

C

נבחן תחת שם קורס אריאל שפת C ב
מספרי קבוצות: 7424332 1442001
הנדסאים וטכנאים – הנדסת תוכנה

שם מרצה: אפי פרופוס

הנחיות לנבחן

ארבע שעות

א. משך הבחינה:

בשאלון זה שמונה שאלות.

ב. מבנה השאלון
ומפתח ההערכה:

יש לענות על שש שאלות בלבד בהתאם לפירוט הבא:

חלק א' – שאלה 1 (חובה) 15 נקודות

חלק ב' – שאלות 2-4 (יש לענות על 2 שאלות בלבד) 40 נקודות

חלק ג' – שאלות 5-8 (יש לענות על 3 שאלות בלבד) 45 נקודות

סך הכול 100 נקודות

ג. חומר עזר 1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית).

מותר לשימוש:

2. קלסר אחד בלבד עם חומר ההרצאות. אין להוציא דפים מהקלסר.

אין לצרף ספרים או חוברות עם פתרונות.

ד. הוראות כלליות: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות.

2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי. (גם בכך תלויה הערכת הבחינה).

3. יש לכתוב בעט בלבד.

4. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה.

5. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק.

6. יש להציג פתרון מלא ומנומק, כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד.

7. יש להסביר בפירוט כל תוכנית שנכתבה, תוכנית ללא הסבר מפורט לא תזכה בניקוד.

8. אם לדעתך חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר לך להמשיך בפתרון השאלה. נמק את בחירתך.

חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!
ההנחיות בשאלון זה מנוסחות בלשון זכר, אך מכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

בשאלון זה ישנם 9 עמודים כולל עמוד זה.

חלק א – שאלת חובה.

תרגיל 1: מה יודפס על ידי התוכנית הבאה?

```
#include <stdio.h>

#define f(x,y) x+y
#define g(x) x++

void main()
{
    int x=7 , y=9 , z= 0xa ;

    switch(x<<2)
    {
        case 29: printf("yes\n");
        case 28: printf("no\n");
        case 27: printf("i really dont know\n");
        default: printf("thats the answer\n");

        break;
    }

    printf("first %d\n" , x=x>>1);

    for(; x>1 ; x--)
        for(; y>0 ; y--);

    printf("second %d %d\n", x,y);

    y=4;

    if ( z & --x )
    {
        y*=2;
    }
    else
    {
        y/=2;
    }

    printf("third %d %d\n" , y , x);

    x=3; y=4;

    x = f(x,g(y))*4;

    printf("final %d\n", x);
}
```

חלק ב: יש לענות על 2 מתוך 3 השאלות הבאות.

תרגיל 2:

עליכם לבנות פונקציה המקבלת מחרוזת S, הפונקציה תמצא ותחזיר את האות הגדולה ביותר אלפבטית, שמופיעה גם באותיות קטנות וגם באותיות גדולות. אות גדולה אלפבטית נקבעת לפי מיקומה ברצף האותיות ABC האנגלי, דהיינו האות B גדולה מהאות A והאות E גדולה מהאות B.

הערות:

1. הפונקציה תחזיר את צורת ה uppercase של האות – דהיינו אם האות B מופיע גם כ b וגם כ B נחזיר את האות B בצורה הגדולה.
2. במידה והפונקציה תקבל תווים שונים מאותיות או מרווח, הפונקציה תדפיס "wrong input" עם מספר האינדקס שהתו נמצא לא תקין (יכול להיות שתדפיס מספר פעמים את ההודעת שגיאה עם מספרי אינדקסים שונים), ותמשיך כרגיל (לא תעצור).
3. במידה ולא נמצאה את העונה לצורכי התרגיל יש להחזיר רווח ' ' ולהדפיס "Not found"

הצהרת הפונקציה:

```
char func ( char *str )
```

דוגמאות:

א. במידה ונקלטה המחרוזת:

```
aaBaabcDaA
```

הפונקציה תחזיר B ולא תדפיס שום דבר נוסף.

שימו לב כי הפונקציה לא תחזיר את האות D מכיוון שהיא לא מופיעה בצורתה הקטנה.

ב. במידה ונקלטה המחרוזת:

```
CoDing
```

מכיוון שלא נמצאה אות גדולה אלפבטית שמופיעה גם בצורתה הגדולה וגם בצורתה הקטנה – יודפס

"Not found" ויוחזר ' '

ג. במידה ונקלטה המחרוזת:

```
Aba#aCfc7qaQe
```

הפונקציה תדפיס "wrong input index no 3" עבור תו #

הפונקציה תדפיס "wrong input index no 8" עבור תו 7

הפונקציה תחזיר Q

ד. במידה ונקלטה המחרוזת:

```
Codi&ng
```

הפונקציה תדפיס "wrong input index no 4" עבור התו &

הפונקציה תדפיס "Not found" ויוחזר ' '

תרגיל 3:

כתבו פונקציה **רקורסיבית** המקבלת ראש של רשימה מקושרת מטיפוס `NODE *`, הפונקציה תקשר את הרשימה מחדש כך שבסוף התהליך הרשימה תהיה ההופכית לרשימה הקיימת.

בסוף התהליך הפונקציה תחזיר את ראש הרשימה ההופכית.

