

שאלה

מטרתנו לכתוב תכנית אשר מטפלת בבית ספר נתון.
בית הספר מכיל 24 כיתות המחולקות ל 3 שכבות גיל :

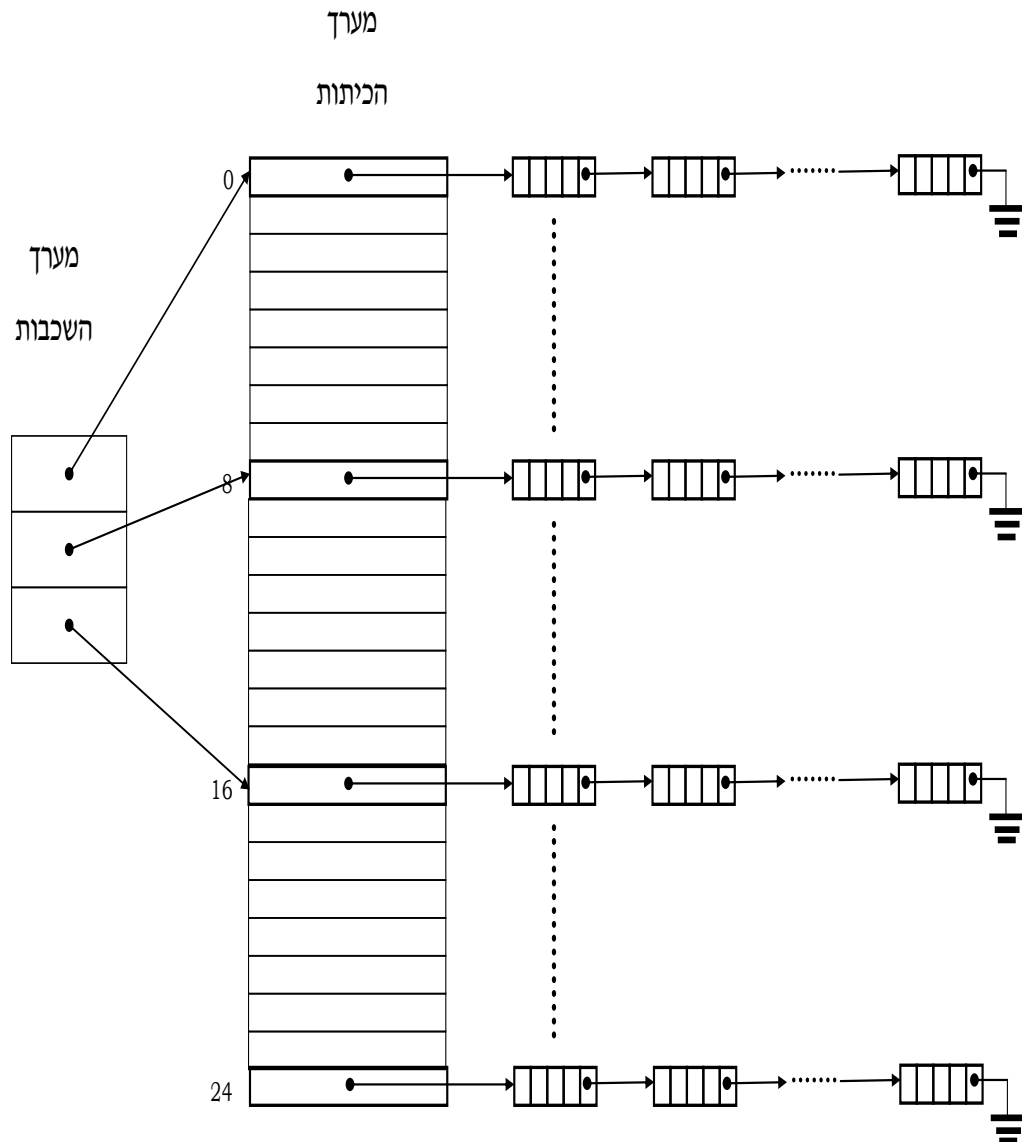
- שכבה 0 : המכילה 8 כיתות י
- שכבה 1 : המכילה 8 כיתות יא
- שכבה 2 : המכילה 8 כיתות יב .

