גירסה 1.00 – 23.6.2007


# תשובות לשאלות - שפת C

ניר אדר



Nir Adar

Email: nir@underwar.co.il

Home Page: http://www.underwar.co.il

אנא שלחו תיקונים והערות אל המחבר.

# 1. תוכן עניינים

# 2. פתיחה

מאגר התשובות לשפת C מכיל פתרונות לכל התרגילים המופיעים במאגר השאלות אותו פרסמתי בשנת 2001, ועדכנתי לאחרונה ב-2005. השאלות הינן שאלות שנאספו לאורך השנים בהן העברתי שיעורים פרטיים בנושא שפת C, והן נבחרו כך שיקיפו את כל יסודות השפה ויאפשרו לפותר אותן להתקל במצבים רבים הקיימים בעולם התכנות.

שנים רבות מסמך הפתרונות היה בכתיבה וחיכה להשלמה בין כל המשימות האחרות שלי. כעת סוף סוף המאגר הושלם והרי הוא לפניכם. המאגר שלפניכם מתאים לגירסה 1.03 של מסמך השאלות. תיקונים והערות יתקבלו בברכה.
מאגר זה אינו מכיל פתרונות מנומקים אלא דוגמאות קוד בלבד. ככל שהזמן יאפשר, אוציא בעתיד גרסה מורחבת שתכלול גם הסברים והכוונות לדרכי הפתרון המוצגות במסמך זה.

כולי תקווה שמסמך זה יהיה לכם לעזר.

ניר אדר,
יוני 2007

# 3. קלט פלט

## 3.1. רמה קלה

### תרגיל 1

```
#include <stdio.h>

int main()
{
	printf("************\n");
	printf("* I Love C *\n");
	printf("************\n");
	return 0;
}
```

### תרגיל 2

```
#include <stdio.h>

int main()
{
	printf("34\n4535\n7899\n23\n");
	return 0;
}
```

### תרגיל 3

```
#include <stdio.h>

int main()
{
	printf("I love learning C\n");
	printf("I\nlove\nlearning\nC\n");
	printf("$$$$$$$$$$$$$$$$$$$$\n");
	printf("$ I love learning C $\n");
	printf("$$$$$$$$$$$$$$$$$$$$\n");
	return 0;
}
```

# 4. משתנים

## 4.1. רמה קלה

### תרגיל 1

```
#include <stdio.h>

int main()
{
	int x, y;

	/* Get an input from the user */
	printf("Please enter two integers: ");
	scanf("%d%d", &x, &y);

	/* Output the results */
	printf("%d + %d = %d\n", x, y, x+y);
	printf("%d - %d = %d\n", x, y, x-y);
	printf("%d * %d = %d\n", x, y, x*y);
	return 0;
}
```

### תרגיל 2

```
#include <stdio.h>

int main()
{
	int x, y, z;
	double average;

	/* Get an input from the user */
	printf("Please enter three integers: ");
	scanf("%d%d%d", &x, &y, &z);

	/* Calculate the average - pay attention to the 3.0! */
	average = (x+y+z) / 3.0;

	/* Output the results */
	printf("The average is %lf\n", average);
	return 0;
}
```

## 4.2. רמה בינונית

### תרגיל 1

פלט התוכנית:

```
5
```

### תרגיל 2

```
#include <stdio.h>

int main()
{
	int rails_number;
	printf("Please enter number of nails: ");
	scanf("%d", &rails_number);

	printf("The number of full nail boxes is %d\n",
		rails_number/50);

	return 0;
}
```

### תרגיל 3

```
#include <stdio.h>

int main()
{
	int x1, y1;
	int x2, y2;
	int len1, len2;
	scanf("%d%d%d%d", &x1, &y1, &x2, &y2);

	/* no need for power and sqrt, as one of
	     the elements is zero as we assume */
	len1 = x1+y1;
	len2 = x2+y2;

	printf("Triangle area: %lf\n", (len1+len2)/2.0);

	return 0;
}
```

### תרגיל 4

א. שגיאה בזמן ריצה – אנחנו מבצעים חלוקה ב-0.

ב. שגיאה בזמן ריצה – אנחנו ניגשים למשתנים שלא אתחלנו בערך משמעותי.

