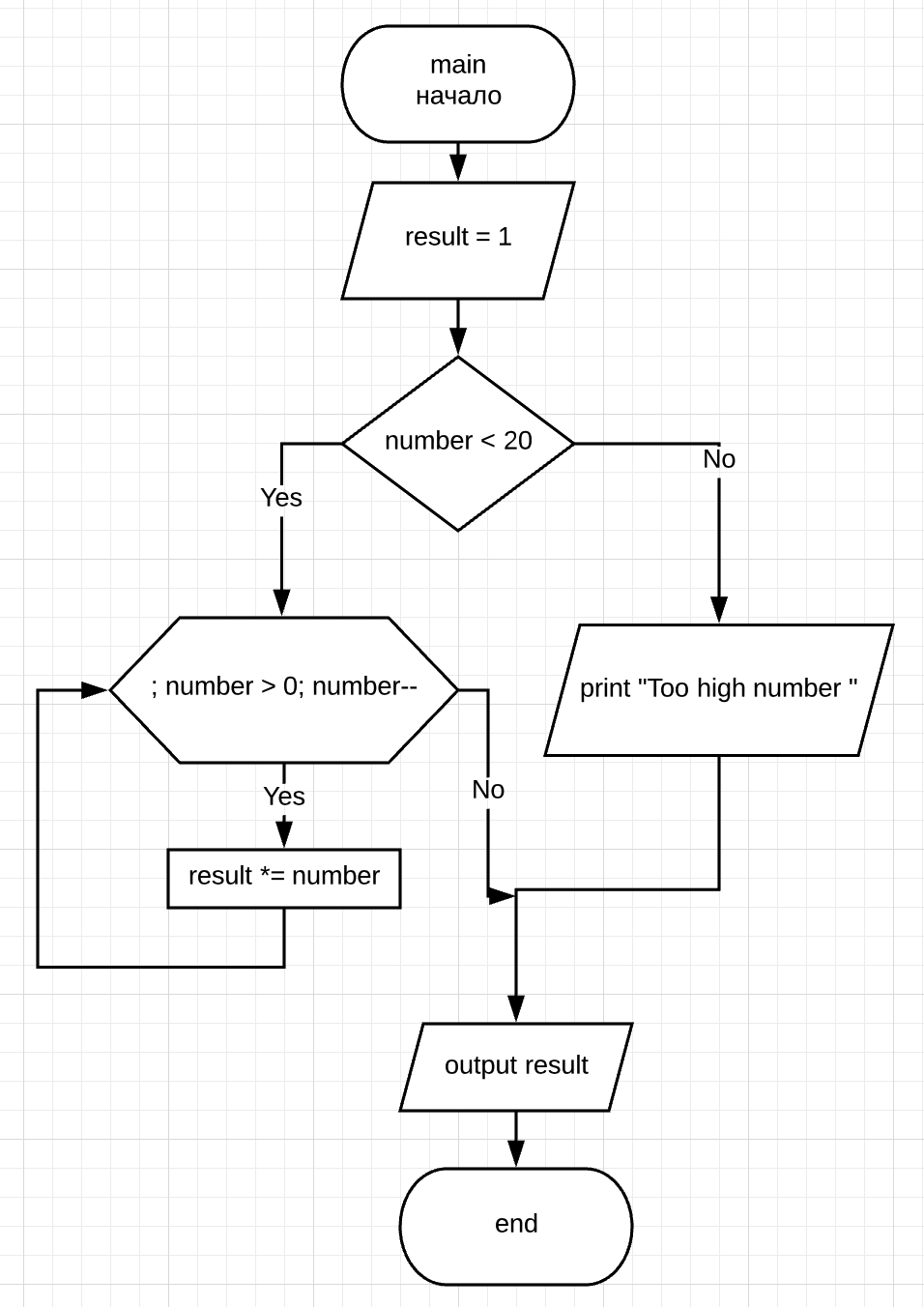
# Дано натуральне число N. Знайти N!. Множення всіх натуральних чисел до N включно



#include <iostream>

using namespace std;

long long factorial(int number) {

long long result = 1;

if(number < 20)

for(; number > 0; number--)

result \*= number;

else

cout << "Too high number!" << endl;

return result;