נתון קובץ בינארי Grades.dat המכיל רשומות תלמידים במבנה הבא :

- מספר שכבה (מספר שלם בתחום $[0..2]$)
- כיתה (מספר שלם בתחום $[0..7]$)
- שם תלמיד
- טלפון
- ציון ממוצע של תלמיד

הנח כי : הקובץ Grades.dat קיים ובשאלה איננו מטפלים ביצירתו ובבנייתו.

לפניך מבנה הנתונים בו מאחסנים את נתוני התלמידים הלומדים בבית הספר הנתון.



איור 1 – מבנה הנתונים

המבנה המתואר לעיל מכיל :

- מערך השכבות של 3 איברים המיצג את 3 השכבות בבית הספר.
- כל תא במערך השכבות מכיל מצביע למערך הכיתות של 8 איברים המיצג את הכיתות באותה השכבה.
- כל תא במערך הכיתות מכיל אף הוא מצביע לרשימה מקושרת המייצגת את קבוצת התלמידים הלומדים באותה כיתה.
- כל צומת ברשימת התלמידים מכיל את הפרטים הבאים :
 - שם תלמיד
 - טלפון
 - ציון ממוצע של תלמיד
 - מצביע לרשימת התלמיד הבא (אם קיים תלמיד נוסף שנמצא אחריו ברשימה) , אחרת מצביע ל- NULL .

הערות :

- שלושת השכבות בבית ספר הנתון מיוצגות באמצעות מערך חד ממדי בשם school .
- כל הכיתות בבית ספר זה מיוצגות באמצעות מערך חד ממדי אחד בשם classes .
- במערך זה האינדקסים מ-0 עד 7 (כולל) מייצגים את השכבה 0 , האינדקסים מ-8 עד 15 (כולל) מייצגים את השכבה 1 , והאינדקסים מ-16 עד 23 (כולל) מייצגים את השכבה 2 .
- בכל אחת מן הרשימות המקושרות רשומת התלמיד הראשון ברשימה מייצגת תלמיד בעל הציון הממוצע הגבוה ביותר מבין כל הלומדים בכיתתו. יתר הרשומות ברשימה אינן מסודרות (לפי סדר עולה או יורד) בהכרח.

הנח כי אין שני תלמידים באותה כיתה שיש להם ציון הכי גבוה.

נניח שרשומת התלמיד הראשון ברשימה מסוימת מייצגת תלמיד בעל ציון ממוצע X .
אם נוסיף לרשימה זו רשומת תלמיד חדש בעל ציון ממוצע Y אז נבצע את הבדיקה הבאה :

אם $(X < Y)$ אז נוסיף את רשומת התלמיד החדש לראש רשימה זו (חשוב מדוע!)
אחרת נוסיף את רשומת התלמיד החדש כאיבר שני ברשימה זו.

התוכנית מבצעת את הצעדים הבאים :

צעד 1: אתחול מבנה הנתונים שתואר לעיל.

צעד 2: קריאת נתונים מהקובץ הבינארי Grades.dat, ואחסון במבנה הנתונים שתואר לעיל.
קריאת הרשומות (מבנים) מהקובץ הבינארי נעשית באופן סידרתי , החל מהמבנה הראשון עד וכולל המבנה האחרון.

צעד 3:

התוכנית מציגה את התפריט הבא :

1. מציאת הציון השכיח בבית הספר.
2. יצירה ובניית מערך דינאמי ממוין של תלמידים מצטיינים בשכבה מסוימת נתונה. מיון המערך הזה נעשה לפי הציון הממוצע של התלמידים באמצעות הפונקציה qsort (פונקציה שקיימת בשפת C). תלמיד נקרא מצטיין אם הציון הממוצע שלו 90 לפחות.
3. חיפוש בינארי של רשומת תלמיד מסוים במערך הדינאמי של התלמידים המצטיינים בשכבה מסוימת נתונה. החיפוש נעשה באמצעות הפונקציה bsearch (פונקציה שקיימת בשפת C).
4. מציאת התלמיד המצטיין בשכבה מסוימת ונתונה.
5. יצירת קובץ תלמידים מצטיינים בשכבה מסוימת אשר ממוין לפי שם התלמיד.
6. יציאה .