### תרגיל 5

```
#include <stdio.h>

int main()
{
	int road;
	int speed;

	printf("Please enter the road (in km): ");
	scanf("%d", &road);

	printf("Please enter the driving speed (km/h): ");
	scanf("%d", &speed);

	printf("It will take you %d hours and %d minutes to arrive
		to your target.\n",
		road/speed, (road % speed) * 60 / speed );


	return 0;
}
```

# 5. הסתעפויות

## 5.1. רמה קלה

### תרגיל 1

```
#include <stdio.h>

int main()
{
	int x1, x2, x3, max;
	printf("Please enter 3 numbers: ");
	scanf("%d%d%d", &x1, &x2, &x3);

	if (x1 >= x2 && x1 >= x3) max = x1;
	else if (x2 >= x1 && x2 >= x3) max = x2;
	else max = x3;

	printf("The biggest numbers is %d\n", max);

	return 0;
}
```

### תרגיל 2

```
#include <stdio.h>

int main()
{
	int num, abs_value;
	printf("Please enter a number: ");
	scanf("%d", &num);

	abs_value = ( (num >= 0) ? num : -num );

	printf("The absolute value of the number is %d\n", abs_value);

	return 0;
}
```

### תרגיל 3

```
#include <stdio.h>

int main()
{
	int age, drink;
	printf("Welcome to BAR system.\n");
	printf("Enter your age: ");
	scanf("%d", &age);
	if (age < 18)
	{
		printf("Sorry, you're too young to use the BAR system,
		 Goodbye.\n");
		return 0;
	}

	printf("Great, please select 1 for beer or 2 for good wine: ");
	scanf("%d", &drink);
	if (drink == 1)
		printf("Beer is the best drink in the world!\n");
	else if (drink == 2) printf("Glass of red wine, yammi\n");
	else printf("Sorry, wrong number!\n");

	return 0;
}
```

## 5.2. רמה בינונית

### תרגיל 1

```
#include <stdio.h>
#include <math.h>

int main()
{
	int a, b, c;
	int delta;
	double real, img;
	printf("Please enter a, b, c: ");
	scanf("%d%d%d", &a, &b, &c);

	/* Check if there are solutions */
	if (a == 0 && b == 0 && c != 0)
	{
		printf("No solution.\n");
		return 0;
```

```
	}

	/* Infinite number of solutions */
	else if (a == 0 && b == 0 && c == 0)
	{
		printf("Infinite number of solutions.\n");
		return 0;
	}

	/* One-degree equation? */
	else if (a == 0)
	{
		printf("X = %.2lf\n", (double)-c / b);
		return 0;
	}

	/* Calculate delta */
	delta = b*b - 4*a*c;

	/* If delta is positive - there are 2 solutions */
	if (delta > 0)
	{
		printf("X1 = %.2lf\n", (-b + sqrt(delta))/(2*a));
		printf("X2 = %.2lf\n", (-b - sqrt(delta))/(2*a));
		return 0;
	}

	/* If delta is zero - there is one solution */
	else if (delta == 0)
	{
		printf("X = %.2lf\n", (double)-b/(2*a));
		return 0;
	}


	/* If delta is negative - the solutions belongs to
	   complex domain */
	else
	{
		real = (double)(-b)/(2*a);
		img = (double)sqrt(-delta)/(2*a);

		/* x1 */
		printf("X1 = ");
		if (real != 0)
		{
			printf("%.2lf", real);
			if (img > 0) printf(" + ");
			else if (img < 0) printf(" - ");
			if (img == 1 || img == -1) printf("i\n");
			else printf("%.2lfi\n", img);
			return 0;
		}
		else
		{
			if (img == 1) printf("i\n");
			else if (img == -1) printf("-i\n");
			else printf("%.2lfi\n", img);
		}

		/* x2 */
```

```
            img = -img;
            printf("X2 = ");
            if (real != 0)
            {
                  printf("%.2lf", real);
                  if (img > 0) printf(" + ");
                  else if (img < 0) printf(" - ");
                  if (img == 1 || img == -1) printf("i\n");
                  else printf("%.2lfi\n", img);
                  return 0;
            }
            else
            {
                  if (img == 1) printf("i\n");
                  else if (img == -1) printf("-i\n");
                  else printf("%.2lfi\n", img);
                  return 0;
            }
      }
      return 0;
}
```

## תרגיל 2

```
#include <stdio.h>
#include <math.h>

#define PI 3.14159265358979323846264338327795

int main()
{
      /* Variables for storing input from the user */
      double num;
      char action;

      /* The following variables will hold the number in radians
         and degrees. Will be used for display */
      double rad_value, deg_value;


      /* Get input from the user */
      scanf("%lf%c", &num, &action);

      /* Number must be possitive or zero */
      if (num < 0)
      {
            printf("Error: Negavive number was entered\n");
            return 0;
      }

      /* Calculate the degrees and the radians */
      switch(action)
      {
      case 'D':
            deg_value = num;
            rad_value = num / 180 * PI;
```

```
            break;
      case 'R':
            rad_value = num;
            deg_value = num / PI * 180;

            break;

      default:
            printf("Error: Unknown Action\n");
            return 0;
            break;
      }

      /* Output results */
      printf("%.3lf Degrees = %.3lf Radians\n",
            deg_value, rad_value);
      printf("cos(%.3lf) = %.3lf, sin(%.3lf) = %.3lf\n",
            deg_value, cos(rad_value), deg_value, sin(rad_value));

      return 0;
}
```

## 5.3. רמה קשה

### תרגיל 1

הערך האפשרי היחיד עבור x הינו 4.

