ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Лекція 11.
Операції з одновимірними масивами

Приклад 1. *Знайти добуток елементів масиву, які розташовані між максимальним та мінімальним значенням (включаючи їх).*

-5	**6**	**3**	**-7**	**3**	**1**	**10**	**1**	**-3**	**5**	**8**

1. Згенеруємо масив і заповнимо його випадковими числами.
2. Знайдемо у масиві мінімальне значення, запам’ятаємо його індекс. Зайдемо у масиві максимальне значення, також запам’ятаємо його індекс.
3. Підрахуємо добуток елементів між мінімальним та максимальним значеннями.
4. Виведемо результат на екран.

1) Введемо значення ***n*** з клавіатури та згенеруємо масив

```
#include<conio.h>
#include<stdlib.h>
#include<time.h>
#define MAX_N 100
int main() {
  int n;
  int arr[MAX_N];
  printf("n = ");    scanf("%d", &n);
  srand(time(0));
  if (n < 1 || n > MAX_N) {
     printf("You make a mistake when entering
                    the number n!\n");
     return 0;
  }
```

Заповнимо масив випадковими числами і виведемо його на екран:

```
printf("arr = {");
for (int i = 0; i < n; i++){
    arr[i] = rand() % 21 - 10;
    printf("%d", arr[i]);
    if (i != n - 1)
        printf(", ");
}
printf("}\n");
```

2) Знайдемо мінімальний та максимальний елементи та запам'ятаємо їх індекси

```
  int maxi = 0, mini = 0;
  int min = arr[0], max = arr[0];
  for (int i = 1; i < n; i++) {
        if (arr[i] > max) {
          max = arr[i];
          maxi = i;
        }
        if (arr[i] < min) {
          min = arr[i];
          mini = i;
        }
  }
printf("Min = %d (index = %d)\nMax = %d (index =
%d)\n", min, mini, max, maxi);
```

3) Рахуємо добуток елементів між мінімальним та максимальним

```
double res = 1;
for (int i = mini; i <= maxi; i++)
  res *= arr[i];
```

Але виникне проблема, коли ***mini > maxi****. Тоді не виконається жодної ітерації циклу.*