}

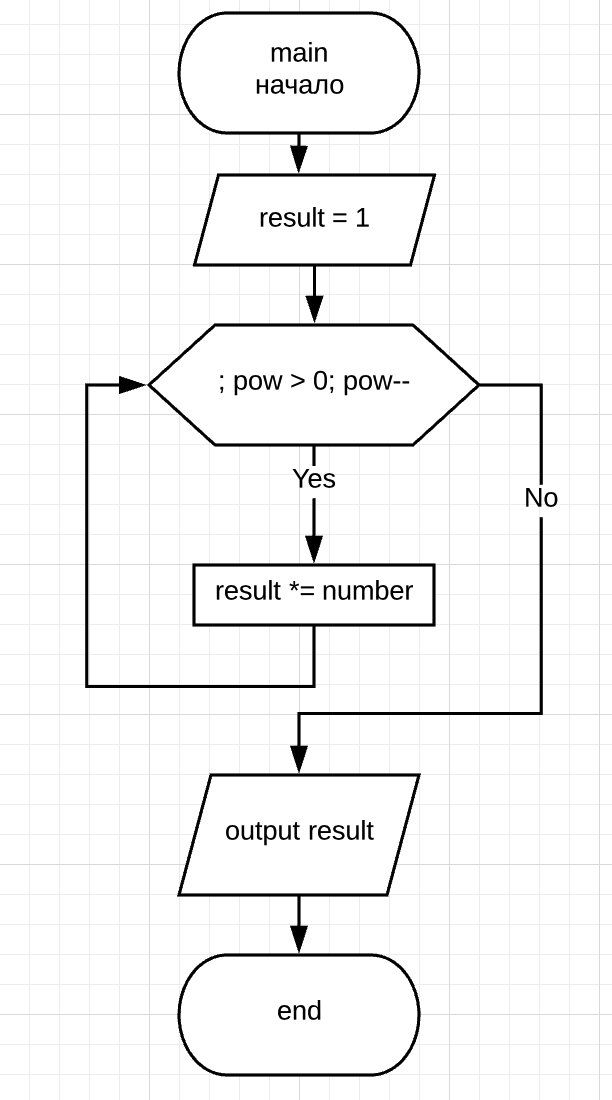
int main() {

cout << factorial(21) << endl;

return 0;

}

# Возвести число в степінь



#include <iostream>

using namespace std;

long long pow(int number, int pow) {

long long result = 1;

for(; pow > 0; pow--)

result \*= number;

return result;

