תרגילים – חובת הגשה וציון – תכנון מערכות בשפת C –

מרצה: אסף בנימין tar2assaf@gmail.com

- טיפ 1 להצלחה: לפתור באופן עצמאי ללא עזרת חבר / גוגל / קומפיילר
- טיפ 2 להצלחה:
  - **<u>לפתור קודם בעט ונייר</u>**
  - בדיקות עם טבלת מעקב וקלטים חכמים
  - בסוף לבדוק במחשב
- חובה לתעד ולכתוב הסברים בתכניות */ ... /*
- שליחה בדוא"ל עד תחילת שיעור הבא

שורת נושא במייל: תכנון מערכות שפת C תרגילים פרק 12 + שם התלמיד

בהצלחה

תרגילים בנושא הקצאה דינאמית פרק 12 בספר

1. א. עיין בקטע קוד להלן ורשום 2 ביטויים אפשריים לשורה החסרה (שורה 9)
אין להשתמש בתו '\0'

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4
5  int main() {
6      int i;
7      int* arr = (int*)malloc(sizeof(int));
8
9       ____(?)_____
10     {
11         printf("ERROR! Not enough memory!\n");
12         exit(1); /* to exit the program immedietly.
13                  use only when memory allocation fails*/
14     }
15
```

ב. עיין בקטע קוד להלן ורשום 2 ביטויים אפשריים לשורה החסרה (שורה 9)
אין להשתמש בתו '\0'

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <assert.h>
4
5  int main() {
6      int i;
7      int* arr = (int*)malloc(sizeof(int));
8
9       ____(?)_____
10     /*{
11         printf("ERROR! Not enough memory!\n");
12         exit(1); /* to exit the program immedietly.
13                  use only when memory allocation fails*//*
14     }*/
15
16
```

ג. יוסי השתמש בפתרון של סעיף ב' כדי לפתור את סעיף א', הסבר מה יקרה בתכנית בסעיף א' כאשר ההקצאה תיכשל

2. עיין בתכנית הבאה וענה על השאלות בדף הבא:

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <string.h>
4
5  #define MAX 30
6
7  int main()
8  {
9      char wordArray[MAX];  /* a string */
10     char *word;
11
12     printf("Enter a word (<%d chars): ", MAX);
13     scanf("%s", wordArray);
14
15     word = (char*)malloc(sizeof(char) * (strlen(wordArray) + 1));
16
17     if (word == NULL)
18     {
19         printf("ERROR! Not enough memory!\n");
20         exit(EXIT_FAILURE);
21     }
22
23     strcpy(word, wordArray);  /* copy the string into the allocated space */
24
25     printf("  you entered: %s\n", word);
26
27     free(word);
28
29     return 0;
30 }
```

א. מה גודל המחרוזת בשורה 10? הסבר

ב. האם שורה 27 הכרחית? הסבר

ג. יוסי רשם בשורה 22 את הביטוי `word = wordArray;` ולאחר מכן הריץ. הסבר מה יקרה לתכנית בזמן ריצה.

3. מתי malloc מחזירה Null?

4. א. איך משפיעה התכנית הבאה על ה heap? ב. מהן התקלות בתכנית?

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void runMe(){
  int* leakingPtr = (int*) malloc(sizeof(int)*1024);
  for(int i=0;i<1024;i++){
    leakingPtr[i] = i+1000;
  }
 }

int main(){
runMe();
return 0;
}
```

5. כתוב פונקציה בשם CallocSheli.

בפונקציה יש לממש במדויק את הפונקציה calloc מהספרייה stdlib.h. מותר להשתמש ב malloc אך אין להשתמש ב size_t

הדרכה לעבודה נכונה : -> יש לעיין היטב בהגדרה של calloc. מהי מקבלת. מהי מחזירה ומהי עושה ולחקות אחד לאחד את הפעילות הזאת בעזרת malloc

המשך בדף בהבא

6. עיין בתכנית השגויה הבאה:

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   #define SIZE(s) s<5?1:(s>10?2:3)
4
5   void GetMoreSpace(int buffer[], int size)
6   {
7
8       buffer = (int*)realloc(*buffer, SIZE(31)*sizeof(int));
9
10  }
11  int main() {
12
13  int size, *buffer     , i;
14
15      printf("Please enter the size of the buffer: ");
16      scanf("%d", &size);
17      GetMoreSpace(buffer, size);
18
19      free(buffer);
20  }
```

א. בהנחה שהמשתמש מזין 4. פי כמה תגדיל הפונקציה את הבאפר?

ב. מהן 2 הטעויות בתכנית? הסבר

רמז: <- [redacted]

ג. תקן את התכנית כך שהיא תעבוד ואכן תגדיל את גודל הבאפר בעזרת realloc

7. הגדר פונקציה GetIchud שמקבלת 4 פרמטרים: 2 מערכים וגודלם. הפונקציה תאחד את הערכים משני המערכים ותחזיר מערך אחד הכולל את כל הערכים משני המערכים ובנוסף היא תחזיר את גודל מערך האיחוד
דוגמה: הפונקציה קיבלה מערך 1,2,3 ומערך 2,3,4,2 בגדלים 3 ו-4 בהתאמה, אז היא תחזיר את מערך האיחוד 1,2,3,2,3,4,2 ואת גודלו 7

8. א. הגדר בתכנית הראשית משתנה בשם size. בקש מהמשתמש להכניס את כמות המספרים שהוא רוצה שיהיו במערך, והקצה מערך בהתאם
ב. מלא את המערך הנ"ל במספרים 1-10 בעזרת הגרלה
ג. הגרל מספר מזל בין 1-10
ד. צור מערך זכיות שיכיל את **רק** את האינדקסים במערך הראשון שמכילים את מספר המזל
ה. הדפס את מערך הזכיות
ו. לא לשכוח לשחרר את ההקצאות הדינאמיות
ז. צרף צילומי מסך לפלט

דוגמה: עבור size=6 ועבור מערך 5,10,9,9,6,1 ועבור מספר מזל 9, מערך הזכיות יהיה בגודל 2 ותוכן איבריו יהיה 2,3

9. א. הגדר פונקציה בתכנית הראשית. היא מקבלת מחרוזת ותו ומחזירה מחרוזת. הפונקציה בודקת כמה פעמים מופיע התו שהתקבל במחרוזת שהתקבלה ומייצרת מחרוזת חדשה הזהה למספר הזה שתכיל רק את התו הזה
ב. צרף צילומי מסך לכל הפלטים החכמים שבחרת לבדוק איתם את הפונקציה
דוגמה: A. הפונקציה מקבלת "fToofr" ותו 'f' היא תחזיר מחרוזת "ff"
B. הפונקציה מקבלת "kukuriku" ותו 'k' היא תחזיר מחרוזת "kkk"

10. א. צור הגדרה של מבנה בשם Point (נקודה) שמכיל 2 שדות: x ו- y
ב. צור הגדרה של מבנה בשם Polygon (מצולע) שמכיל 2 שדות: מספר הקדקודים ומערך לקדקודים
ג. בתכנית הראשית צור 3 נקודות ואתחל אותם
ד. בתכנית הראשית צור מצולע ועדכן את מספר קדקודיו על-פי בקשת המשתמש
ה. עדכן את מערך הקדקודים כאשר שלושת הראשונים יהיו על בסיס הערכים של הנקודות מסעיף ג', ושאר ערכי הקדקודים על בסיס ערכים מוגרלים
ו. כתוב פונקציה שמקבלת מצולע (by value או by pointer, נמק מה עדיף) ומדפיסה את נתוניו
ז. צרף תמונת פלט