# 6. לולאות

## 6.1. רמה קלה

### תרגיל 1

```
#include <stdio.h>

int main()
{
	int i;
	for (i = 1; i <= 10; i++)
	{
		if (i == 7) continue;
		printf("%d\t", i);
	}
	putchar('\n');
	return 0;
}
```

### תרגיל 2

```
#include <stdio.h>

int main()
{
	int i;
	for (i = 1; i <= 100; i++)
	{
		if (i % 7 == 0 && i % 3 == 0) printf("%d\t", i);
	}
	putchar('\n');
	return 0;
}
```

### תרגיל 3

```
int azeret(int n)
{
	int result = 1, i;
	for (i = 1; i <= n; ++i) result *= i;
	return result;
}
```

### תרגיל 4

```
int power(int a, int b)
{
	int result = 1, i;
	for (i = 0; i < b; ++i) result *= a;
	return result;
}
```

### תרגיל 5

```
int fib(int n)
{
	int num1 = 0, num2 = 1, cur, cur_index = 1;
	if (n < 2) return n;

	while (cur_index < n)
	{
		cur = num1+num2;
		num1 = num2;
		num2 = cur;
		cur_index++;
	}

	return cur;
}
```

### תרגיל 6

```
int prime(int n)
{
	int i;
	for (i = 2; i < n; i++)
	{
		if (n % i == 0) return 0;
	}

	return 1;
}
```

### תרגיל 7

```
void kefel()
{
	int i, j;

	for (j = 1; j <= N; j++)
	{
		for (i = 1; i <= N; i++) printf("%d\t", i*j);
		putchar('\n');
	}
}
```

### תרגיל 8

```
int get_digit(unsigned long x, int i)
{
	int counter;
	for (counter = 1; counter < i; counter++) x /= 10;
	return x % 10;
}
```

### תרגיל 9

```
#include <stdio.h>

int main()
{
	int sum = 0, num;
	while (scanf("%d", &num) == 1) sum += num;
	printf("The sum is %d\n", sum);
	return 0;
}
```

### תרגיל 10

מושאר כתרגיל מחשבתי למזוכיסטים ביניכם. אין לי כוונה לפתור אותו.

### תרגיל 11

עקב השימוש ב-()random ו-()randomize, תוכנית זו תעבור קומפילציה ב-Borland C בלבד.

ניתן להשתמש ב-()rand על מנת לכתוב תוכנית שתעבור כל מהדר.

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

int main()
{
      int x, number, tries;

      /* the computer select a number */
      randomize();
      number = random(100) + 1;

      /* let the user guess number */
      printf("Please enter a number between 1 to 100: ");
      scanf("%d", &x);

      for (tries = 1; tries <= 10; tries++)
      {

            /* If the number is bigger than the number
               the computer selected, tell "number is bigger" */
            if (x < number) printf("Try bigger number...\n");
            else if (x > number) printf("Try smaller number...\n");
            else
            {
                  /* Tell the user he found the number: */
                  printf("You found it!\n");
                  return 0;
            }

            /* Let the user guess another number */
            printf("Please enter a number between 1 to 100: ");
            scanf("%d", &x);

      }

      printf("Number of tries exceed. Game over\n");

      return 0;
}
```

## 6.2. רמה בינונית

### תרגיל 1

```
int digits_number(unsigned long x)
{
	int counter = 0;
	while (x > 0)
	{
		counter++;
		x /= 10;
	}

	return counter;
}
```

### תרגיל 2

```
int sum_digit(unsigned long x)
{
	int sum = 0;
	while (x > 0)
	{
		sum += x % 10;
		x /= 10;
	}

	return sum;
}
```

### תרגיל 3

```
int main()
{
	int max1, max2, max3;
	int i1, i2;
	int x;

	scanf("%d%d%d", &max1, &max2, &max3);
	i1 = max2; i2 = max3;
	while (scanf("%d", &x) == 1)
	{
		if (i1+i2+x > max1+max2+max3)
		{
			max1 = i1;
```

```
                    max2 = i2;
                    max3 = x;
              }

              i1 = i2;
              i2 = x;
       }
       printf("maximum : %d %d %d\n", max1, max2, max3);
       return 0;
}
```

## תרגיל 4

```
#include <stdio.h>

void swap(int *a, int *b)
{
       int temp = *a;
       *a = *b;
       *b = temp;
}

int main()
{
       int n1, n2;
       scanf("%d%d",&n1, &n2);
       if (n1 > n2) swap(&n1, &n2);

       while(scanf("%d",&check) == 1 && check != -1)
       {
              if (check > 0)
              {
                     if (n2 < 0) n2 = check;
                     else if (n1 < 0) n1 = check;
                     else if (check < n1)
                     {
                            n2 = n1;
                            n1 = check;
                     }
                     else if (check < n2)
                     {
                            n2 = check;
                     }
              }
       }

       printf("%d %d\n", n1, n2);

       return 0;
}
```

### תרגיל 5

הפתרון משתמש ב-()digits_number מתרגיל 1.

