

פתרון
מבחן סוף סמסטר
מועד א'

פתרון שאלה מס' 1 (30 נק')

א. קובץ ממשק (הצהרות) של ADT שעונה על הדרישות הב"ל:

```

struct matrix;
typedef struct matrix Matrix;
typedef void *Element;
typedef Element (*Add_func)(Element,Element);
typedef void (*Free_func)(Element); /* For freeing an element */
typedef void (*Print_func)(Element); /* Printing an element */

/* Create - create a new matrix of MxN elements */
Matrix* Mat_Create(int M, int N, Add_func af, Print_func
pf, Free_func ff);

/* Index – Returns the element in the (i,j)'th loc., NULL on error */
Element Mat_Index(Matrix* mat, int i,int j);

/* Set – Set's the element in the (i,j)'th loc. Ret. FALSE on error */
BOOL Mat_Set(Matrix * mat, int i, int j, Element new);

/* Add2 – Return the addition of two Matrixes, or NULL if error */
Matrix *Mat_Add2(Matrix * mat1, Matrix * mat2);

/* Print – Prints a matrix ... */
Matrix *Mat_Print(Matrix * mat);

/* Free - frees the matrix's memory */
void Mat_Free(Matrix* mat);

```

ב. מימוש המטריצה:

```

#include "vector.h"
#include "matrix.h"
struct matrix {
    Vector *mat;
    int m,n;
    Print_func pf;

```

};

/ Create - create a new matrix of MxN elements */***Matrix* Mat_Create(int M, int N, Add_func af, Print_func pf, Free_func ff) {**

Matrix *mat;

int i;

mat=(Matrix *)malloc(sizeof(Matrix));

mat->pf = pf; mat->m=M; mat->n=N;

/ Create vector of rows */*

mat->mat = Vec_Create(M, Vec_Add2, Vec_Free);

/ Create M row - vectors of size N */*

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

Vec_Set(mat->mat,i,Vec_Create(N,af,ff));

return mat;

}

/ Index – Returns the element in the (i,j)'th loc., NULL on error */***Element Mat_Index(Matrix* mat, int i,int j) {**Vector *vec; */* Will point to line i ...*/*

if ((vec = Vec_Index(mat->mat, i)) == NULL)

return NULL;

else

return Vec_Index(vec,j);

}

/ Set – Set's the element in the (i,j)'th loc. Ret. FALSE on error */***BOOL Mat_Set(Matrix* mat, int i, int j, Element new) {**Vector *vec; */* Will point to line i ...*/*

if ((vec = Vec_Index(mat->mat, i)) == NULL)

return FALSE;

else

```

        retrun Vec_Set(vec,j,new);
    }

/* Add2 – Return the addition of two Matrixes, or NULL if error */
Matrix *Mat_Add2(Matrix * mat1, Matrix * mat2) {
    Matrix *mat;
    int i;

    if (mat1->m != mat2->m || mat1->n != mat2->n) return NULL;

    mat=(Matrix *)malloc(sizeof(Matrix));
    mat->pf = mat1->pf; mat->m=mat1->m;mat->n=mat1->n;

    /* Create the new matrix itself */
    mat->mat=Vec_Add2(mat1->mat, mat2->mat);

    return mat;
}

/* Print – Prints a matrix ... */
Matrix *Mat_Print(Matrix * mat) {
    int i,j;
    Vector *vec;
    Element e;

    for (i=0; i<mat->m; i++) {
        if ((vec = Vec_Index(mat,i)) == NULL) continue;
        for (j=0; j<mat->n; j++) {
            if (e = Vec_Index(vec,j)) {
                printf("(%d,%d): ", i, j);
                mat->pf(e);
            }
        }
    }
}

```

```
}

/* Free - frees the matrix's memory */
void Mat_Free(Matrix* mat) {
    Vec_Free(mat->mat);
    Free(mat);
}
```

פתרון שאלה מס' 2 (25 נק')

א. הגדרה של struct בשם node עבור צומת בעץ:

```
struct node {
    int key;
    struct node *left, *right, *middle;
};
```

ב. תוצאות הסיור TreeTour (תו"כ הסיור מודפס הערך בזמן הביקור בצומת) בעץ זה:
3,7,1,2,5,6,8,4,9 (יודפס משמאל לימין).

