

ЛАБОРАТОРНА РОБОТА № 11

Варіант 2

Одновимірні масиви

Мета : дослідити функції генерування послідовностей випадкових чисел, ознайомитись з можливостями створення одновимірних масивів і навчитись обробляти елементи масивів.

Хід роботи:

Завдання 1: Написати програму, що реалізує десять генераторів псевдовипадкових чисел. Кожний генератор викликати за допомогою меню, яке реагує на введення цілого числа: 1, ... , 10.

1	[-4; -1)	6	[-7,85; 28,3]
2	[100; 299]	7	[-100; 100]
3	[-35; -1] парні	8	[23; 71]
4	[-128; 127]	9	[0; 2)
5	[-7; 13) непарні	10	[$\sqrt{17}$; $\sqrt{82}$)

#de-

```
fine _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <windows.h>
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <stdlib.h>
#include <ctime>
```

```
int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(0));
```

```
    int x, i, n;
    double z;
    for (;;) {
        scanf_s("%d", &i);
        switch (i)
        {
            case 1:
                x = -4 + rand() % 3;
                printf("->%d\n", x);
                break;
```

					ДУ«Житомирська політехніка».21.121.02.000–Лр11			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Маньківський В.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Чижмотря О.В.						1
Керівник								6
Н. контр.							ФІКТ Гр. ВТ-21-1[2]	
Зав. каф.								

```

case 2:
    x = 100 + rand() % 200;
    printf("->%d\n", x);
    break;
case 3:
    x = -35 + rand() % 35;
    while (x % 2) {
        x = -35 + rand() % 35;
    }
    printf("->%d\n", x);
    break;
case 4:
    x = -128 + rand() % 256;
    printf("->%d\n", x);
    break;
case 5:
    x = -7 + rand() % 20;
    while (!(x % 2)) {
        x = -7 + rand() % 20;
    }
    printf("->%d\n", x);
    break;
case 6:
    z = (double)rand() * (28.3 + 7.85) / RAND_MAX - 7.85;
    printf("->%.2lf\n", z);
    break;
case 7:
    x = -100 + rand() % 201;
    printf("->%d\n", x);
    break;
case 8:
    x = 23 + rand() % 48;
    printf("->%d\n", x);
    break;
case 9:
    x = 0 + rand() % 2;
    printf("->%d\n", x);
    break;
case 10:
    x = 17 + rand() % 65;
    printf("->%d^(1/2)\n", x);
    break;
}
}
return 0;
}

```

		Маньківський В.			ДУ «Житомирська політехніка».21.121.02.000 – Лр11	Арк.
		Чижмоторя О.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

1
->-3
2
->123
3
->-18
4
->-125
5
->9
6
->13.95
7
->59
8
->57
9
->1
10
->59^(1/2)

```

Завдання 2: Числа m , n і k ($3 \leq k \leq 10$) вводяться з клавіатури. Згенерувати і вивести на екран m цілих випадкових чисел з проміжку, вказаному у пункті а, та n дійсних чисел (виводити на екран з вказаною точністю) з проміжку, вказаному у пункті б. Виведення на екран здійснювати по k чисел у рядку.

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <windows.h>
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <stdlib.h>
#include <ctime>

```

```

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(0));

    int array[9999], m, k, n;
    printf("Кількість чисел (а) - ");
    scanf_s("%d", &m);
    printf("Кількість чисел (б) - ");
    scanf_s("%d", &n);
    printf("Кількість чисел в рядку - ");
    scanf_s("%d", &k);
    printf("a");
    for (; m > 0; m--) {
        if (k < 3 || k > 10) {
            break;
        }
        if (!(m % k)) {
            printf("\n");
        }
        array[m] = 13 + rand() % 387;
        printf(" ");
        printf("%d", array[m]);
    }
}

```

		Маньківський В.			ДУ «Житомирська політехніка».21.121.02.000 – Лр11	Арк.
		Чижмоторя О.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}
printf("\n-----\n");
double arr[9999], x;
printf("6)\n");
for (; n > 0; n--) {
    if (k < 3 || k > 10) {
        break;
    }
    if (!(n % k)) {
        printf("\n");
    }
    arr[n] = -2.0 + rand() % 5;
    x = 1 + rand() % 10;
    arr[n] = arr[n] / x;
    printf(" ");
    printf("%0.11f", arr[n]);
}
return 0;
}

```

```

Кількість чисел (a) - 12
Кількість чисел (6) - 18
Кількість чисел в рядку - 6
a)
 34 107 165 129 310 242
234 350 233 86 30 183
-----
6)

