



# Programming

Arrays



# Example

- Write a program that reads 3 numbers from the user and print them in reverse order.
- How many variables do we need to store the input?
- How could we cope with 100 numbers? 1000?
- We need a “variable” that can store a series of values

# Solution

```
/* get 10 integers from the user and print them in reversed order */
#include <stdio.h>
#define ARRAY_SIZE 10

int main(void)
{
    int i = 0;
    int array[ARRAY_SIZE];

    printf("please enter %d numbers:\n", ARRAY_SIZE);
    for (i = 0; i < ARRAY_SIZE; ++i)
        scanf("%d", &array[i]);

    printf("numbers in reversed order:\n");
    for (i = ARRAY_SIZE - 1; i >= 0; --i)
        printf("%d ", array[i]);
    printf("\n");

    return 0;
}
```



# Exercise @ class

- Write a program that reads 5 numbers from the user and determines if any number was entered more than once.

# Exercise 1– Part A

כיתבו פונקציה בעלת המפרט (prototype) הבא:

```
int how_many(int array[], int size, int value);
```

הפונקציה מקבלת מערך של מספרים שלמים, את גודל המערך, size, ומספר שלם נוסף, value. היא מחזירה את כמות ההופעות של המספר value במערך, כלומר כמה תאים במערך שווים ל-value.

לדוגמא, עבור מערך שערכיו הם 1 2 1 2 1 2 1 12 והמספר 1 יוחזר 3.

# Solution – Part A

```
int how_many(int arr[], int size, int value)
{
    int i, count=0;
    for(i = 0; i < size; i++)
        if (arr[i] == value)
            count++;
    return count;
}
```

# Exercise 1– Part B

- כיתבו פונקציה בעלת המפרט (prototype) הבא:

```
int most_popular(int array[], int size);
```

- הפונקציה מקבלת מערך של מספרים שלמים ואת גודל המערך. עליה להחזיר את המספר הנפוץ ביותר במערך, כלומר המספר שמופיע בו הכי הרבה פעמים (אם יש כמה מספרים עם כמות הופעות זהה, אז יוחזר אחד מהם).
- לדוגמא, עבור מערך שערכיו הם: 4 1 2 3 4 5 4 3 2 3 4 יוחזר 4 (כי הוא מופיע במערך שלוש פעמים, וזה יותר מכל מספר אחר)

# Solution – Part B

```
int most_popular(int array[], int size)
{
    int i, count, max_index, max=0;
    for (i = 0; i < size; i++)
    {
        count = how_many(array, size, array[i]);
        if (count > max)
        {
            max = count;
            max_index = i;
        }
    }
    return array[max_index];
}
```



# Exercise 2

כתבו פונקציה בעלת המפרט הבא:

```
void merge (int arr1 [],int arr2[],int size1 ,int size2,int result[]);
```

הפונקציה מקבלת שני מערכים של מספרים שלמים ממוינים  $arr1$  ו- $arr2$  ושני מספרים שלמים חיוביים שמייצגים את האורכים של  $arr1$  ו- $arr2$ . בנוסף היא מקבלת מערך  $result$  שאורכו הוא סכום אורכי  $arr1$  ו- $arr2$ . הפונקציה מעתיקה את איברי  $arr1$  ו- $arr2$  למערך  $arr3$ , באופן כזה שבסיומה הוא יכיל את איברי שני המערכים בסדר עולה.

**דוגמא:**

מערכים:

1 4 8

2 5 9

תוצאה:

1 2 4 5 8 9

# Solution

```
void merge(int arr1[],int arr2[],int size1,int size2, int result[])
{
    int idx1, idx2, itr;

    for (idx1=0,idx2=0,itr=0; idx1<size1 && idx2<size2; itr++)
    {
        if (arr1[idx1] < arr2[idx2])
            result[itr] = arr1[idx1++];
        else
            result[itr] = arr2[idx2++];
    }

    while (idx1 < size1)
        result[itr++] = arr1[idx1++];

    while (idx2 < size2)
        result[itr++] = arr2[idx2++];
    return;
}
```

# Example 3

- Implement a function that accepts:
  - Two integer matrices
  - Number of rows
- The function returns 1 if they are equal, 0 otherwise
- Assume the matrices are of the same size
- Note that when passing matrices as parameters the second subscript must be specified and it must be constant