```
int digits_number(unsigned long x)
{
      int counter = 0;
      while (x > 0)
      {
            counter++;
            x /= 10;
      }

      return counter;
}

int get_left_digit(unsigned long x, int i)
{
    int n, x_length = digits_number(x);
    for (n = 0; n < x_length - i; ++n)
    {
        x /= 10;
    }
    return x % 10;
}
```

### תרגיל 6

```
void print_zigzag()
{
    int i = 1, negative = 1;
    for (i = 1; i <= 10; ++i)
    {
        printf("%d\t", i*negative);
        negative *= -1;
    }
}
```

### תרגיל 7

```
int main(int argc, char *argv[])
{
  int max, cur;

  if (scanf("%d", &max) != 1)
  {
      printf("Input Error\n");
      return -1;
  }
  while (scanf("%d", &cur) == 1)
  {
      if (cur > max) max = cur;
  }

  printf("Biggest Number on stream: %d\n", max);

  return 0;
}
```

### תרגיל 8

```
int main(int argc, char *argv[])
{
  int max_zero = 0, zero_counter = 0, cur;

  while (scanf("%d", &cur) == 1)
  {
      if (cur == 0)
      {
          ++zero_counter;
          if (zero_counter > max_zero) max_zero = zero_counter;
      }
      else zero_counter = 0;
  }

  printf("Max zeros: %d\n", max_zero);

  return 0;
}
```

## 6.3. רמה קשה

### תרגיל 1

```
#include <stdio.h>

int main()
{
	int n, k;
	int n1, n2, k1, k2;
	printf("Please enter n, k: ");
	scanf("%d%d", &n, &k);

	/* print n chess lines */
	for (n1 = 0; n1 < n; ++n1)
	{
		/* repeat one line k times */
		for (k1 = 0; k1 < k; ++k1)
		{
			/* print n cells */
			for (n2 = 0; n2 < n; ++n2)
			{
				/* draw a cell */
				for  (k2 = 0; k2 < k; ++k2)
				{
					if ((n1 + n2) % 2) putchar('#');
					else putchar(' ');
				}
			}
			putchar('\n');
		}
	}
	return 0;
}
```

### תרגיל 2

הפתרון באדיבות שושן כהן (psclil@gmail.com).

```
//Alef - can be converted to n/2+n^2/2
int sum1(int n)
{
    if (n <= 0) return 0;
    int s = n;
    while (--n) s+=n;
    return s;
}

// Bet - sum of (j+1)(k-j) for j from 0 to k-1
int sum2(int k)
{
    int s = 0;
    int j;
    for (j=0;j<k;++j)
        s += sum1(j+1); // sum1(j+1) - same as (j+1)(k-j)
    return s;
}
```

# 7. מצביעים

## 7.1. רמה קלה

### תרגיל 1

```
void swap(double *d1, double *d2)
{
    double temp = *d1;
    *d1 = *d2;
    *d2 = temp;
}
```

## 7.2. רמה בינונית

### תרגיל 1

```
i = 5, j = 10
i = 31, j = 5
```

### תרגיל 2

```
void swap(int *i1, int *i2)
{
    *i1 = *i1 + *i2;
    *i2 = *i1 - *i2;
    *i1 = *i1 - *i2;
}
```

# 8. מערכים

## 8.1. רמה קלה

### תרגיל 1

```
int sum_array(int a[], int n)
{
	int i, sum = 0;
	for (i = 0; i < n; i++) sum += a[i];
	return sum;
}
```

### תרגיל 2

```
int has_sum(int a[], int n)
{
	/* x1 + x2 = x3 */
	int x1, x2, x3;


	for (x3 = 0; x3 < n; ++x3)
	{
		for (x1 = 0; x1 < n; ++x1)
		{
			if (x1 == x3) continue;

			for (x2 = 0; x2 < n; ++x2)
			{
				if (x2 == x3 || x2 == x1) continue;
				if (a[x1] + a[x2] == a[x3]) return 1;
			}
		}
	}

	return 0;
}
```

### תרגיל 3

```
void Analyze_array(int* arr, int arr_size, int *x1, int *x2, double
*x3)
{
	int min = arr[0];
	int max = arr[0];
	int sum = 0;
	int counter;

	for (counter = 0; counter < arr_size; counter++)
	{
		if (arr[counter] < min) min = arr[counter];
		if (arr[counter] > max) max = arr[counter];
		sum += arr[counter];
	}