לפניך הכרזות והגדרות בשפת C בקובץ Grades.h

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#define LEVELS 3 /* מספר שכבות */
#define CLASSES_IN_LEVEL 8 /* מספר כיתות לשכבה */
typedef struct pupilType /* טיפוס צומת בעבור רשומת תלמיד ברשימה */
{
    char lname[24]; /* שם משפחה */
    char fname[12]; /* שם פרטי */
    char tel[12]; /* מספר טלפון */
    int grade ; /* ציון ממוצע */
    struct pupilType *next; /* מצביע למבנה הבא */
} pupil,*pupilPtr;

typedef struct bufferType /* טיפוס של רשומת תלמיד בקובץ בינארי */
{
    int level; /* מספר שכבה */
    int clas; /* מספר כיתה */
    char lname[24]; /* שם משפחה */
    char fname[12]; /* שם פרטי */
    char tel[12]; /* מספר טלפון */
    int grade ; /* ציון ממוצע */
} buffer,*bufferPtr;

typedef struct classType /* טיפוס נתון כיתה */
{
    struct pupilType *first; /* מצביע לראש רשימת התלמידים הלומדים בכיתה זו */
} class,*classPtr;
```

לפניך הכרזות והגדרות בשפת C בקובץ Grades.c

```
#include "Grades.h"
int n ;
pupilPtr *classes; /* מערך אחד של כל הכיתות בבית הספר הנתון */
pupilPtr **school; /* מערך של שכבות */
pupilPtr pupils; /* לצורך מערך דינאמי */
```

א. לפניך שגרה createStructure אשר יוצרת ומאתחלת את מבנה הנתונים כמתואר **בצעד 1**.

בשגרה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (4) **בלבד**, בסדר עולה, וכתוב לצד כל מספר את הביטוי החסר שהוא מייצג.

```
void createStructure(void)
{
    int k,nrows,ncols;
    nrows = LEVELS;
    ncols = CLASSES_IN_LEVEL;

    /* Allocate space for classes */
    classes = (pupilPtr*)calloc(____(1)____ , ____ (2)____ );
    if (classes == NULL)
    {
        puts("\n failure to allocate room for classes");
        exit(1);
    }
    /*Allocate space for levels */
    school = (pupilPtr**)calloc(____(3)____ , sizeof(pupilPtr));
    if (school == NULL)
    {
        puts("\n failure to allocate room for levels");
        exit(1);
    }
    for (k=0;k < nrows;k++)
    {
        school[k] = ____ (4)____ ;
    }
}
```

ב. לפניך שגרה insertGrade אשר מקבלת כפרמטר מצביע (buff) על **רשומת תלמיד מהקובץ**

הבינארי Grades.dat ומאחסנת את פרטי התלמיד במבנה הנתונים, כמפורט בפתיח השאלה **בצעד 2**, **כרשומת תלמיד ברשימה**.

בשגרה חסרים שישה ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (6) בלבד, בסדר עולה, וכתוב לצד כל מספר את הביטוי החסר שהוא מייצג.

```
void insertGrade(bufferPtr buff)
{
    int row,col;
    pupilPtr p,q;
    /*Allocate a record for pupil and extract data from buffer into it */
    p = (pupilPtr)malloc(sizeof(pupil));
    strcpy(p->lname, buff->lname);
    strcpy(p->fname, buff->fname);
    strcpy(p->tel, buff->tel);
    p->grade = buff->grade;
    row = _____(1)_____ ;
    col = _____(2)_____ ;
    q = _____(3)_____ ;
    if ( q == NULL )
    {
        _____(4)_____ ;
        p->next = NULL;
    }
    else
        if( q->grade > p->grade)
        {
            p->next = _____(5)_____ ;
            _____(5)_____ = p;
        }
        else
        {
            p->next = _____(6)_____ ;
            _____(6)_____ = p;
        }
    _____(3)_____ =q;
}
```

ג. לפניך שגרה populateStructure הקוראת נתונים מהקובץ **הבינארי** Grades.dat ומאחסנת אותם כמפורט בפתיח השאלה **בצעד** 2.

בשגרה חסרים ארבעה ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (4) בלבד, בסדר עולה, וכתוב לצד כל מספר את הביטוי החסר שהוא מייצג.

שים לב: הינך רשאי להשתמש בפונקציות שבסעיפים הקודמים.