# Solution

```
int compare_matrices(int mat1[][SIZE],
                    int mat2[][SIZE], int rows) {
    int i,j;

    /* for each row */
    for(i = 0; i < rows; i++)
        /* for each column */
        for (j=0; j < SIZE; j++)
            /* compare the values */
            if (mat1[i][j] != mat2[i][j])
                /* if not equal, return false */
                return 0;

    /* If we got here, matrices are identical */
    return 1;
}
```

# Example 4 – array vs. primitives as functions arguments

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 3


void f1 (int arr[], int size) {
    int rr = rand() % size;
    arr[rr] = 0;
}

void f2 (int x) {
    x = x + 1;
}

int main(void) {
    int a [] = {1, 2, 3};
    int b = 1;
    f1(a, SIZE);
    f2(b);
    printf(" a = [%d %d %d], b = %d\n", a[0], a[1], a[2], b);

    return 0;
}
```

*Random index in the range 0 – SIZE-1*



# Debug 1: Approximate PI

```
/*This program approximates pi using the formula:
 pi^2/6 = 1/1 + 1/4 + 1/9 + ...*/
#include <stdio.h>
#include <math.h>
int main() {
    int i, num;
    double result = 0.0;
    printf("enter number of terms\n");
    scanf("%d", &num);

    for (i = 1; i <= num; i++);
        result += 1.0 / (i * i);

    printf("Approximated pi = %g\n", sqrt(6*result));
    return 0;
}
```

# Debug 2: One Equal Element

```
#include <stdio.h>
#define SIZE 5
int oneEqualElement (int arr1[], int arr2[], int size) {
    int i;
    for (i = 0; i <= size; ++i)
        if (arr1[i] = arr2[i])
            printf("found an equal element = %d\n",arr1[i]);
            return 1;
    return 0;
}

int main() {
    int a1 [] = {1,2,3,4,5};
    int a2 [] = {6,7,8,9,10};
    int a3 [] = {9,8,7,6,5};
    int a4 [] = {4,3,2,1};
    printf("equal %d\n",oneEqualElement(a1,a2,SIZE)); // false
    printf("equal %d\n",oneEqualElement(a1,a3,SIZE)); // true
    printf("equal %d\n",oneEqualElement(a1,a4,SIZE)); // false
    return 0;
}
```



If Time Allows...loops questions



# Q1

עיינו בקטע הבא וסמנו את כל התשובות הנכונות:

```
int k = 10, sum = 0;
int j;
for (j = 0; j < k; j = k - 1)
    sum = sum + 1;
```

- a. After execution the value of k is 10
- b. This code will fail in compilation
- c. After execution the value of j is 9
- d. There will be an infinite loop
- e. Non of these answers is correct

# Q2

עיינו בקטע הבא וסמנו את כל התשובות הנכונות:

```
int sum = 0,number;  
for (number = 1;number <= 10; number = number + 1)  
    number = number - 1;  
printf(“%d”,sum);
```

- a. This code will fail in compilation
- b. 0 will be printed
- c. 55 will be printed
- d. 45 will be printed
- e. Non of these answers is correct

# Q3

עיינו בקטע הבא וסמנו את כל התשובות הנכונות:

```
int i;  
for (i = 0; i < 10; i = i + 1)  
    for (i = 0; i < 9; i = i + 1)  
        printf("*");
```

- a. 90 '\*' will be printed
- b. There will be an infinite loop
- c. 10 '\*' will be printed
- d. This code will fail in compilation
- e. Non of these answers is correct

# Solution to class exercise

```
#include <stdio.h>
#define SIZE 5

int main()
{
    int itr1, itr2, twice = 0;
    int arr[SIZE];
    printf("Enter %d numbers:\n", SIZE);
    for (itr1 = 0; itr1 < SIZE; itr1++)
    {
        scanf("%d", &arr[itr1]);
        for (itr2 = 0; itr2 < itr1; itr2++)
            if (arr[itr1] == arr[itr2])
                twice = 1;
    }
    if (twice)
        printf("A number was entered more than once\n");

    return 0;
}
```