	*x1 = min;
	*x2 = max;
	*x3 = (double)sum / arr_size;
}
```

### תרגיל 4

```
int main()
{
	int arr1[10] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
	int arr2[10] = { 0 };
	int arr3[10], counter;
	for (counter = 0; counter < 10; counter++) arr[counter] = 0;
}
```

אם נדרש לאתחל את כל התאים ל-7, אנו יכולים לעשות זאת בשתי דרכים: אתחול מפורש תא אחר תא, או לולאה. לא ניתן להשתמש בטריק שהשתמשנו בו במקרה של arr2.

### תרגיל 5

```
#define N 10
typedef int mat[N][N];

int sum_matrix(mat m)
{
    int i, sum = 0;
    for (i = 0; i < N; ++i) sum += m[i][i];
    return sum;
}
```

### תרגיל 6

```
void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

void rotate_array(int arr[], int n)
{
    int counter;
    for (counter = 0; counter < n / 2; ++counter)
        swap(&arr[counter], &arr[n-1-counter]);
}
```

### תרגיל 7

```
int only_zeros(int mat[N][M])
{
      int i, j;
      for (i = 0; i < N; ++i)
      {
            for (j = 0; j < M; ++j)
            {
                  if (mat[i][j] != 0) return 0;
            }
      }
      return 1;
}
```

### תרגיל 8

```
void print_reverese_odd(int mat[N][M])
{
      int i, j;
      for (i = 0; i < N; ++i)
      {
            for (j = 0; j < M; ++j)
            {
                  if (i % 2 == 0) printf("%d\t", mat[i][j]);
                  else printf("%d\t", mat[i][M-1-j]);
            }
      }

}
```

## 8.2. רמה בינונית

### תרגיל 1

```
int has_sum (int arr[], int n, int target)
{
    int low_value = 0, high_value = n-1;

    while (low_value < high_value)
    {
        if (arr[low_value] + arr[high_value] == target) return 1;
        if (arr[low_value] + arr[high_value] > target) high_value--;
        else if (arr[low_value] + arr[high_value] < target)
            low_value++;
    }
    return 0;
}
```

### תרגיל 2

לצורך מימוש תרגיל זה נשתמש באלגוריתם המוכר merge_sort שהוא בעל סיבוכיות O(nlogn).

```
#define TRUE 1
#define FALSE 0

int two_equals(int* arr, int n)
{
    int counter;
    merge_sort(arr, n);
    for (counter = 0; counter < n - 1; ++counter)
        if (arr[counter] == arr[counter+1]) return TRUE;
    return FALSE;

}
```

## תרגיל 3

```
#include <stdio.h>

#define NUMBER_OF_DAYS 7

int main()
{
  // holds the number of tickets ordered in each day
  int days[NUMBER_OF_DAYS] = { 0 };
  int cur_day, tickets_number;

  int max, min, counter;

  int days_counter, print_counter;

  // get all the data about the tickets from the user
  while (scanf("%d%d", &cur_day, &tickets_number) == 2)
  {
      if (cur_day < 0 || cur_day >= NUMBER_OF_DAYS)
      {
          printf("ERROR: invalid date\n");
          continue;
      }

      if (tickets_number < 0)
      {
          printf("ERROR: invalid number of tickets\n");
          continue;
      }

      days[cur_day] += tickets_number;
  }

  // find max and min
  max = min = 0;
  for (counter = 0; counter < NUMBER_OF_DAYS; ++counter)
  {
      if (days[counter] > days[max]) max = counter;
      if (days[counter] < days[min]) min = counter;
  }

  printf("max day is %d. min day is %d\n\n", max, min);

  // make all days array to be between 0-10
  for (counter = 0; counter < NUMBER_OF_DAYS; ++counter)
  {
      days[counter] /= (days[max] / 10);
  }

  for (print_counter = 0; print_counter < 10; ++print_counter)
  {
      for (days_counter = 0; days_counter < NUMBER_OF_DAYS;
++days_counter)
      {
            if (days[days_counter] >= 10 - print_counter)
putchar("*");
            putchar('\t');
      }
```

```
      putchar('\n');
  }
  for (print_counter = 0; print_counter < 10; ++print_counter)
    printf("%d\t", print_counter);
  putchar('\n');


  system("PAUSE");
  return 0;
}
```

## תרגיל 4

```
#include <stdio.h>

#define X 10
#define Y 20

#define SUCCESS 1
#define FAILURE 0

#define FREE 0
#define OCCUPIED 1

typedef int Cinema[X][Y];


int FindPlaces(Cinema theCinema, int tickets, int* x, int* y);
int NextCellsAreFree(Cinema theCinema, int tickets, int i, int j);
void MarkPlacesAsFull(Cinema theCinema, int tickets, int i, int j);


int main()
{
      Cinema theCinema = { 0 };
      int tickets;
      int x, y;

      while (scanf("%d", &tickets) == 1)
      {
            if (FindPlaces(theCinema, tickets, &x, &y) == SUCCESS)
            {
                  MarkPlacesAsFull(theCinema, tickets, x, y);
                  printf("The places are from (%d, %d) to
                          (%d, %d)\n", x, y, x, y+tickets);
            }
            else printf("Not enough place in the cinema\n");
      }
      return 0;
}



int FindPlaces(Cinema theCinema, int tickets, int* x, int* y)
{
      int i, j, k;
```

