

מבוא למדעי המחשב – מבחן מסכם

במבחן זה 5 שאלות. יש לענות על לפחות 3 מתוכן. שתי התשובות הטובות ביותר מהוות 40% מהציון כל אחת והשלישית הכי טובה מהווה 20% מהציון. משך המבחן: 120 דקות

- שאלה 1:** כתבו פונקציה שמקבלת מחרוזת (string) של אותיות גדולות בין A ל-Z. על הפונקציה להחזיר את המספר 1 אם מופיע במחרוזת רצף עולה של ארבע אותיות (כגון EFGH) ועליה להחזיר את המספר 1- אם מופיע במחרוזת רצף יורד של ארבע אותיות (כגון HGFE) אם לא מופיע רצף כזה על הפונקציה להחזיר את המספר 0. אם מופיעים רצפים משני הסוגים הרצף הראשון מבין השניים הוא הקובע. יש להניח שהנתונים לפונקציה מגיעים באחת משתי הדרכים הבאות לפי בחירתכם:
- הפונקציה מקבלת קלט מהמשתמש באמצעות קריאה חוזרת ונשנית לפונקציה `getchar()`. התו האחרון בקלט הוא `'\n'` ומלבדו כל התווים הם אותיות גדולות בין A ל-Z.
  - הפונקציה מקבלת כפרמטר מערך מסוג `char[ ]`. התו האחרון במערך הוא `'\0'` ומלבדו כל התווים הם אותיות גדולות בין A ל-Z.

יש לרשום את הקוד בדף נפרד.

**שאלה 2:**

כתבו תוכנית מחשב שקוראת ביטוי חשבוני מתוך קובץ בשם `expression.data`, מחשבת את ערכו ומדפיסה אותו. יש להניח שבקובץ `expression.data` יש מספר אי-זוגי של שורות (כולן מסתימות ב'`\n`'). בכל שורה אי-זוגית רשום מספר ממשי ובכל שורה זוגית רשום אחד מארבעת התווים `+/*-` החישוב מתבצע לפי סדר הכתיבה (אין קדימות של כפל וחילוק על חיבור וחסור). למשל, אם בקובץ `expression.data`

1  
+  
1  
\*  
5.5  
-  
1.2  
/  
7

אזי התוכנית תדפיס לפלט את המספר 1.4 כי

$$1+1 = 2$$

$$2*5.5 = 11$$

$$11-1.2 = 9.8$$

$$9.8 / 7 = 1.4$$

**שאלה 3:**

- א. כתבו פונקציה שמקבלת כפרמטר מספר  $k$  בין 1 ל-26 ומדפיסה את  $k$  האותיות הראשונות באלף-בית הלטיני. למשל עבור  $k=8$  הפונקציה תדפיס abcdefgh
- ב. כתבו פונקציה שמקבלת כפרמטר מספר  $k$  בין 1 ל-26 ומדפיסה את  $k$  האותיות הראשונות באלף-בית הלטיני בסדר הפוך. למשל עבור  $k=8$  הפונקציה תדפיס hgfedcba
- ג. כתבו פונקציה שמקבלת כפרמטר מספר  $k$  בין 1 ל-26 ומדפיסה את ריבוע המורכב מאותיות כמו בדוגמא. למשל, עבור  $k=8$  הפונקציה תדפיס

```

a a a a a a a a a a a a a a
a b b b b b b b b b b b b a
a b c c c c c c c c c c b a
a b c d d d d d d d d d c b a
a b c d e e e e e e e d c b a
a b c d e f f f f f e d c b a
a b c d e f g g g f e d c b a
a b c d e f g h g f e d c b a
a b c d e f g g g f e d c b a
a b c d e f f f f f e d c b a
a b c d e e e e e e e d c b a
a b c d d d d d d d d d c b a
a b c c c c c c c c c c b a
a b b b b b b b b b b b b a
a a a a a a a a a a a a a a
    
```

ניתן להשתמש בפונקציות שכתבת בסעיפים א. וב. וכן בפונקציה repeat שנלמדה בכיתה. כזכור, הפונקציה מוצהרת כך:

```
void repeat(char c, int k);
```

והיא מדפיסה  $k$  פעמים את התו  $c$ . יש לרשום את הקוד בדף נפרד.

**שאלה 4:** התבוננו בקוד המקור הבא (מספרי השורות אינם חלק מקוד המקור והם הובאו לצורך הנוחיות בלבד):

```
1  #include<stdio.h>
2  #include<malloc.h>
3
4  typedef struct node {
5      struct node * next;
6      int number;
7  } * node_p;
8
9  node_p enter_new_node(int number, node_p first){
10
11     np->next = first;
12     np->number = number;
13     return np;
14 }
15
16 int calculate_sum(node_p first){
17
18
19
20
21
22 }
23
24 int main(){
25     int x=1;
26     node_p first = NULL;
27     do{
28         printf("Enter number ");
29         scanf("%d",&x);
30         first = enter_new_node(x,first);
31         printf("The sum so far is %d\n",calculate_sum(first));
32     }while (x != 0);
33     return 0;
34 }
```

התוכנית אמורה לקלוט מספרים מהמשתמש ולרשום את סכומם. יש להשלים את הקוד החסר בשורה 10 ובשורות 17-21 כך שהתוכנית תתהדר ותפעל כהלכה.