```

void populateStructure(void)
{
    FILE *fp;
    buffer buffin;
    bufferPtr buff = &buffin;

    if ((fp = _____(1)_____)
        { puts("\n error opennig File");
          exit(1);
        }
    _____(2)_____ ;
    while (!feof(fp))
    {
        _____(3)_____ ;
        _____(4)_____ ;
    }
    fclose(fp);
}

```

ד. לפניך שגרה diligentPupils אשר מקבלת כפרמטר שכבה – level. השגרה יוצרת ובונה מערך דינאמי של תלמידים מצטיינים בשכבה הנתונה. בנוסף השגרה ממיינת את המערך הדינאמי הזה לפי הציון הממוצע של התלמידים, באמצעות הפונקציה qsort (פונקציה שקיימת בשפת C) כנדרש בצעד 3.2.

בשגרה חסרים שישה ביטויים, המסומנים במספרם בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (6) בלבד, בסדר עולה, וכתוב לצד כל מספר את הביטוי החסר שהוא מייצג.

```

void dilligentPupils(int level)
{
    int nrows = CLASSES_IN_LEVEL;
    int row;
    pupilPtr p;
    pupils = NULL;
    n= 0;

```

```

for (row=0;row < nrows;row++)
{
    p = (*(school+level)+row);
    while(p)
    {
        If ( _____(1)_____ )
        {
            pupils = _____(2)_____ ;
            strcpy(pupils[n].lname, p->lname);
            strcpy(pupils[n].fname, p->fname);
            strcpy(pupils[n].tel, p->tel);
            _____(3)_____ = p->grade;
            n++;
        }
        _____(4)_____ ;
    }
}
qsort(pupils,n, _____(5)_____ , _____(6)_____ );
}

```

כאשר

```

int comp(const void *e1, const void *e2) // Compare routine used in qsort
{
    if ( ( (pupil*) e1)->grade < ( (pupil*) e2)->grade ) return -1 ;
    if ( ( (pupil*) e1)->grade > ( (pupil*) e2)->grade ) return 1 ;
    return 0 ;
}

```

ה. לפניך שגרה electedPupil אשר מקבלת כפרמטר שכבה – level. השגרה מוצאת ומדפיסה את פרטי התלמיד המצטיין בשכבה נתונה.

בשגרה חסרים שלושה ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (3) בלבד, בסדר עולה, וכתוב לצד כל מספר את הביטוי החסר שהוא מייצג.


```

void electedPupil(int level)
{
    int nrows = CLASSES_IN_LEVEL;
    pupilPtr p, q, pmax ;
    int row ;
    p= _____(1)_____ ;
    pmax = p;

    for (row=1; row < nrows ;row++)
    {
        q = (*(school+level)+row);
        if (_____(2)_____)
        {
            pmax =_____(3)_____ ;
        }
    }
    printf("\n The best pupil is:%s  %s with average %d ",pmax→lname,
        pmax→fname,pmax→grade);
}

```

ו. לפניך שגרה frequentGrade אשר מוצאת ומחזירה את הציון השכיח בבית הספר .

בשגרה חסרים ארבעה ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (4) ב**בלבד**, בסדר עולה, וכתוב לצד כל מספר את הביטוי החסר שהוא מייצג.