Якщо ***mini > maxi****, то потрібно їх поміняти місцями.*

3) Рахуємо добуток елементів між мінімальним та максимальним

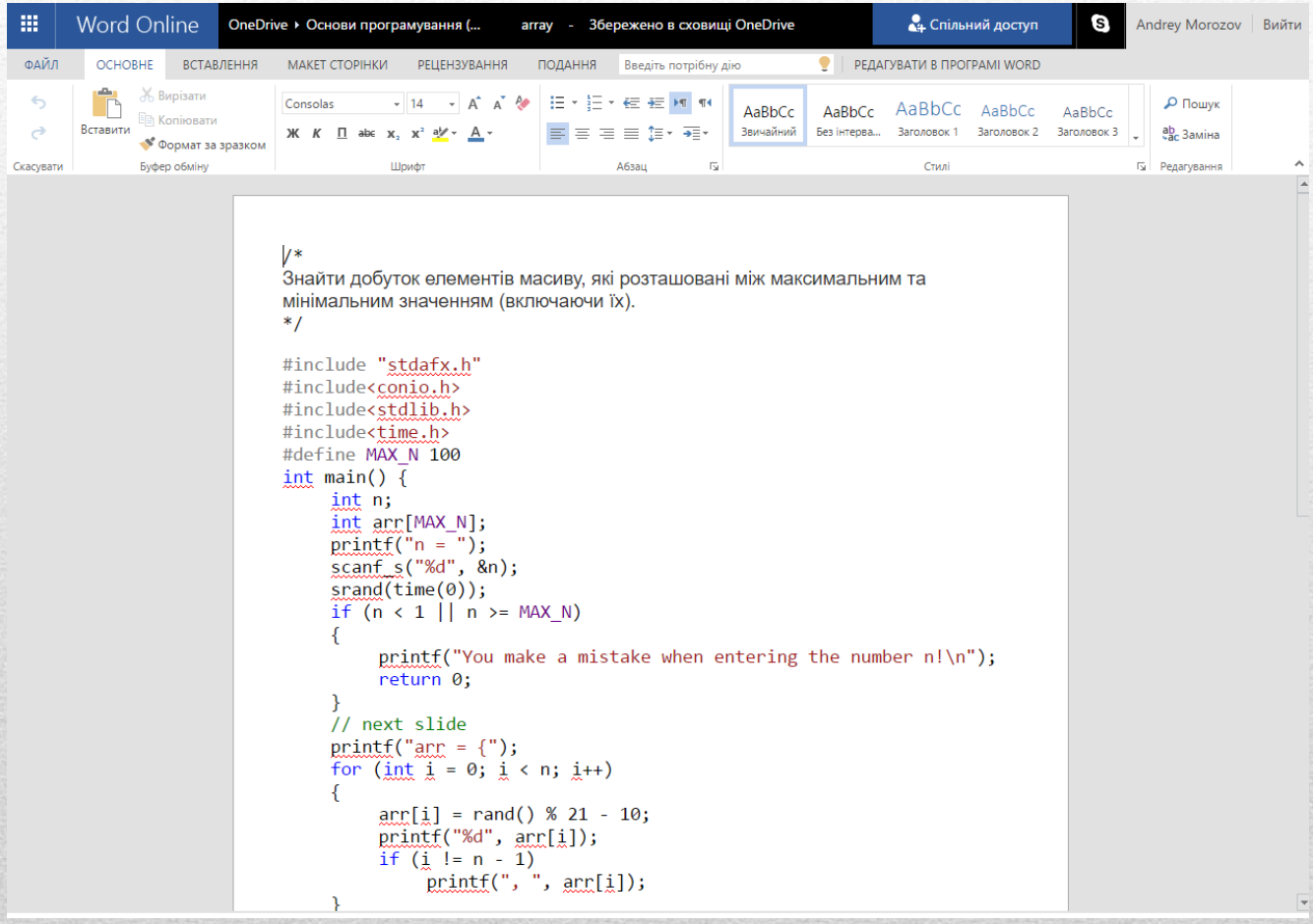
```
int tmp;
if (mini > maxi){
  tmp = mini;
  mini = maxi;
  maxi = tmp;
}
double res = 1;
for (int i = mini; i <= maxi; i++)
  res *= arr[i];
```

4) Виводимо результат:

```
printf("Result = %f\n", res);
```

Даний приклад розміщено в Інтернеті за адресою:

https://1drv.ms/w/s!AvLKc6r1gw0VtXBjqjrUlGlN2KHk



Приклад 2. *Дано масив цілих чисел розміру N. Вивести спочатку всі парні числа в порядку зростання їх індексів, а потім - всі непарні числа в порядку убування їх індексів.*

-5	**6**	**3**	**-7**	**3**	**1**	**10**	**1**	**-3**	**5**	**8**

1. Згенеруємо масив і заповнимо його випадковими числами.
2. Проходячи по масиву, починаючи з 0 елемента, перевіряємо парне число чи ні. Якщо парне виводимо його на екран.
3. Проходячи по масиву, починаючи з n-1 елемента, перевіряємо парне число чи ні. Якщо не парне виводимо його на екран.

```
for (int i = 0; i < n; i++)
    if (arr[i] % 2 == 0)
    printf("%2i\n", arr[i]);
printf("---------\n");
for (int i = n - 1; i >= 0; i--)
    if (arr[i] % 2 != 0)
    printf("%2i\n", arr[i]);
```

```
n = 10
arr = {-9, 6, 10, -6, 6, 10, 3, -6, -7, 6}
 6
10
-6
 6
10
-6
 6
---------
-7
 3
-9
```

Приклад 3. *Дано масив цілих чисел розміру N. Знайти номер його першого локального мінімуму (локальний мінімум - це елемент, який менше будь-якого зі своїх сусідів).*