11. א. צור הגדרה של מבנה בשם Person שמכיל 2 שדות: שם (אורך בלתי מוגבל) ות.ז. (4 ספרות)
ב. צור הגדרה של מבנה בשם Family שמכיל את השדות הבאים: זוג הורים מסוג Person, מערך בשם kids המכיל 20 מצביעים ל- Person
ג. מהי כמות הזיכרון המינימלית בבתים שתופס Family? ומהי המקסימלית? פרט את דרך החישוב ופצל לזיכרון ב- stack לעומת זיכרון ב heap
טבלת עזר

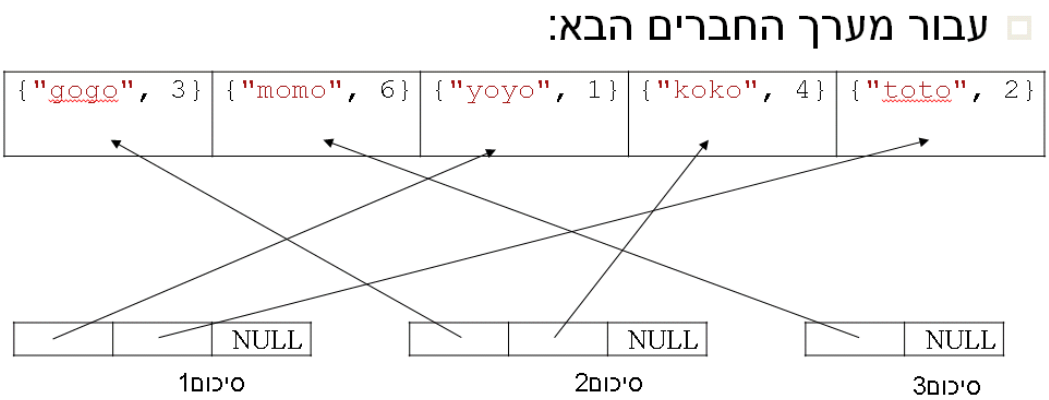
זיכרון מינימלי במחסנית	**זיכרון מינימלי בערימה heap**	**זיכרון מקסימלי במחסנית**	**זיכרון מקסימלי בערימה heap**
Xבתים	Yבתים	Zבתים	Rבתים

פירוט החישוב: ...

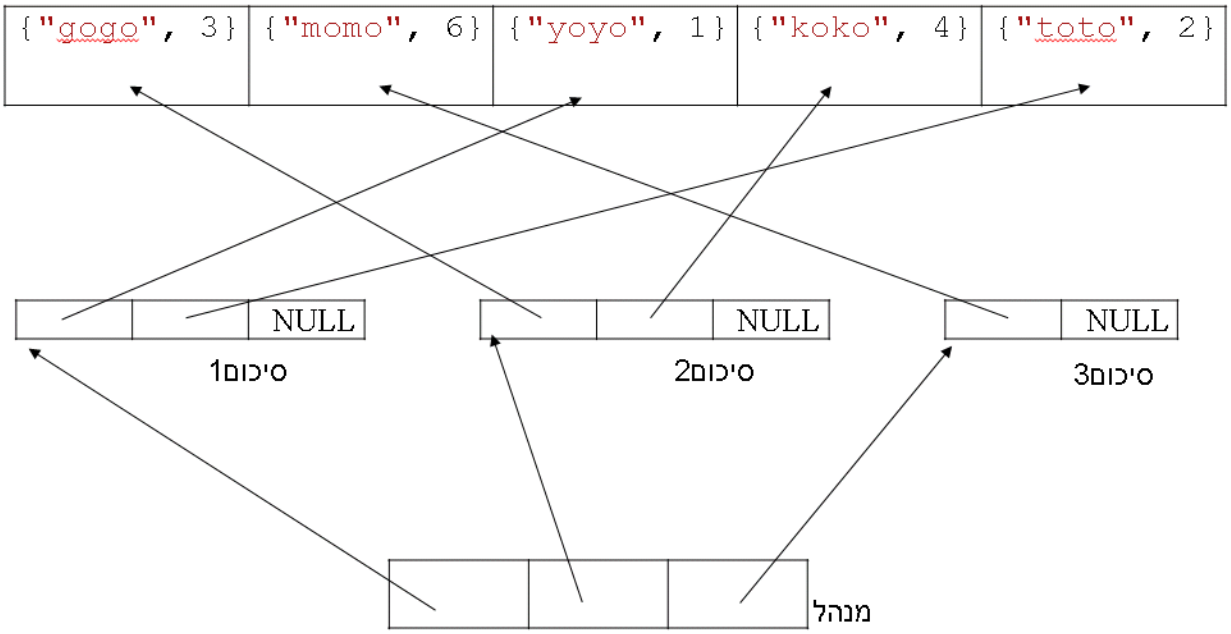
ד. כתוב פונקציות לקליטת נתוני משפחה
ה. כתוב פונקציות להדפסת נתוני משפחה
ו. כתוב תכנית ראשית המשתמשת בכל הפונקציות שכתבת
ז. מהם הקלטים "החכמים" בעזרתם אתה בודק את התכנית?

רמז -> כדאי להגדיר את מספר הילדים המקסימלי ב define. ולצורך הבדיקה לשנות את המספר ל-3. ובבדיקה לבדוק בעזרת 2 הרכבי משפחות שיכולות לבדוק את כלל התכנית. מיהם ההרכבים?
ז. צרף תמונת פלט

12. **רשות לא להגשה לשבוע זה.** א. צור הגדרה של מבנה בשם Friend המכיל 2 שדות: שם (אורך בלתי מוגבל) ואת המרחק ממנו הוא גר ממני (בק"מ).
את החברים נחלק לוגית ל-3 קטגוריות: חברים קרובים (גרים עד 2 קמ ממני), חברים בינוניים (גרים עד 5 קמ ממני) וחברים רחוקים (גרים יותר מ-5 קמ ממני)
ב. כתוב פונקציה המקבלת מערך של חברים ומייצרת 3 מערכי סיכום. כל מערך סיכום מכיל כתובת של מערך חברים.
מערך סיכום1 יכיל כתובות של איברי מערך החברים המכילים חברים קרובים
מערך סיכום2 יכיל כתובות של איברי המערך החברים המכילים חברים בינוניים
מערך סיכום3 יכיל כתובות של איברי המערך החברים המכילים חברים רחוקים.
(אין לשכפל מידע, רק מצביעים)
בסוף כל מערך סיכום יהיה null
דוגמה



ג. הפונקציה מסעיף ב' תייצר בנוסף גם מערך מנהל בגודל 3 המכיל כתובת של מערך של מצביעים לחברים וכל איבר במערך המנהל יצביע על מערך סיכום אחד
דוגמה



ד. בתכנית הראשית הגדר מערך שאבריו הם מסוג Friend ושלח אותו לפונקציה שהגדרת בסעיף ב'-ג'
ה. בתכנית הראשית: הדפס את החברים לפי החלוקה הלוגית (שתוארה בסעיף א') בעזרת מערך שיוחזר מהפונקציה של סעיף ב'-ג'

******** מכאן והלאה הכנה לבחינה. ללא הגשה ************

שאלה 1 – שאלת חובה (20 נקודות)

I. לפניך תכנית בשפת C שהפונקצייה main שלה מכילה הגדרה של משתנה בשם str .
במשתנה הזה תשוכן מחרוזת שאינה ריקה, המכילה אותיות גדולות בלבד, ללא רווחים ביניהן, ולכל אות יהיו תשעה מופעים לכל היותר במחרוזת.

בתכנית מוגדרת גם פונקצייה בשם func , אשר מקבלת כפרמטר את המחרוזת str , ומחזירה מחרוזת חדשה באופן הזה:

```
char* func(const char* str)
{
      char i, size = 0, *p = NULL;
      char englishLetters[26] = {0};
      while (*str)
      {
            englishLetters[ _____(1)_____]++;
            str++;
      }
      for (i = 0; i < 26; i++)
      {
            if (englishLetters[i])
            {
                  size =_____(2)_____;
                  p = realloc(p,size*sizeof(char));
                  if (!p){printf("Not enough memory!"); exit(1);}
                  p[_____(3)_____] = _____(4)_____;
                  p[_____(5)_____] = _____(6)_____;
            }
       }
        p =_____(7)_____;
        if (!p){ printf ("Not enough memory!"); exit(2);}
        p[_____(8)_____] = _____(9)_____;
        _____(10)_____;
    }
```

```
void main()
{
        char str[] = "ABCCBAXXX";
        char* p = func(str);
        puts(p);
        free(p);                                  //שחרור המחרוזת הדינמית
}
```

בקטע התכנית הזה חסרים **עשרה** ביטויים, המסומנים במספרים בין סוגריים עגולים. בכל אחד מן הסעיפים **א'–י'** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

א. הביטוי החסר (1) הוא:

1. *str–'A'
2. *str–1
3. *str+'A'
4. *str

ב. הביטוי החסר (2) הוא:

1. size+2
2. size+1
3. size+i
4. size*2

ג. הביטוי החסר (3) הוא:

1. size*2+1
2. size–2
3. size–1
4. size/2

ד. הביטוי החסר (4) הוא:

1. size
2. (char)('A'+i)
3. size+2
4. i

ה. הביטוי החסר (5) הוא:

1. size+1
2. size
3. size–1
4. i

ו. הביטוי החסר (6) הוא:

1. englishLetters[i+size]
2. englishLetters[i]
3. (char)(englishLetters[i]+'0')
4. englishLetters[i]+'A'

ז. הביטוי החסר (7) הוא:

1. realloc(p, size*sizeof(char)+size)
2. realloc(p, – –size*sizeof(char))
3. realloc(p, size*sizeof(char))
4. realloc(p, ++size*sizeof(char))

ח. הביטוי החסר (8) הוא:

1. size+1
2. size
3. size*2
4. size–1

ט. הביטוי החסר (9) הוא:

1. –2
2. i
3. 0
4. size

י. הביטוי החסר (10) הוא:

1. return englishLetters
2. return str
3. return p
4. return &p

שאלה 2

.II משרד התיירות עורך בקרת איכות בשישה בתי־מלון הממוספרים מ־0 עד 5 (כולל).

בכל בית־מלון נבדקים חמישה סוגי שירותים, הממוספרים מ־0 עד 4 (כולל), ולכל שירות יש **משקל**, שאינו בהכרח שווה.

סוגי השירותים הם: שירות קבלה, שירות חדרים, בריכה, שירות הסעדה ומשחקייה, שיסומנו באותיות A–E בהתאמה.

לכל בית־מלון ניתן **דירוג** עבור כל אחד מחמשת סוגי השירותים שהוא מציע לאורחיו.

לכל בית־מלון מחושב **דירוגו הסופי** (שהוא ציון משוקלל המחושב על־פי הדירוגים שקיבל בעבור השירותים שהוא מציע לאורחיו).

לכל סוג של שירות מחושב **דירוגו הממוצע** בין בתי־המלון.

נסמן את **משקל** השירות ואת **דירוג** בית־המלון, כדלהלן:

$\mathbf{W_j}$ – **המשקל** הניתן לשירות מסוג j , כאשר $0 \le j \le 4$ וגם $\sum_{j=0}^{4} W_j = 100$

$\mathbf{M_{ij}}$ – **הדירוג** שקיבל בית־המלון ה־i , בשירות מסוג j , כאשר $0 \le i \le 5$, $0 \le j \le 4$ וגם $0 \le M_{ij} \le 10$

וכעת נציג את דרך חישובם של **הדירוג הסופי** של בית־המלון ה־i , ושל **הדירוג הממוצע** של השירות ה־j :

$\mathbf{S_i}$ – **הדירוג הסופי** של בית־המלון ה־i , והוא יחושב באמצעות הנוסחה שלהלן:

$$S_i = \left(\sum_{j=0}^{4} M_{ij} * W_j \right) / 100$$

$\mathbf{P_j}$ – **הדירוג הממוצע** של השירות מסוג j בין כל בתי־המלון, והוא יחושב באמצעות הנוסחה שלהלן:

$$P_j = \left(\sum_{i=0}^{5} M_{ij} \right) / 6$$

תוצאות הבקרה נרשמות בבסיס נתונים ממוחשב, הכולל את המערכים וטיפוס הנתונים, שהגדרותיהם מובאות להלן:

```
#define NUM_OF_HOTELS  6
#define NUM_OF_SERVICES  5

typedef struct serviceType    //טיפוס סוג השירות
{
   char name[12];             //שם סוג השירות
   float avgPosition;         //דירוגו הממוצע של סוג השירות
}service,*servicePtr;
```

- `service services[NUM_OF_SERVICES]`

מערך השירותים.

- `char *names[NUM_OF_HOTELS] = {"DAN","SHERATON","HILTON","DANYEL","AKADIA","CARLTON"}`

מערך המכיל את שמות בתי־המלון.

- `char *serviceNames[NUM_OF_SERVICES]= {"A","B","C","D","E"}`

מערך המכיל את שמות סוגי השירותים.

- `int factors[NUM_OF_SERVICES] = {10,40,15,15,20};`

מערך המכיל את משקלי השירותים באחוזים.

לדוגמה: אם factors[2]=15 אז המשקל הניתן לשירות מספר 2 הוא 15 אחוז.

- `int rateArr[NUM_OF_HOTELS][NUM_OF_SERVICES]`

מערך דו־ממדי שבו משכנים בעבור כל בית־מלון את הדירוג שקיבל בעבור כל אחד מבין סוגי השירותים הניתנים בו.

לדוגמה: אם rateArr[2][4] = 8 אז בית־המלון שמספרו 2 קיבל בעבור השירות שמספרו 4 את הדירוג 8 .

- `int firstRate[NUM_OF_SERVICES]`

כאשר firstRate[j] - מכיל את מספר בית־המלון שקיבל את הדירוג הגבוה ביותר עבור סוג השירות שמספרו j , לכל $0 \le j \le 4$.

- `float hotelRates[NUM_OF_HOTELS]`

מערך "הדירוג הסופי של בתי־המלון"

כאשר hotelRates[i] יכיל את הדירוג הסופי של בית־המלון ה־i , לכל $0 \le i \le 5$.

ענה על הסעיפים י"א–י"ג שלהלן:

לפניך קטע קוד המדפיס את פרטי כל אחד מבין חמשת סוגי השירותים, כאשר השירותים ממוינים בסדר עולה לפי שדה "הדירוג הממוצע" שלהם.

בקטע הקוד מזמנים פונקצייה בשם intcmp אשר מוצגת לאחריו.

בקטע הקוד הנ"ל חסרים **שלושה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
בכל אחד מן הסעיפים **י"א–י"ג** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

