**ЛАБОРАТОРНА РОБОТА № 11**

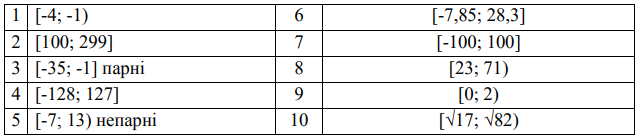
**Варіант 2**

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.

#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 x, i, n;

double z;

for (;;) {

scanf\_s("%d", &i);

switch (i)

{

case 1:

x = -4 + rand() % 3;

printf("->%d\n", x);

break;

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);

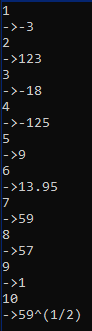
break;

}

}

return 0;

}



Завдання 2: Числа m, n і k (3≤k≤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 arrey[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("а)");

for (; m > 0; m--) {

if (k < 3 || k > 10) {

break;

}

if (!(m % k)) {

printf("\n");

}

arrey[m] = 13 + rand() % 387;

printf(" ");

printf("%d", arrey[m]);

}

printf("\n-----------------\n");

double arr[9999], x;

printf("б)\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;

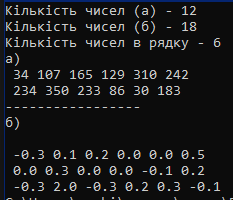
printf(" ");

printf("%.1lf", arr[n]);

}

return 0;

}



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 arrey[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) {

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];

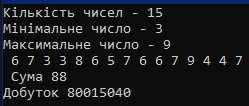
}

printf("\n Сума %d\n", sum);

printf("Добуток %d\n", dob);

return 0;

}



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>

#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(" ");

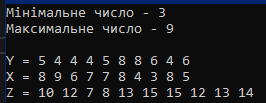
z[n] = x[n] + y[c--];

printf("%d", z[n]);

}

return 0;

}



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