Structured programming

Exercises 6

Table of Contents

1. Arrays [ ]

1.1. Declaring arrays

```
type variable_name[SIZE];

int a[10];
float x[99];
char c[5];
```

1.2. Accessing array element

```
array[element_index];

int a[10];
a[0] = 1; // assigning value 1 of the first element
printf("%d", a[9]); // printing the value of the last element
```

2. Problems

2.1. Problem 1

Write a program that for two arrays read from SI will check if they are equal. Print out the result from the comparison. The maximum size of arrays is 100.

Solution `p6_1_en.c`

```c
#include<stdio.h>
#define MAX 100
int main() {
    int n1, n2, element, i;
    int a[MAX], b[MAX];
    printf("First array size:  ");
    scanf("%d", &n1);
    printf("Second array size:  ");
    scanf("%d", &n2);
    if (n1 != n2)
        printf("Arrays are equal\n");
    else {
        printf("Elements of the first array: \n");
        for (i = 0; i < n1; ++i) {
            printf("a[%d] = ", i);
            scanf("%d", &a[i]);
        }
        printf("Elements of the second array: \n");
        for (i = 0; i < n2; ++i) {
            printf("b[%d] = ", i);
            scanf("%d", &b[i]);
        }
        // check if arrays are equal:
        for (i = 0; i < n1; ++i)
            if (a[i] != b[i])
                break;
        if (i == n1)
            printf("Arrays are equal\n");
        else
            printf("Arrays are not equal\n");
    }
    return 0;
}
```

2.2. Problem 2

Write a program that for an array read from SI, will compute the sum of even elements, the sum of odd elements and will compute the ratio even/odd.

Example

For array: 3 2 7 6 2 5 1 The program should print:

```
Sum even: 8
Sum odds: 16
Ratio: 0.75
```

Solution `p6_2_en.c`

```c
#include <stdio.h>
#define MAX 100
int main() {
    int i, n, a[MAX], count_even = 0, count_odd = 0, sum_even = 0, sum_odd = 0;
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for (i = 0; i < n; ++i) {
        if (a[i] % 2) {
            count_odd++;
            sum_odd += a[i];
        } else {
            count_even++;
            sum_even += a[i];
        }
    }
    printf("Sum even: %d\nSum odd: %d\n", sum_even, sum_odd);
    printf("Ratio: %.2f\n", (float)count_even / count_odd);
    return 0;
}
```

2.3. Problem 3

Write a program that will compute the scalar product of two vectors with n coordinates. The number of coordinates n and the coordinates are read from SI. Print the result on the SO.

Solution `p6_3_en.c`

```c
#include<stdio.h>
#define MAX 100
int main() {
    int a[MAX], b[MAX], n, i, scalar = 0;
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for (i = 0; i < n; ++i)
        scanf("%d", &b[i]);
    for (i = 0; i < n; ++i)
        scalar += a[i] * b[i];
    printf("The scalar product is: %d\n", scalar);
    return 0;
}
```

2.4. Problem 4

Write a program that will check if a given array with n elements read from SI is ascending, descending or neither. Print the result.

Solution `p6_4_en.c`

```c
#include <stdio.h>
#define MAX 100
int main() {
    int n, element, a[MAX], i;
    short ascending = 1, descending = 1;
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for (i = 0; i < n - 1; ++i) {
        if (a[i] >= a[i + 1]) {
            ascending = 0;
            break;
        }
    }
    for (i = 0; i < n - 1; ++i) {
        if (a[i] <= a[i + 1]) {
            descending = 0;
            break;
        }
    }
    if (!descending && !ascending)
        printf("Array is not ascending and not descending\n");
    else if (descending)
        printf("Array is descending\n");
    else if (ascending)
        printf("Array is ascending\n");
    return 0;
}
```

2.5. Problem 5

Write a program that will rotate elements of an array for one place in right.

Solution `p6_5_en.c`

```c
#include<stdio.h>
#define MAX 100
int main() {
    int n, i;
    int a[MAX];
    scanf("%d", &n);

    for(i = 0; i < n; ++i) {
        scanf("%d", &a[i]);
    }

    int temp = a[n-1];
    for(i = n - 1; i > 0; i--) {
        a[i] = a[i-1];
    }
    a[0] = temp;

    for(i = 0; i < n; ++i) {
        printf("%d ", a[i]);
    }

    return 0;
}
```

2.6. Problem 6

Write a program that will rotate elements of an array for m places in right.