}

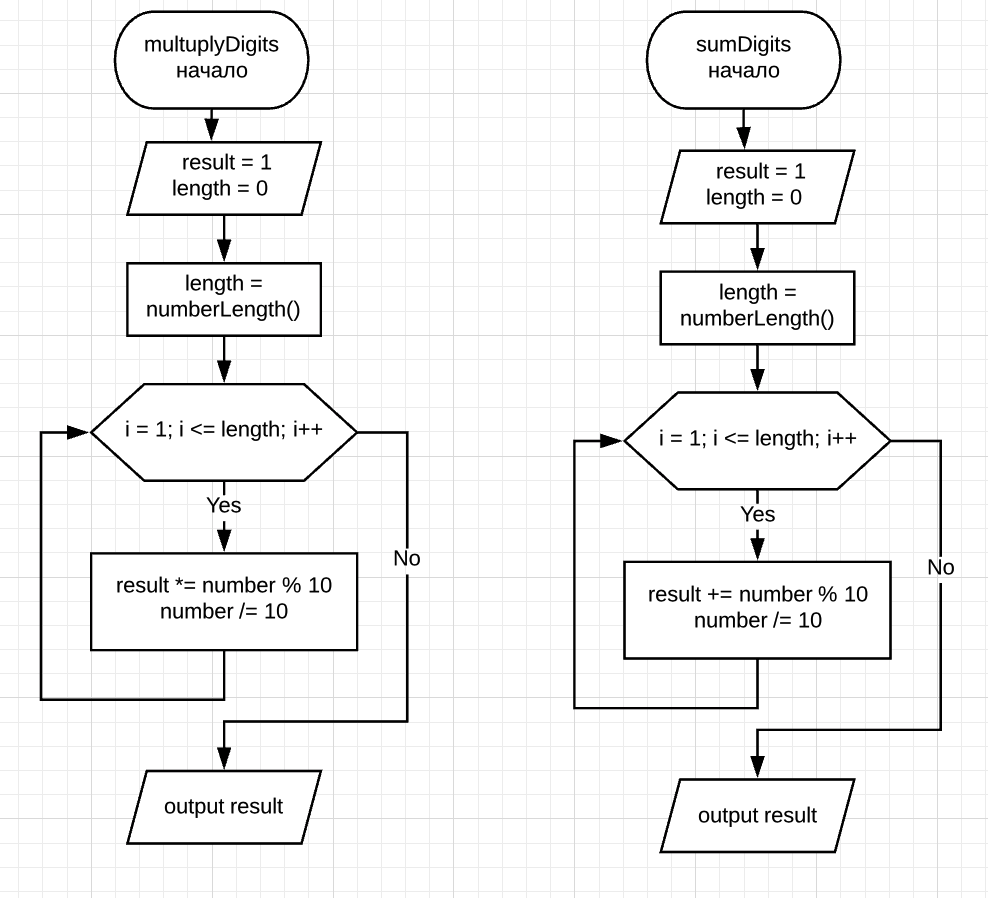
int main() {

cout << pow(2, 5) << endl;

return 0;

}

Дано число. Знайти суму та добуток його цифр



#include <iostream>

using namespace std;

long long pow(int number, int pow) {

long long result = 1;

for(; pow > 0; pow--) result \*= number;

return result;

}

int numberLength(int number) {

int result = 0;

if(number == 0) result = 1;

while(number != 0) {

number /= 10;

result++;

}

return result;

}

int multiplyDigits(int number) {

int result = 1;

int length = numberLength(number);

for(int i = 1; i <= length; i++) {

result \*= number % 10;

number /= 10;

}

return result;

}

int sumDigits(int number) {

int result = 0;

int length = numberLength(number);

for(int i = 1; i <= length; i++) {

result += number % 10;

number /= 10;

}

return result;