-0.3 0.1 0.2 0.0 0.0 0.5
0.0 0.3 0.0 0.0 -0.1 0.2
-0.3 2.0 -0.3 0.2 0.3 -0.1

```

Завдання 3: В одновимірному масиві, що складається з n цілих елементів, обчислити: 1) суму додатних елементів масиву; 2) добуток елементів масиву, розташованих між максимальним за модулем і мінімальним за модулем елементами.

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <windows.h>
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <stdlib.h>
#include <ctime>

```

```

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(0));

    int array[9999], n, a, b, sum = 0, c, max = 0, min = 9999, nmax = 0, nmin = 9999, dob;
    printf("Кількість чисел - ");
    scanf_s("%d", &n);
    printf("Мінімальне число - ");
    scanf_s("%d", &a);
    printf("Максимальне число - ");
    scanf_s("%d", &b);
    c = n;
    for (; n > 0; n--) {
        if (a > b) {

```

		Маньківський В.			ДУ «Житомирська політехніка».21.121.02.000 – Лр11	Арк.
		Чижмоторя О.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        break;
    }

    arrey[n] = a + rand() % (b - a + 1);
    printf(" ");
    printf("%d", arrey[n]);
}
n = c;
for (; n > 0; n--) {

    if (arrey[n] > 0) {
        sum = sum + arrey[n];
    }
}
n = c;
for (; n > 0; n--) {
    if (max <= abs(arrey[n]))
    {
        max = arrey[n];
        nmax = n;
    }
}
n = c;
for (; n > 0; n--) {
    if (min >= abs(arrey[n]))
    {
        min = arrey[n];
        nmin = n;
    }
}
if (nmax > nmin) {
    for (dob = 1; nmax > nmin; nmax--) {
        dob = dob * arrey[nmax];
    }
}
else {
    for (dob = 1; nmax <= nmin; nmin--) {
        dob = dob * arrey[nmin];
    }
    dob = dob * arrey[nmin + 2];
}
printf("\n Сума %d\n", sum);
printf("Добуток %d\n", dob);

return 0;
}

```

```

Кількість чисел - 15
Мінімальне число - 3
Максимальне число - 9
6 7 3 3 8 6 5 7 6 6 7 9 4 4 7
Сума 88
Добуток 80015040

```

Завдання 4: Використовуючи два заданих масиви X[10] та Y[10] створити масив Z[10], елементи якого обчислюються наступним чином: Z[1]=X[1]+Y[10], Z[2]= X[2]+Y[9],..., Z[10]= X[10]+Y[1] Вивести масив Z та його максимальний елемент.

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <windows.h>

```

		Маньківський В.			ДУ «Житомирська політехніка».21.121.02.000 – Лр11	Арк.
		Чижмоторя О.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

#include<stdio.h>
#include<math.h>
#include <conio.h>
#include <stdlib.h>
#include <ctime>

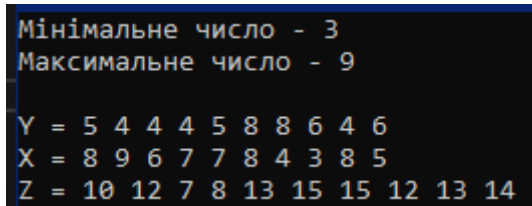
int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(0));

    int x[10], n, a, b, y[10], z[10], c = 9;
    printf("Мінімальне число - ");
    scanf_s("%d", &a);
    printf("Максимальне число - ");
    scanf_s("%d", &b);
    printf("\nY =");
    for (n = 0; n < 10; n++) {

        x[n] = a + rand() % (b - a + 1);
        printf(" ");
        printf("%d", x[n]);
    }
    printf("\nX =");
    for (n = 0; n < 10; n++) {

        y[n] = a + rand() % (b - a + 1);
        printf(" ");
        printf("%d", y[n]);
    }
    printf("\nZ =");
    for (n = 0; n < 10; n++) {
        printf(" ");
        z[n] = x[n] + y[c--];
        printf("%d", z[n]);
    }
    return 0;
}

```



```

Мінімальне число - 3
Максимальне число - 9

Y = 5 4 4 4 5 8 8 6 4 6
X = 8 9 6 7 7 8 4 3 8 5
Z = 10 12 7 8 13 15 15 12 13 14

```

Висновки: я дослідив функції генерування послідовностей випадкових чисел, ознайомитись з можливостями створення одновимірних масивів і навчитись обробляти елементи масивів.

		Маньківський В.			ДУ «Житомирська політехніка».21.121.02.000 – Лр11	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6