. ……. { ……. }
  int getValue() { return i; }
  void setValue(int x) { i = x; }
};

Singleton Singleton::s(47);

int main() {
  Singleton& s = Singleton::instance();
  cout << s.getValue() << endl;
  Singleton& s2 = Singleton::instance();
  s2.setValue(9);
  cout << s.getValue() << endl;
} ;
```

ב. (4 נק')

הלקוח יניב (Customer) מגיע לחנות ורוכש פריטים בעזרת הזמנה (Order) שהוא מכין. עליו גם לשלם כמובן, (payment) במזומן (Cash), באשראי (Credit) או ע"י צ'ק (Check) .

שרטט UML class diagram המתארת את המחלקות והיחסים ביניהם במערכת שתיארנו.