שאלה 5: כתבו פונקציה שמקבלת כפרמטר מספר ממשי  $a$  ומחשבת את האינטגרל

$$\int_a^{a+1} e^{-x^2} dx$$

בדיוק של 3 ספרות אחרי הנקודה.

(תזכורת: עבור פונקציה אינטגרבילית  $f(x)$  אינטגרל רימן על קטע  $I$  מוגדר ע"י חלוקה של הקטע  $I$

לקטעים קטנים יותר  $I_1 \dots I_r$  ובחירת נקודות כרצוננו  $x_i \in I_i$  ואז הסכום

$$\sum_{i=1}^r f(x_i) \text{length}(I_i)$$

מהווה קירוב של האינטגרל ששואף לערך האינטגרל כאשר אורכי הקטעים שואפים ל-0).

פתרון שאלה 1

הרעיון לעבור על המערך ולהשוות את האיבר שהיגענו אליו עם שלושת האיברים הבאים אחריו נשתמש בעובדה שאותיות אלפא בת אנגלית באים זה אחר זה בקוד אסקי. כלומר

'A'='B'-1

#include <stdio.h>

```
int f(char str[]){
    int i;
    for(i=0;str[i]!='\0';i++){
        if(str[i]==str[i+1]-1&&str[i]==str[i+2]-2&&str[i]==str[i+3]-3)return 1;
        if(str[i]==str[i+1]+1&&str[i]==str[i+2]+2&&str[i]==str[i+3]+3)return -1;
    }
    return 0;
}
```

שאלה 2 לא נכנסת לחומר

פתרון שאלה 3

(א) הרעיון נקבל את המספר נגדיר אינדקס חדש מאותחל מאפס נדפיס את האות הראשונה ונוסיף 1 לאינדקס שלב הבא נדפיס את האות השנייה ונוסיף אחד לאינדקס המתקבל וכך הלאה כל עוד האינדקס קטן מהמספר שקיבלנו נבצע את התהליך

#include <stdio.h>

```
void line(int k){
    int i;
    for(i=0;i<k;i++){
        printf("%c",'a'+i);
    }
}
```

(ב) הרעיון דומה לסעיף א אבל נאתחל את האינדקס לא מאפס אלא מהמספר שקיבלנו פחות אחד ובמקום להוסיף אחד נחסיר אחד עד שנגיע לאפס (בפונקציה רשום  $i > -1$  זה כמו לרשום  $i \geq 0$ ) בהתחלה נדפיס את האות ה k ית

#include <stdio.h>

```
void inv_line(int k){
    int i;
    for(i=k-1;i>-1;i--){
        printf("%c",'a'+i);
    }
}
```

(ג) ננסה להתפיס את החצי העליון של בריבועואחר כך החצי התחתון אבל קודם נגדיר את הפונקציה שמעתיקה תו מספר פעמים

```
void copy(int a,int k){
    int i;
    for(i=0;i<k;i++){
        printf("%c",'a'+a);
    }
}
```

```
void board(int k){\ \פונקצי שעושה את החצי העליון\ \
    int i;
```

```

for(i=0;i<k;i++){
    line(i);
    copy(i,2*k-2*i-1);
    inv_line(i);
    printf("\n");
}
}

void inv_board(int k){\פונקצי שעושה את החצי התחתון\
int i;
for(i=k-2;i>-1;i--){
    line(i);
    copy(i,2*k-2*i-1);
    inv_line(i);
    printf("\n");
}
}

void ful_board(int k){
    board(k);\החצי העליון של הריבוע\
    inv_board(k);\החצי נתחתון של הריבוע\
}

```

פתרון שאלה 4  
נמלא את החסר

```

1  #include<stdio.h>
2  #include<malloc.h>
3
4  typedef struct node{
5      struct node * next;
6      int number;
7  } * node_p;
8
9  node_p enter_new_node(int number, node_p first){
10     node_p np;\צריך להגדיר את המשתנה בו אנו משתמשים\
10+    np=malloc(sizeof(int)+1);\כדי שהתוכנית תעבוד נכון\
11     np->next = first;
12     np->number = number;
13     return np;
14 }
15
16 int calculate_sum(node_p first){
17     int sum=0;
18     while(first!=NULL){

```

```
19     sum+=first->number;
20     first=first->next;}
21 return sum;
22 }
23
24 int main(){
25     int x=1;
26     node_p first = NULL;
27     do{
28         printf("Enter number ");
29         scanf("%d",&x);
30         first = enter_new_node(x,first);
31         printf("The sum so far is %d\n",calculate_sum(first));
32     }while (x != 0);
33     return 0;
34 }
```

פתרון שאלה 5

```
#include <stdio.h>
#include <math.h>
#define epselon 0.001

double sum(double y){

    int i;
    double sum=0;
    for(i=0;i<1/epselon;i++){
        sum+=exp(-y*y)*epselon;
        y=y+epselon;}

    return sum;
}
```