```
	for (i = 0; i < X; ++i)
	{
		for (j = 0; j < Y; ++j)
		{
			if (theCinema[i][j] == FREE &&
				NextCellsAreFree(theCinema, tickets, i, j))
			{
				*x = i;
				*y = j;
				return SUCCESS;
			}
		}
	}

	return FAILURE;
}

int NextCellsAreFree(Cinema theCinema, int tickets, int i, int j)
{
	int counter;
	for (counter = 0; counter < tickets; ++counter)
	{
		if (j+counter >= Y) return FAILURE;
		if (theCinema[i][j+counter] != FREE) return FAILURE;
	}
	return SUCCESS;
}


void MarkPlacesAsFull(Cinema theCinema, int tickets, int i, int j)
{
	int counter;
	for (counter = 0; counter < tickets; ++counter)
	{
		theCinema[i][j+counter] = OCCUPIED;
	}
}
```

# 9. מיונים

## 9.1. רמה קלה

### תרגיל 1

```
void bubbleSort(int numbers[], int array_size)
{
  int i, j, temp;

  for (i = (array_size - 1); i >= 0; i--)
  {
    for (j = 1; j <= i; j++)
    {
      if (numbers[j-1] > numbers[j])
      {
        temp = numbers[j-1];
        numbers[j-1] = numbers[j];
        numbers[j] = temp;
      }
    }
  }
}
```

## תרגיל 2

```
void merge(int a[], int na, int b[], int nb, int c[])
{
	int i=0,j=0,k=0;

	while(i < na && j < nb)
	{
		if(a[i]>b[j])
		{
			c[k]=b[j];
			k++;
			j++;
		}
		else
		{
			c[k]=a[i];
			k++;
			i++;
		}
	}

	while (i < na)
	{
		c[k]=a[i];
		k++;
		i++;
	}

	while (j < nb)
	{
		c[k]=b[j];
		k++;
		j++;
	}
}
```

# 10. מחרוזות

## *10.1. רמה קלה*

### תרגיל 1

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

#define MAX_PASS        (20)

int main(int argc, char *argv[])
{
  char pass[MAX_PASS];
  printf("Please enter the password: ");
  gets(pass);
  if (strcmp(pass, "12345") != 0)
  {
    printf("Wrong Password\n");
    return -1;
  }
  printf("Password OK\n");

  system("PAUSE");
  return 0;
}
```

### תרגיל 2

```
unsigned my_strlen(char *s)
{
      int counter = 0;
      /* while we didn't reach the end of the string */
      while (*s != '\0')
      {
            counter++;
            s++;
      }
      return counter;
}
```

### תרגיל 3

```
long num(char *s)
{
      long n1=*s-'0',n2;
      while(*(++s))
      {
            n1*=10;
            n2=*s-'0';
            n1+=n2;
      }
      return n1;
}
```

**תרגיל 4**

```
char make_cap(char c)
{
    if (c >= 'a' && c <= 'z') return c - 'a' + 'A';
    else return c;
}
```

## *10.2. רמה בינונית*

**תרגיל 1**

```
#define TRUE      (1)
#define FALSE     (0)

void analyize_string(char *s)
{

      int bLegalWord, bLegalNumber;
      int WordCounter = 0, NumbersCounter = 0;
      int MaxIllegal = 0, CurIllegal = 0;
      /* while not reach end of string */
      while (*s != '\0')
      {

            /* skip spaces */
            while (*s == ' ') ++s;

            /* if reach end of string, quit */
            if (*s != '\0') break;

            /* if the current letter is capital letter, mark
               the current word is not a number, but might be a word.
            */

            if (*s >= 'A' && *s <= 'Z')
            {
                  bLegalWord = TRUE;
                  bLegalNumber = FALSE;
            }
            /* else if the current letter is a number, mark that it
```

```
            might be number but its not a word */

            else if (*s >= '0' && *s <= '9')
            {
                  bLegalWord = FALSE;
                  bLegalNumber = TRUE;
            }

          /* else this word is not legal */
            else
            {
                  bLegalWord = FALSE;
                  bLegalNumber = FALSE;
            }

            /* read till the end of the word */
            while (*s != ' ' && *s != '\0')
            {
                  /* if the current char is small letter, mark it is
not a number */
                  if (*s >= 'a' && *s <= 'z') bLegalNumber = FALSE;

                  /* if the current char is number, mark it is not
                  word */

                  if (*s >= '0' && *s <= '9') bLegalWord = FALSE;
            }


            /* if its word, increase word counter */
            if (bLegalWord)
            {
                  CurIllegal = 0;
                  WordCounter++;
            }
            /* if its number, increase number counter */
            else if (bLegalNumber)
            {
                  CurIllegal = 0;
                  NumbersCounter++;
            }
            /* else increase illegal counter and check for largest
serias */
            else
            {
                  CurIllegal++;
                  if (CurIllegal > MaxIllegal) MaxIllegal =
                        CurIllegal;
            }
      }

      /* print results */
      /* TASK FOR THE READER :) */
}
```