```

int frequentGrade(void)
{
    int freq[101];
    int nrows = LEVELS;
    int ncols = CLASSES_IN_LEVEL;
    int row,col;
    int k,max=0;

```

```

pupilPtr p;

for(k=0;k<101;k++) freq[k]=0 ;

for(row=0;row < nrows;row++)
{
    for (col=0;col < ncols;col++)
    {
        p = school[row][col];
        while(p)
        {
            k = (int)(____(1)____);
            ____ (2) ____ ++;
            ____ (3) ____ ;
        }
    }
}
for (k=0;k < 101;k++)
{
    if (freq[k] > freq[____(4)____]) max = k;
}
return ____ (4) ____;
}

```

ז. לפניך שגרה saveArray אשר יוצרת קובץ pupils.dat של תלמידים מצטיינים בשכבה

מסוימת כאשר הקובץ ממוין לפי שם התלמיד.

שים לב: הינך רשאי להשתמש בפונקציות שבסעיפים הקודמים או במשתנה גלובלי שנבנה בסעיפים הקודמים.

בשגרה חסרים שני ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (2) בלבד, בסדר עולה, וכתוב לצד כל מספר את הביטוי החסר שהוא מייצג.

```

void saveArray()
{
    FILE *fp;
    if ((fp = fopen("pupils.dat" , _____(1)_____)== NULL)
        { puts("\n Error opening file\n");
          exit(1);
        }
    if ( _____(2)_____ != 1)
        { puts("\nError writing to the file\n");
          exit(1);
        }
    fclose(fp);
}

```

ח. לפניך שגרה mainProcess אשר מציגה את התפריט כמפורט בצעד 3 בפתח השאלה.

כל משימה תבוצע בהתאם לבחירת האופציה שבתפריט.

שים לב: הינך רשאי להשתמש בפונקציות שבסעיפים הקודמים או במשתנה גלובלי שנבנה בסעיפים הקודמים.

בשגרה חסרים ארבעה ביטויים, המסומנים במספרים בין סוגריים עגולים. רשום במחברת הבחינה את מספרי הביטויים החסרים (1) – (4) בלבד, בסדר עולה, וכתוב לצד כל מספר את הביטוי החסר שהוא מייצג.

```

void mainProcess()
{
    int reply = 1;
    int level;
    int maxfreq;
    char buf[80];
    pupilPtr ptr;
    pupil st ;

    while (reply !=6)
    { /* main Menu */
        puts("\n\n 1. מציאת הציון השכיח ");
        puts("\n 2. מציאת תלמידים מצטיינים בשכבה מסוימת והצגתם בצורה ממוינת ");
        puts("\n 3. חיפוש בינארי של תלמיד מסוים בקרב תלמידים מצטיינים ");
        puts("\n 4. מציאת התלמיד המצטיין בשכבה ");
        puts("\n 5. יצירת קובץ ממוין של תלמידים מצטיינים בשכבה ");
        puts("\n 6. יציאה ");
        printf("\n\n\n Your choice: ");
    }
}

```

```

scanf("%d", &reply);
switch(reply)
{
case 1:
    maxfreq = _____(1)_____ ;
    printf("\nGrade with Max frequency is %d ",maxfreq);
    break ;
case 2:
    printf("\n\n Enter Level: ");
    scanf("%d", &level);
    if(level<0 || level>2) puts("Out of range , try again ");
    else _____(2)_____ ;
    break;
case 3:
    printf("\n\n Enter a search Key: ");
    scanf("%s", st.lname);    scanf("%s", st.fname);
    scanf("%s", st.tel);      scanf("%d", st.grade);
    ptr = _____(3)_____ ;
    if(ptr)
        printf("\n Found: %s %s Tel:%s",ptr->fname,ptr->lname,ptr->tel);
    else
        printf ("\n %s Not found",buf);
    break;
case 4:
    printf("\n\n Enter Level: ");
    scanf("%d", &level);
    if(level<0 || level>2) puts("Out of range , try again ");
    else _____(4)_____ ;
    break;
case 5: saveArray();
        break;
case 6: return 6;

default:
    puts("Out of range , try again ");
}
}
getchar();
return 0;
}

```