```
/* Init services*/
for (i = 0; i < NUM_OF_SERVICES; i++)
{
      strcpy(services[i].name , serviceNames[i]);
      services[i].avgPosition = 0;
}
/*averages per service */
for (j = 0; j < NUM_OF_SERVICES; j++)
{
  for (i = 0; i < NUM_OF_HOTELS; i++)
      services[j].avgPosition += (float)rateArr[i][j];
   ________ (1)________;
}
qsort(___________(2)_________);
      printf("\n Category  \t Average ");
for (j = 0; j < NUM_OF_SERVICES ; j++)
   printf("\n%s \t \t%5.2f", services[j].name,services[j].avgPosition);
```

```
int intcmp(___________(3)____________)
{
  servicePtr p = v1;
  servicePtr q = v2;
  return (int)(p->avgPosition - q->avgPosition) ;
}
```

י"א. הביטוי החסר (1) הוא:

1. services[j].avgPosition /= NUM_OF_HOTELS
2. services[j].avgPosition += services [j].average
3. services[j].avgPosition += NUM_OF_HOTELS
4. avgPosition /= NUM_OF_HOTELS

י"ב. הביטוי החסר (2) הוא:

1. services, NUM_OF_SERVICES, 5, intcmp
2. services, NUM_OF_SERVICES, sizeof(services [0]), intcmp
3. &services, NUM_OF_SERVICES, 5, intcmp
4. services, NUM_OF_HOTELS, sizeof(services [0]), intcmp

י"ג. הביטוי החסר (3) הוא:

1. char *v1, char *v2
2. const void **v1, const void **v2
3. const void *v1, const void *v2
4. servicePtr *v1, servicePtr *v2

לפניך קטע קוד המחשב את דירוגו הסופי של כל בית־מלון, מאתר את בית־המלון **שהדירוג הסופי** שלו הוא הגבוה ביותר, ומדפיס את שם בית־המלון הזה ואת הדירוג הסופי שלו.

הנחה: קיים רק בית־מלון אחד שדירוגו הסופי הוא הגבוה ביותר.
כל תא במערך hotelRates מאותחל לאפס.

בקטע התכנית הזה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים. בכל אחד מן הסעיפים **י"ד–י"ז** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

```
for(i = 0; i < NUM_OF_HOTELS; i++)
 {
     for (j = 0; j < NUM_OF_SERVICES; j++)
             hotelRates[i] += _____(1)_____;
     hotelRates[i] /= 100;
 }
bestHotel = 0;
for(i = 1; i < NUM_OF_HOTELS; i++)
      bestHotel = (_____(2)_____)? i : bestHotel;
printf("\n\n The best hotel is %s with grade %5.2f",
                _____(3)_____ ,_____(4)_____);
```

י"ד. הביטוי החסר (1) הוא:

1. (float) rateArr [i][j] – factors [j]
2. (float) rateArr [i][j] / factors [j]
3. (float) rateArr [i][j] + factors [j]
4. (float)rateArr[i][j] * factors[j]

ט"ו. הביטוי החסר (2) הוא:

1. hotelRates[i] < hotelRates[bestHotel]
2. hotelRates[i] > hotelRates[bestHotel–1]
3. hotelRates[i] > hotelRates[bestHotel]
4. hotelRates[i] > hotelRates[bestHotel+1]

ט"ז. הביטוי החסר (3) הוא:

1. bestHotel
2. names[bestHotel]
3. hotelRates[bestHotel]
4. names[0]

י"ז. הביטוי החסר (4) הוא:

1. hotelRates[bestHotel]
2. names[bestHotel]
3. bestHotel
4. hotelRates[NUM_OF_HOTELS]

לפניך קטע קוד המדפיס עבור כל שירות את שמו ואת שם בית־המלון שקיבל בו את הדירוג הגבוה ביותר.

קטע הקוד בונה את המערך firstRate .

הנחה: עבור כל שירות, קיים רק בית מלון אחד שקיבל בו את הציון הגבוה ביותר.

בקטע התכנית הזה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים. בכל אחד מן הסעיפים **י"ח–כ"א** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