-5	6	3	-7	3	1	10	1	-3	5	8

1. Згенеруємо масив і заповнимо його випадковими числами.
2. Проходячи по масиву, знаходимо перший локальний мінімум.
3. Виводимо індекс локального мінімуму на екран.

```
for (int i = 1; i < n-1; i++)
    if (arr[i] < arr[i - 1] && arr[i] < arr[i + 1]) {
    printf("%i\n", i); break;
    }
```

```
n = 10
arr = {-2, 8, -5, 2, -6, 4, 1, -10, -9, 2}
2
```

```
int i = 1;
while ((i<n-1) && !((arr[i]<arr[i-1]) && (arr[i]<arr[i+1]))) {
++i;
}
printf("%i\n", i);
```

```
n = 10
arr = {7, 8, 5, 1, 3, -3, -1, 10, -3, 5}
3
```

Приклад 4. *Дано масив A розміру N. Сформувати новий масив B того ж розміру, елементи якого визначаються наступним чином:*

$$B_i = \begin{cases} 2 * A_i, \text{якщо } A_i < 5 \\ \dfrac{A_i}{2}, \text{ інакше} \end{cases}$$

-5	**6**	**3**	**-7**	**3**	**1**	**10**	**1**	**-3**	**5**	**8**

```
n = 10
arr A = {  0,   8,   6,  -2,  -8,  -1,   9,   5,  10,  -1}
arr B = {  0,   4,   3,  -4, -16,  -2,   4,   2,   5,  -2}
```

```c
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define MAX_N 100
int main() {
    int n, arr[MAX_N], arrb[MAX_N];
    printf("n = ");     scanf("%d", &n);
    srand(time(0));
    printf("arr A = {");
    for (int i = 0; i < n; i++) {
        arr[i] = rand() % 21 - 10; printf("%3d", arr[i]);
        if (i != n - 1) printf(", ");
    }
    printf("}\n");
    for (int i = 0; i < n; i++) {
        if (arr[i] < 5) arrb[i] = 2 * arr[i];
        else arrb[i] = arr[i] / 2;
    }
    printf("arr B = {");
    for (int i = 0; i < n; i++){
        printf("%3d", arrb[i]);
        if (i != n - 1) printf(", ");
    }
    printf("}\n");
```

Приклад 5. *Дано масив розміру n. Поміняти місцями його мінімальний і максимальний елементи.*

```
int min = 0, max = 0;
for (int i = n - 1; i >= 0; --i) {
    if (arr[i] > arr[max]) max = i;
    if (arr[i] < arr[min]) min = i;
}
if (max != min) {
    arr[max] += arr[min];
    arr[min] = arr[max] - arr[min];
    arr[max] = arr[max] - arr[min];
}
printf("arr A = { ");
for (int i = 0; i < n; ++i) {
    printf("%3d", arr[i]);
    if (i != n - 1)printf(", ");
}
printf("}\n");
```

```
n = 10
arr = {13, 14, 19, 12, 18, 15, 14, 20, 15, 21}
maxInd = 9, minInd =3
arr = {  13,  14,  19,  21,  18,  15,  14,  20,  15,  12}
Для продолжения нажмите любую клавишу . . .
```

Приклад 6. *Дано масив цілих чисел розміру N. Видалити з масиву всі непарні числа і вивести розмір отриманого масиву і його вміст..*

```
int k = 0;
for (int i = 0; i < n; i++) {
    if (arr[i] % 2 == 0) {
    arr[k] = arr[i];
    ++k;
    }
}
printf("arr = {");
for (int i = 0; i < k; i++) {
       printf("%3d", arr[i]);

       if (i != n - 1) printf(", ");
}
printf("}\n");
```

```
n = 15
arr = { 24,  11,  16,  24,  13,  21,  22,  13,  15,  22,  22,  13,  24,  12,  22}
arr = { 24,  16,  24,  22,  22,  22,  24,  12,  22, }
Для продолжения нажмите любую клавишу . . .
```