}

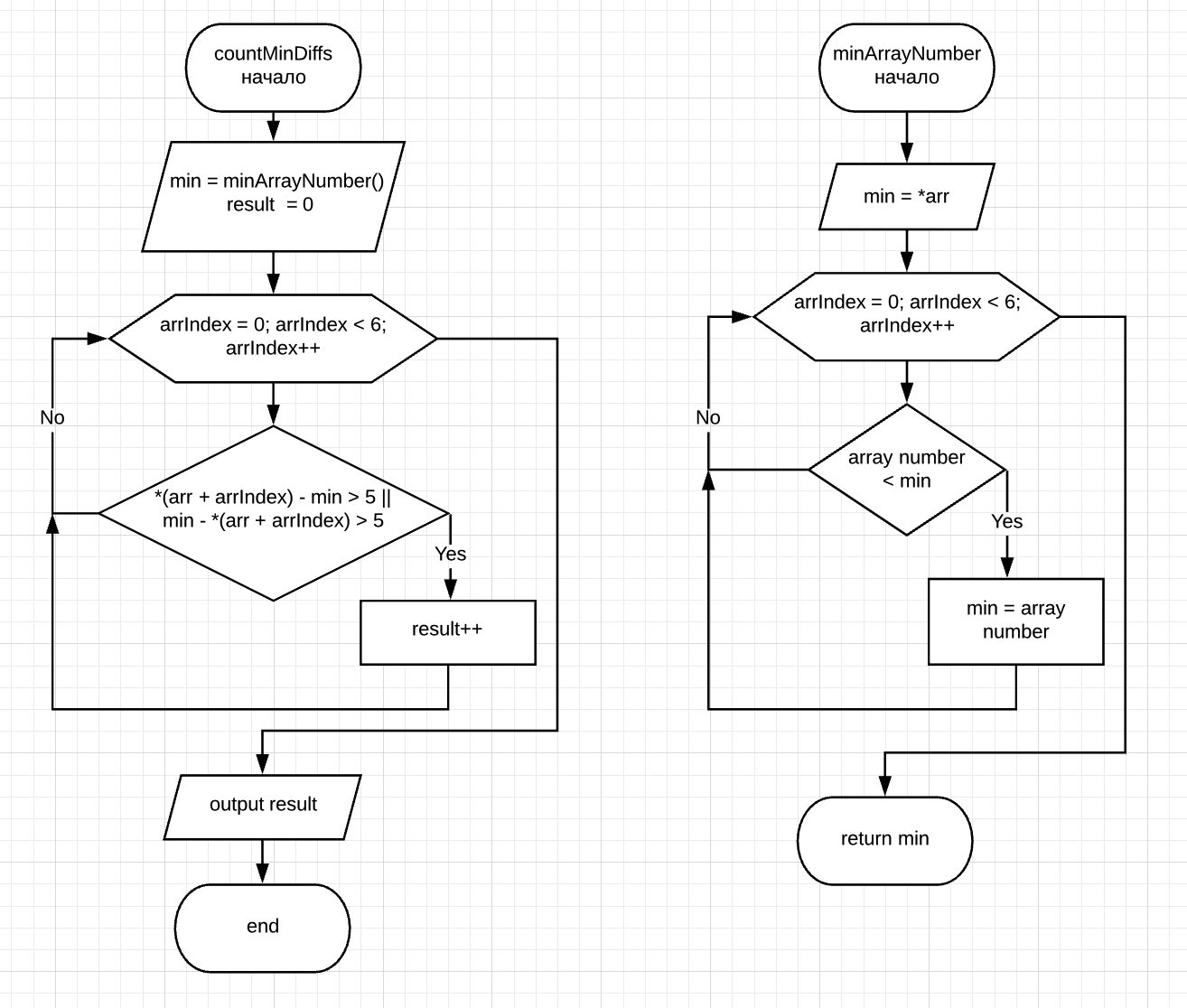
int main() {

cout << multiplyDigits(143) << endl;

cout << sumDigits(143) << endl;

}

# Знайти кількість елементів у масиві, які відрізняються від мінімального на 5



#include <iostream>

using namespace std;

int minArrayNumber(int\* arr) {

int min = \*arr;

for(int arrIndex = 0; arrIndex < 6; arrIndex++)

if(\*(arr + arrIndex) < min)

min = \*(arr + arrIndex);

return min;

}

int countMinDiffs(int\* arr) {

int min = minArrayNumber(arr);

int result = 0;

for(int arrIndex = 0; arrIndex < 6; arrIndex++)

if(\*(arr + arrIndex) - min > 5 || min - \*(arr + arrIndex) > 5)

result++;

return result;

}

int main() {

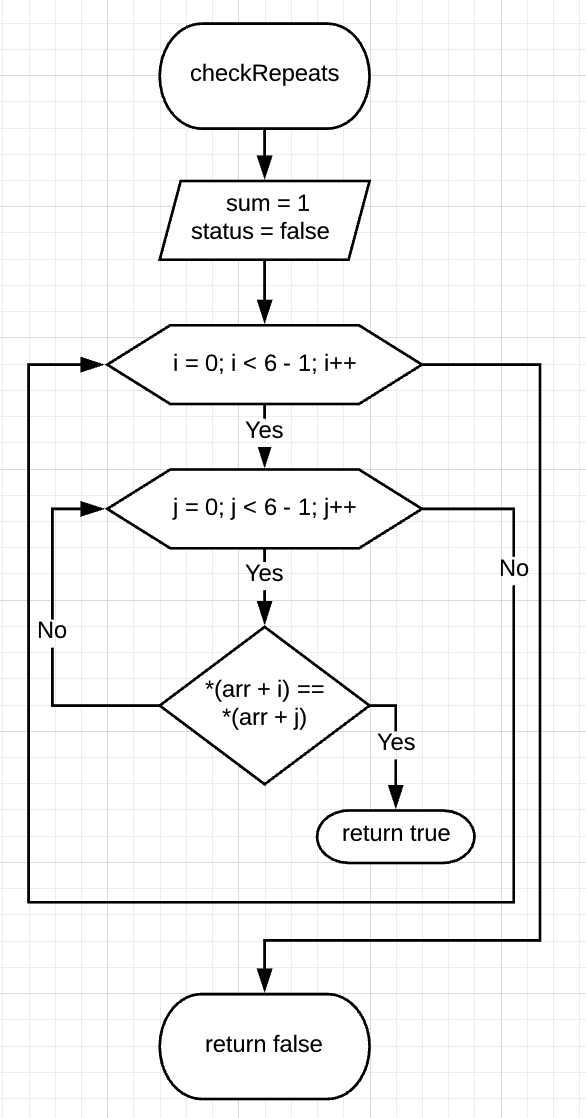
int arr[] = {32, -5, 3, 9, 12, 43};

cout << countMinDiffs(arr) << endl;

return 0;

}

# Дано масив цілих чисел. Знайти, чи є в ньому повторення



#include <iostream>

using namespace std;

bool checkRepeats(int\* arr) {

int sum = 1;

for(int i = 0; i < 6 - 1; i++) {

for(int j = i + 1; j < 6; j++) {

if(\*(arr + i) == \*(arr + j)) {

return true

}

}

}

return false;

}

int main() {

int arr[] = {1, 2, 6, 3, 2, 5};

if(checkRepeats(arr))

cout << "Founded repeats" << endl;