```
for(j=0; j < NUM_OF_SERVICES; j++)
  {
    bestQ =0;
    for (i=0; i< _____________(1)_______; i++)
            bestQ = (rateArr[i][j] > ______(2)____)? i : bestQ;
    __________(3)_____________;
  }

  for (i=0; i < NUM_OF_SERVICES; i++)
  printf("\n\n Best quailty %s has the Hotel %s",
            services[i].name, ________(4)_________);
  getchar();
  return 0 ;
```

י"ח. הביטוי החסר (1) הוא:

1. j
2. NUM_OF_SERVICES
3. bestQ
4. NUM_OF_HOTELS

י"ט. הביטוי החסר (2) הוא:

1. firstRate[j]
2. rateArr[i–1][j]
3. rateArr[bestQ][j]
4. rateArr[i][j–1]

כ. הביטוי החסר (3) הוא:

1. bestQ = firstRate[i]
2. firstRate[j] = bestQ
3. bestQ = firstRate[j]
4. firstRate[i] = bestQ

כ"א. הביטוי החסר (4) הוא:

1. names[firstRate[i]]
2. names[i]
3. firstRate[i]
4. names[bestQ]

שאלה 3

I. לפניך הגדרה חדשה:

$b = (b_0, b_1, \ldots, b_i, b_{i+1}, b_{i+2}, \ldots, b_{n-1})$ הוא מערך הממוין בסדר עולה, שאין בו איברים כפולים, כלומר: $b_0 < b_1 < \ldots b_i < b_{i+1} < b_{i+2} < \ldots < b_{n-1}$.

מערך a ייקרא **"מערך ממוין מוסט"** כאשר המערך a מתקבל מן המערך הממוין b , לאחר שכל איבר במערך b הוסט ממקומו i פעמים שמאלה בצורה מעגלית, כאשר $0 < i < n-1$, כלומר: $a = (b_i, b_{i+1}, \ldots, b_{n-1}, b_0, b_1, b_2, \ldots, b_{i-1})$.

דוגמה:

עבור המערך הממוין הזה:

	0	1	2	3	4
b =	1	2	3	4	5

ה"מערך הממוין המוסט" בעבור i=2 , ייראה כך:

	0	1	2	3	4
a =	3	4	5	1	2

נוסף על כך, להלן פונקצייה הממשת את החיפוש הבינארי. תוכל להיעזר בה.

```
int binarySearch(int a[], int n, int x)
{
   int l, h, m;
   l = 0;
   h = n-1;
   while(l <= h)
   {
      m = (l+h) / 2;
      if (a[m] == x) return m;
      if (a[m] < x) l = m+1;
      else h = m-1;
   }
   return -1;
}
```

לפניך פונקצייה שכותרתה:

```
int findShifted(int arr[], int n, int x)
```

פונקצייה זו מקבלת שלושה פרמטרים:

arr – מערך ממוין מוסט בגודל n , המכיל מספרים השונים זה מזה

n – גודל המערך

x – מספר נוסף כלשהו

אם הערך x קיים במערך arr אזי הפונקצייה תחזיר את מיקומו במערך.

אחרת – היא תחזיר את הערך (-1) .

בפונקצייה חסרים **חמישה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
בכל אחד מן הסעיפים **א'–ה'** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

```
int findShifted(int arr[], int n, int x)
{
   int low = 0,high = n-1,mid = 0;
   int index;
          // arr[mid] < arr[mid-1]  מציאת מקום במערך כך שיתקיים:
   while (low <= high)
   {
           mid = (low+high) / 2;
           if(arr[n-1] <= arr[mid])
               low = mid+1;
           else if (arr[0] >= arr[mid])
                      high = mid;
           if (_______(1)_____) break ;
   }
   index =_____________(2)__________;
   if(_________(3)______) return index;
   index = binarySearch(___________(4)_________);
   return(index == -1)? -1 :_________(5)_______;
}
```

א. הביטוי החסר (1) הוא:

1. arr[mid] == x
2. low == 0 && mid == high
3. low == mid
4. arr[mid] < arr[mid–1]

ב. הביטוי החסר (2) הוא:

1. low
2. mid
3. binarySearch(arr+mid, n–mid+1, x)
4. binarySearch(arr, mid, x)

ג. הביטוי החסר (3) הוא:

1. low <= high
2. arr[mid] == x
3. index != –1
4. arr[index] == x

ד. הביטוי החסר (4) הוא:

1. arr, n, x
2. arr+index, n–index, x
3. arr+mid, n–mid, x
4. arr, mid, x

ה. הביטוי החסר (5) הוא:

1. index
2. index+mid
3. (arr[mid] < x)? index : mid+index
4. (arr[0] > x)? mid+index : index

II. לפניך פונקצייה שכותרתה:

```
void sortPartialSorted(int a[], int n, const int k)
```

פונקצייה זו מקבלת שלושה פרמטרים:

a – מערך בגודל n , המכיל מספרים שלמים

n – גודל המערך

k – מיקום כלשהו על־פני המערך

ידוע כי כל המספרים במערך, החל מן המיקום ה־k ואילך, ממוינים בסדר עולה, וכי שאר המספרים במערך, החל מן המיקום ה־0 ועד למיקום ה־k-1 , לא בהכרח ממוינים.

הפונקצייה תמיין את המערך a בסדר עולה.

שים לב :

- יש לסרוק את המערך פעם אחת בלבד.
- המספרים שנמצאים עד המיקום ה־k-1 , כולל אותו, יכולים להיות גדולים מן המספרים שנמצאים **לאחר** המיקום ה־k-1.

דוגמה:

בעבור המערך a הזה:

0	1	2	3	4	5	6	7	8	9
17	1	23	1	4	6	9	20	25	30

וכן בעבור n = 10 ו־ k = 3 ,
הפונקצייה תחזיר את המערך שלהלן:

0	1	2	3	4	5	6	7	8	9
1	1	4	6	9	17	20	23	25	30

בארבע הפונקציות הבאות **יחד** חסרים **חמישה** ביטויים, המסומנים במספרים בין סוגריים עגולים. בכל אחד מן הסעיפים **ו'–י'** שלהלן, בחר את הביטוי החסר מבין ארבע האפשרויות הנתונות, והקף בעיגול את הספרה המייצגת אותו בדף התשובות שבנספח א'.

הערה: ביטוי להשלמה יכול לכלול כמה ביטויים להשלמה, כגון: "a, right".

```
void sortPartialSorted(int a[], int n, const int k)
{
   int i ;
   int *helper = malloc(sizeof(int)* k);
   for (i=0; i<k; i++) helper[i] = a[i];
   my_sort(helper, k);
   merge(___(1)___ , _____(2)____ , helper, k, a);
}

void my_sort(int a[], int n)
{
   int *tmp_array = malloc(sizeof(int) * n);
   internal_sort(a, n, tmp_array);
   free(tmp_array);
}

void internal_sort(int a[], int n, int helper_array[])
{
   int i;
   int left = n/2, right = n-left;
   if (n < 2) return;
   internal_sort(____(3)_____, helper_array);
```

```
    internal_sort(_______(4)_______, helper_array);
    merge(_______(3)_____, _____(4)_____, helper_array);
    for(i = 0; i < n; i++) a[i] = helper_array[i];
}
void merge(int a[], int na, int b[], int nb, int c[])    //  פעולת מיזוג
{
    int ia, ib, ic;
    for(ia = ib = ic = 0; (___________(5)_____); ic++)
    {
            if(a[ia] < b[ib])
            {
                    c[ic] = a[ia];
                    ia++;
            }
            else
            {
                    c[ic] = b[ib];
                    ib++;
            }
    }
    for(; ia < na; ia++, ic++) c[ic] = a[ia];
    for(; ib < nb; ib++, ic++) c[ic] = b[ib];
}
```

ו. הביטוי החסר (1) הוא:

1. a
2. *a[k]
3. a+k+1
4. a+k

ז. הביטוי החסר (2) הוא:

1. n
2. 1
3. k
4. n–k

ח. הביטוי החסר (3) הוא:

1. a+left, right–left
2. a+right, left
3. a, left
4. *a[left], right

ט. הביטוי החסר (4) הוא:

1. *a[left], right
2. a+left, right–left
3. a, right
4. a+right, left

י. הביטוי החסר (5) הוא:

1. ic < na+nb
2. (ia < na) && (ib < nb)
3. (ic < na) || (ic < nb)
4. (ic != na) && (ic != nb)