שימו לב: אין לבנות רשימה חדשה או להקצות מקום חדש בזכרון! יש להשתמש ברשימה הקיימת ולשנות אותה בהתאם לצורך. ביצוע רשימה חדשה לא יזכה את הסטודנט/ית בנקודות.

הצהרת הפונקציה:

```
NODE *func ( NODE *head)
```

הצהרת המבנה:

```
Typedef struct NODE{  
    Int num;  
    struct NODE *next;  
}NODE;
```

דוגמה:

בהינתן הרשימה הבאה:

```
Head -> 3 -> 6 -> 4 -> 2 -> 23 -> 71 -> NULL
```

הפונקציה תשנה את הרשימה:

```
Head -> 71 -> 23 -> 2 -> 4 -> 6 -> 3 -> NULL
```

תרגיל 4:

כתבו פונקציה המקבלת מספר מטריצות בגודל 5*5 מטיפוס char, המטריצות יתארו מערך של מחרוזות.

הפונקציה תבנה מטריצה שלישית בגודל 5×10 שתהיה "תמונה פנורמית" של שתי המטריצות.

"תמונה פנורמית" דהיינו למצוא את האיבר המשותף בין שתי מטריצות בכל שורה ולבנות מטריצה שלישית כך שהמחרוזת עד אותו איבר במטריצה A, והמחרוזת מאותו איבר במטריצה B יהיו ביחד באותה שורה של מטריצה C (המטריצה החדשה שתבנו).

הערה: במידה ובשורה ישנם מספר איברים זהה באותה שורה של שני המטריצות – יש ליצור את התמונה עם האיבר הראשון שנמצא זהה בשני המטריצות.

דוגמא:

a	b	c	e	r
r	y	g	q	b
i	y	l	f	e
o	p	t	u	s
v	q	d	d	a

A מטריצה

q	s	c	b	d
t	g	v	c	s
a	z	z	f	p
d	u	u	f	u
w	d	d	d	a

B מטריצה

a	b	c	b	d	\0			
r	y	g	v	c	s	\0		
i	y	e	f	p	\0			
d	p	t	u	u	f	u	\0	
w	q	d	d	d	a	\0		

C מטריצה

חלק ג: יש לענות על 3 מתוך 4 השאלות הבאות:

תרגיל 5:

מה התוכנית תדפיס במידה ויועבר אליה הפרמטרים הבאים:

start היא רשימה מקושרת המכילה את האיברים הבאים:

Head -> 3 -> 6 -> 8 -> 3 -> 7 -> 2 -> 9 -> NULL

והפרמטר K שווה ל 3

```
struct Node {
    int data;
    struct Node* next;
};

Node* funcNode(Node* start, int k)
{
    if (k < 1)
        return start;

    if (start == NULL)
        return NULL;

    if (k == 1)
    {
        Node *res = start->next;
        free start;
        return res;
    }

    start->next = funcNode(start->next, k-1);
    return start;
}
```

תרגיל 6:

מה התוכנית מדפיסה?

```
void main()
{
    unsigned int x = -1;
    int y = ~0;
    if (x == y)
        printf("same\n");
    else
        printf("not same\n");

    int a[][3] = {1, 2, 3, 4, 5, 6};
    int (*ptr)[3] = a;
    printf("%d %d \n", (*ptr)[1], (*ptr)[2]);
    ++ptr;
    printf("%d %d\n", (*ptr)[1], (*ptr)[2]);
    ptr--;
}
```

תרגיל 7:

מה התוכנית מדפיסה?

```
int func2(char *str) {

    int num = 0;
    while (*str && *str>='0' && *str<='9') {
        num = num*10+(*str - '0');
        str++;
    }

    return num;
}

int func1(int num) {

    int len = 0;
    if (0==num) return 1;
    while (num) {
        len++;
        num /= 10;
    }
    return len;
}

char *addNums(char *str, int num) {

    int strnum = func2(str);
    int add = strnum + num;
    int lenadd = func1(strnum + num);
    char *newNum = (char *)malloc(sizeof(char)*(lenadd+1));
    newNum[lenadd--] = '\0';
    for (; lenadd >= 0; lenadd--) {
        newNum[lenadd] = (char)(add % 10+'0');
        add /= 10;
    }
    return newNum;
}

Void main()

{

char *str = addNums ("125ddab4d5*+",1000);

puts(str);

char *str = addNums ("123",4);

puts(str);

}
```


תרגיל 8:

אם

letters = "cTest"

digits=21423

מה יוחזר על ידי הפונקציה?

```
char *duplicate(char *letters, int *digits)
{
    int size = 1, i, j;
    char *str_res = (char *)malloc(1 * sizeof(char));

    int lenLet = strlen(letters);

    for (i = 0; i < lenLet; i++)
    {
        if (letters[i] >= 'a' && letters[i] <= 'z' || letters[i] >= 'A'
&& letters[i] <= 'Z' || letters[i] == ' ')
        {
            for (j = 0; j < digits[i] ; j++)
            {
                str_res[size - 1] = letters[i];
                size++;
                str_res = (char *)realloc(str_res, size *
sizeof(char));
            }
        }
        else
        {
            return NULL;
        }
    }
    str_res[size - 1] = '\0';
    return str_res;
}
```

בהצלחה !