# 11. רקורסיה

## 11.1. רמה קלה

### תרגיל 1

הפתרון באדיבות שושן כהן (psclil@gmail.com).

```
void tryangle(int n)
{
    int i;
    if (n > 1) tryangle(n-1);
    for (i = 0; i < n; i++) putchar('*');
    putchar('\n');
}
```

### תרגיל 2

```
int azeret(int n)
{
      if (n == 0) return 1;
      return azeret(n-1) * n;
}
```

### תרגיל 3

```
int max_arr(int a[], int n)
{
      int max;
      if (n == 1) return a[0];
      max = max_arr(a+1, n-1);
      if (max > a[0]) return max;
      return a[0];
}
```

### תרגיל 4

```
int fib(int n)
{
	if (n < 2) return n;
	return fib(n-1)+fib(n-2);
}
```

### תרגיל 5

```
int is_possitive(int a[], int n)
{
	if (n == 1) return (a[0] > 0);
	return (a[0] > 0) * is_possitive(a+1,n-1);
}
```

### תרגיל 6

```
void print_reverese(int arr[], int n)
{
	if (n == 1)
	{
		printf("%d ", arr[0]);
		return;
	}

	print_reverese(arr+1, n-1);
	printf("%d ", arr[0]);
}
```

### תרגיל 7

```
void print(int n)
{
	if (n == 1)
	{
		printf("ab");
		return;
	}
	putchar('a');
	print(n-1);
	putchar('b');
}
```

**תרגיל 8**

תשובה: הפונקציה מחזירה את הערך המוחלט של האיבר במערך שערכו המוחלט הוא הגדול ביותר.

## 11.2. רמה בינונית

### תרגיל 1

```
int strlen_rec(char *s)
{
	if (*s == '\0') return 0;
	return strlen_rec(s+1)+1;
}
```

```
char* strcpy_rec(char *str1, char *str2)
{
	if (*str2 == '\0')
	{
		*str1 = *str2;
		return str1;
	}

	strcpy_rec(str1+1, str2+1);
	return str1;
}
```

```
char* strcat_rec(char* str1, char* str2)
{
	if (*str1 != '\0')
	{
		strcat_rec(str1+1, str2);
		return str1;
	}
	*str1 = *str2;
	if (*str2 == '\0') return str1;
	str1[1]='\0';
	strcat_rec(str1+1, str2+1);
	return str1;
}
```

```
char* strcmp_rec(char *str1, char *str2)
{
	if (*str1 == '\0' && *str2 == '\0') return 0;
	if (str1[0] != str2[0]) return str1[0]-str2[0];
	return strcmp_rec(str1+1, str2+1);
}
```

### תרגיל 2

```
void avg_arr(int arr[], int n, double *avg)
{
    if (n == 1)
    {
        *avg = arr[0];
        return;
    }
    avg_arr(arr+1, n-1, avg);
    *avg = (arr[0] + ( *avg * (n-1) )) / n;
}
```

### תרגיל 3

נשים לב שבתרגיל לא הוגדר הערך המוחזר של הפונקציה, ולכן אנחנו יכולים לבחור כל ערך שיהיה לנו נוח.

במקרה של התרגיל הנוכחי המידע שחסר לנו עם התחלת הרקורסיה הוא ערכו של התו האחרון, ולכן נוח לנו להגדיר כי הפונקציה תחזיר לנו את ערכו של התו האחרון, ולהשתמש בהנחה זו בצעד הרקורסיה.

```
char print_last(char* s)
{
      char last;
      if (s[2] == '\0')
      {
            if (s[0] < s[1]) putchar(s[0]);
            return s[1];
      }
      last = print_last(s+1);
      if (s[0] < last) putchar(s[0]);
      return last;
}
```

### תרגיל 4

```
int max_arr(int arr[], int n)
{
      int max;
      if (n == 1) return arr[0];

      max = max_arr(arr+1,n-1);
      if (abs(max) > abs(a[0])) return max;
      return a[0];
}
```

### תרגיל 5

הפונקציה מלווה בתוכנית המציגה את השימוש בה.

```
#include <stdio.h>

int is_prime_helper(int n, int k)
{
	if (n == k) return 1;
	if (n % k == 0) return 0;
	else return is_prime_helper(n, k+1);
}

int is_prime(int n)
{
	if (n < 4) return 1;
	return is_prime_helper(n, 2);
}


int main()
{
	int num;
	printf("Please enter a number: ");
	scanf("%d", &num);
	printf("IsPrime=%d\n", is_prime(num));
	return 0;
}
```