Solution p6_6_en.c

```c
#include<stdio.h>
#define MAX 100
int main() {
    int n, i;
    int a[MAX];
    scanf("%d", &n);

    int m;
    scanf("%d", &m);

    for(i = 0; i < n; ++i) {
        scanf("%d", &a[i]);
    }
    int j;
    for(j = 0; j < m; j++) {
        int temp = a[n-1];
        for(i = n - 1; i > 0; i--) {
            a[i] = a[i-1];
        }
        a[0] = temp;
    }

    printf("\n");

    for(i = 0; i < n; ++i) {
        printf("%d ", a[i]);
    }

    return 0;
}
```

2.7. Problem 7

Write a program that will remove duplicate from an array. After the transformation print the array.

Solution `p6_5_en.c`

```c
#include <stdio.h>
#define MAX 100
int main() {
    int a[MAX], n, i, j, k, removed = 0;
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for (i = 0; i < n - removed; ++i)
        for (j = i + 1; j < n - removed; ++j)
            if (a[i] == a[j]) {
                for (k = j; k < n - 1 - removed; ++k)
                    a[k] = a[k + 1];
                removed++;
                --j;
            }
    n -= removed;
    for (i = 0; i < n; ++i)
        printf("%d\t", a[i]);
    return 0;
}
```

3. Matrices (two dimensional arrays)

3.1. Problem 1

On matrix read from SI compute the difference of sum of elements of odd columns and sum of elements of even rows. Print the result.

Solution `p7_1.c`

```c
#include<stdio.h>
#define MAX 100
int main() {
    int a[MAX][MAX], n, m;
    int i, j, sumCols = 0, sumRows = 0;
    scanf("%d %d", &n, &m);
    for (i = 0; i < n; ++i)
        for (j = 0; j < m; ++j)
            scanf("%d", &a[i][j]);

    for (i = 0; i < n; ++i)
        for (j = 0; j < m; ++j) {
            if ((j + 1) % 2)
                sumCols += a[i][j];
            if (!((i + 1) % 2))
                sumRows += a[i][j];
        }
    printf("%d", sumCols - sumRows);
    return 0;
}
```

3.2. Problem 2

Write a program that for a given matrix read from SI will replace the elements from the main diagonal with the difference between the maximum and minimum element

from the matrix. Print the result matrix.

Solution p7_2.c

```c
#include <stdio.h>
#define MAX 100

int main() {
    int a[MAX][MAX];
    int n;
    scanf("%d", &n);
    int i, j;
    for (i = 0; i < n; ++i) {
        for (j = 0; j < n; ++j) {
            scanf("%d", &a[i][j]);
            if (i == 0 && j == 0) {
                min = max = a[i][j];
            } else if (a[i][j] > max) {
                max = a[i][j];
            } else if (a[i][j] < min) {
                min = a[i][j];
            }
        }
    }

    for (i = 0; i < n; ++i) {
        a[i][i] = max - min;
    }

    for (i = 0; i < n; ++i) {
        for (j = 0; j < n; ++j) {
            printf("%d\t", a[i][j]);
        }
        printf("\n");
    }
    return 0;

}
```

3.3. Problem 3

Write a program that will print on screen if a given matrix is symetric based on the main diagonal. Dimensions and the matrix are read from SI.

Solution `p7_3.c`

```c
#include <stdio.h>
#define MAX 100
int main () {
    int a[MAX][MAX], n, i, j, is_symmetrical = 1;
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        for (j = 0; j < n; ++j)
            scanf ("%d", &a[i][j]);
    for (i = 0; i < n - 1; ++i) {
        for (j = i + 1; j < n; ++j)
            if (a[i][j] != a[j][i]) {
                is_symmetrical = 0;
                break;
            }
        if (!is_symmetrical) break;
    }
    if (is_symmetrical)
        printf("Symmetrical\n");
    else
        printf("Not symmetrical\n");
    return 0;
}
```

4. Source code of the examples and problems

https://github.com/finki-mk/SP/

Source code ZIP