ג. כתוב קוד לממש פונקציית הסיור, (תו"כ הסיור מודפס הערך בזמן הביקור בצומת):

```
void TreeTour(node *ptr) {
    if (ptr==null) return;
    TreeTour(ptr->middle);
    printf("%d , ",ptr->key);
    TreeTour(ptr->left);
    TreeTour(ptr->right);
}
```

ד.

1.7. נוסף שדה בכל צומת שיציין לאיזה בן המסלול הכבד ביותר. (0,1,2) לשמאלי, אמצעי ימני, בהתאמה.)

שלב א: סיור רקורסיבי, לכל צומת מחשבים משקל הבנים, לוקחים כיוון לפי בן מקסימלי ומעדכנים את השדה direction, מחזירים את ערך הבן המקסימלי ועוד ערך הצומת.

שלב ב: עוברים על המסלול לפי הסימונים ומדפיסים האיברים מהשורה.

2. כתוב פונקציה רקורסיבית למימוש השלב הראשון.
פתרון:

```
int phase_1(node* ptr) {
    int a[3],max=0;//b,c;
    if(ptr==NULL) return 0;

    a[0]=phase_1(ptr->left);
    a[1]=phase_1(ptr->middle);
    a[2]=phase_1(ptr->right);

    if( a[1]>a[0] ) max=1;
    if( a[2]>a[max] ) max =2;

    ptr->direction = max;
    return(ptr->key+a[max]);
}
```

3. כתוב פונקציה למימוש השלב השני.

```
void phase_2(node* ptr) {
```

```
while(ptr!=NULL) {  
    printf("%d , ", ptr->key);  
    switch(ptr->direction) {  
        case(0): ptr=ptr->left;  
                break;  
        case(1): ptr=ptr->middle;  
                break;  
        case(2): ptr=ptr->right;  
                break;  
    }  
}
```

פתרון שאלה מס' 3 (25 נק')

א. הפלט

C
A
B
A
D
~C

ב.

מחלקה A :

הקוד נכון היות ומוגדר default constructor

מחלקה B :

הקוד נכון היות וה- default constructor נוצר כאשר הפרמטר מקבל את ערך ברירת המחדל.

מחלקה C :

הקוד שגוי !!! לא מוגדר default constructor ולכן אין לקומפילר דרך לאתחל את המערך נכון.

מחלקה D :

הקוד שגוי !!! אמנם מוגדר default constructor אבל בעל הרשאה private (ההרשאה ברירת המחדל כאשר לא מציינים כלום) ולכן לא ניתן לקרוא לו מקטע הקוד שלא שייך למחלקה ובו מוגדר המערך.

ג.

המתודה המסומנת 1 תתקמפל מכיוון שהיא שייכת למחלקה A ופועלת עליה.

המתודה המסומנת 2 לא תתקמפל כי היא פועלת על איבר זר למחלקה. חייבים להגדיר אותה כ-Friend כדי שתעבוד.

פתרון שאלה מס' 4 (20 נק')

```
#!/bin/csh
# mail the user if the files a.out/.zip/.tar/core in his directories are
bigger (in KB)
#then the parameter amount.
#Usage: total amount

set nonomatch
set amount=$3
set usr = $1
set dir=$2

foreach file ($2/*)
    if (-f $file) then
        if (( $file == a.out) || ( $file =~ *.zip) || ( $file == core ) then
            set a = `ls -l $file`
            set size=$a[5]
            if ($size >= $amount) echo "$file $size" >> /tmp/files$$
        endif
    endif
end

if (-e /tmp/files$$) then
    mail -s "files to remove" $usr <<ABC
    The date is `date`
    please remove the following files:
    `cat /tmp/files$$`
    Message was sent by $0
ABC
endif
```