### תרגיל 6

```
#include <stdio.h>

int square_sum (int N)
{
	int temp1,temp2;

	if (N==2 || N==3) return 1;
	if (N==1) return 0;

	temp1 = square_sum(N-1);
	temp2 = square_sum(N-2);

	return temp1*temp1 + temp2*temp2;
}


int square_sum2 (int N)
{
	if (N <= 3) return N-1;
	return square_sum2(N-1) + square_sum(N);
```

```
}


int main()
{
	int val;
	val = square_sum2(7);
	printf ("%d\n",val);
	return 0;
}
```

## 11.3. רמה קשה

### תרגיל 1

```
char between(char *s)
{
	char last, temp;
	if (s[3] == '\0')
	{
		if (s[1] > s[0] && s[1] < s[2]) putchar(s[1]);
		return s[2];
	}

	temp = s[1];
	s[1] = s[0];
	last = between(s+1);
	s[1] = temp;

	if (s[1] < last && s[1] > s[0]) putchar(s[1]);

	return last;
}
```

## תרגיל 2

א.

```
void extreme_to_middle(int a[], int n)
{
	if (n == 1)
	{
		printf("%d\t", a[0]);
		return;
	}
	printf("%d\t%d\t", a[0], a[n-1]);
	extreme_to_middle(a+1, n-2);

	return;
}
```

ב.

```
void middle_to_extreme(int a[], int n)
{
	if (n == 1)
	{
		printf("%d\t", a[0]);
		return;
	}
	middle_to_extreme(a+1, n-2);
	printf("%d\t%d\t", a[0], a[n-1]);

	return;
}
```

## תרגיל 3

```
#include <stdio.h>

int print_series(int n)
{
	int val;
```

```
	if (n == 1)
	{
		printf("%d ", n);
		return 1;
	}

	val = print_series(n-1);

	printf("%d ", n-1+val);
	return n-1+val;
}

int main()
{
	print_series(10);
	return 0;
}
```

הרעיון:

בסיס האינדוקציה: עבור n=1 אנחנו מדפיסים 1 ומסיימים.

עבור n > 1, אנחנו למעשה מוסיפים n-1 לסכום שחושב עד הצעד ה-n-1 (הצעד הקודם), ולכן קודם נחשב (ונדפיס) בצורה רקורסיבית את כל הערכים הקודמים. אנחנו מניחים שהרקורסיה מחזירה לנו את הסכום של האיברים עד האיבר ה-n-1 (כולל), (הסכום נשמר במשתנה val) ולכן כל מה שעלינו לעשות זה להוסיף n-1 לסכום ולהדפיס אותו.

# 12. חיפוש לעומק

## 12.1. רמה קלה

### תרגיל 1

```
int findsum(int a[], int n, int k)
{
        int bSuccess;
        if (n == 0) return 0;
        if (arr[0] == k) return 1;

        bSuccess = findsum(arr+1, n-1, k-arr[0]);
        if (bSuccess) return 1;
        return findsum(arr+1, n-1, k);
}
```

### תרגיל 2

```
#include <stdio.h>

#define N 8
typedef enum { FAIL, SUCCESS } Result;

void PrintResult(int* locations)
{
      int i, j;
      for (i = 0; i < N; ++i)
      {
            for (j = 0; j < N; ++j)
            {
                  if (locations[j] == i) putchar(254);
                  else putchar(176);
            }
            putchar('\n');
      }
      putchar('\n');
}

Result Solve(int* locations, int *cols, int *rows, int *diagonal1,
int *diagonal2, int CurrentPos)
{
      int x;
      if (CurrentPos == N) return SUCCESS;
```

```
	for (x = 0; x < N; ++x)
	{
		if (cols[x] == 0 && rows[CurrentPos] == 0 &&
		diagonal1[x + CurrentPos] == 0 &&
		diagonal2[CurrentPos-x] == 0)
		{
			cols[x] = rows[CurrentPos] =
		diagonal1[x + CurrentPos] = diagonal2[CurrentPos-x] = 1;

			if (Solve(locations, cols, rows, diagonal1,
				diagonal2, CurrentPos+1) == SUCCESS)
			{
				locations[CurrentPos] = x;
				return SUCCESS;
			}
			cols[x] = rows[CurrentPos] =
		diagonal1[x + CurrentPos] = diagonal2[CurrentPos-x] = 0;
		}
	}
	return FAIL;
}

int main()
{
	int Locations[N] = { 0 }, Cols[N] = { 0 }, Rows[N] = { 0 };
	int diagonal1[2*N] = { 0 }, diagonal2[2*N] = { 0 };
	if (Solve(Locations, Cols, Rows, diagonal1, &diagonal2[N], 0)
		== SUCCESS) PrintResult(Locations);
	return 0;
}
```