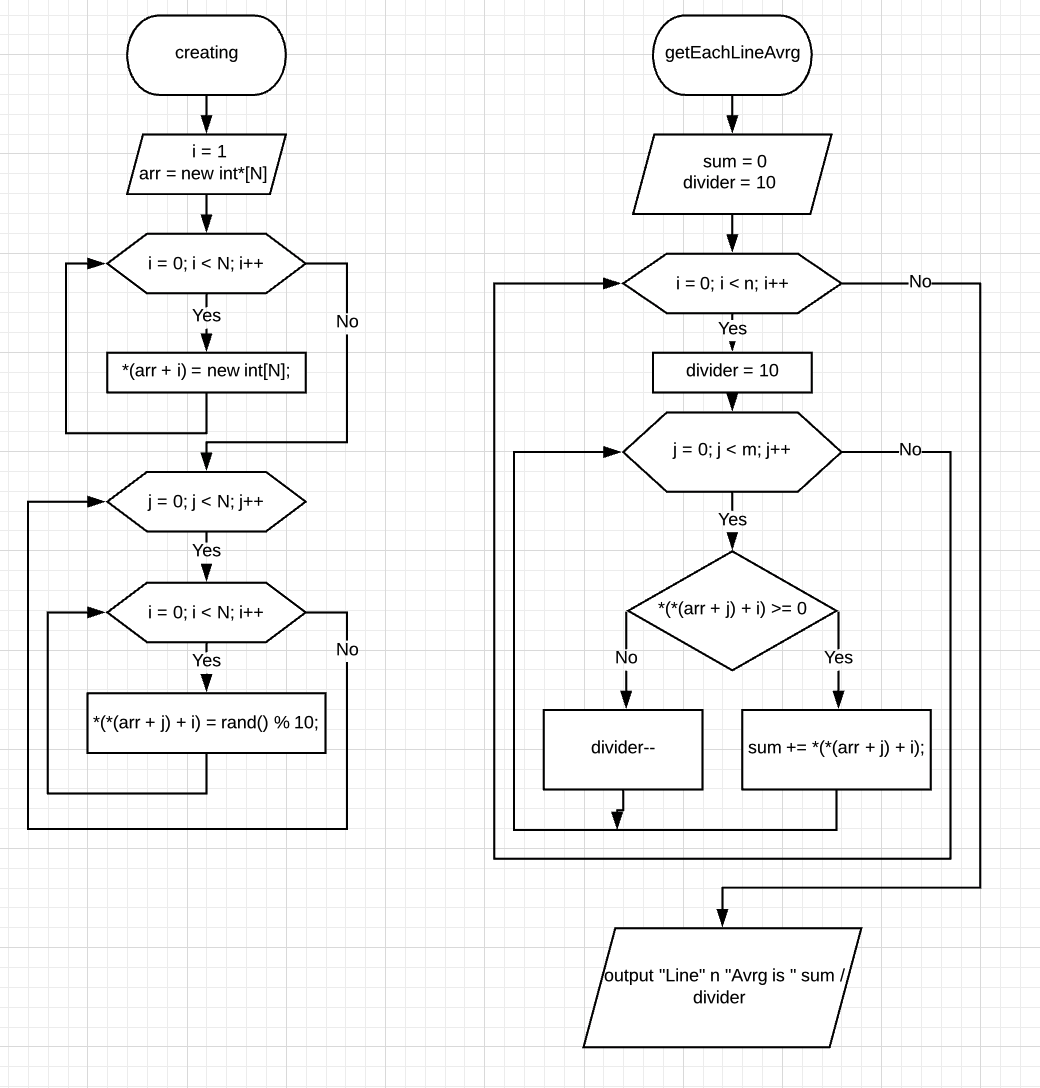
else

cout << "All numbers are unique" << endl;

return 0;

}

# Знайти середнє арифметичне значення невід’ємних елементів кожної строки матриці



#include <iostream>

using namespace std;

int\*\* creating(int N) {

int i;

int\*\* arr = new int\* [N];

for (i = 0; i < N; i++) {

\*(arr + i) = new int[N];

}

for (int j = 0; j < N; j++) {

for (i = 0; i < N; i++) {

\*(\*(arr + j) + i) = rand() % 10;

}

}

return arr;

}

void getEachLineAvrg(int\*\* arr, int n, int m) {

int sum = 0;

int divider = 10;

for(int i = 0; i < n; i++) {

divider = 10;

for(int j = 0; j < m; j++) {

if(\*(\*(arr + j) + i) >= 0)

sum += \*(\*(arr + j) + i);

else

divider--;

}

cout << "Line " << i + 1 << " Avrg is - "<< sum / divider << endl;

sum = 0;

}

}

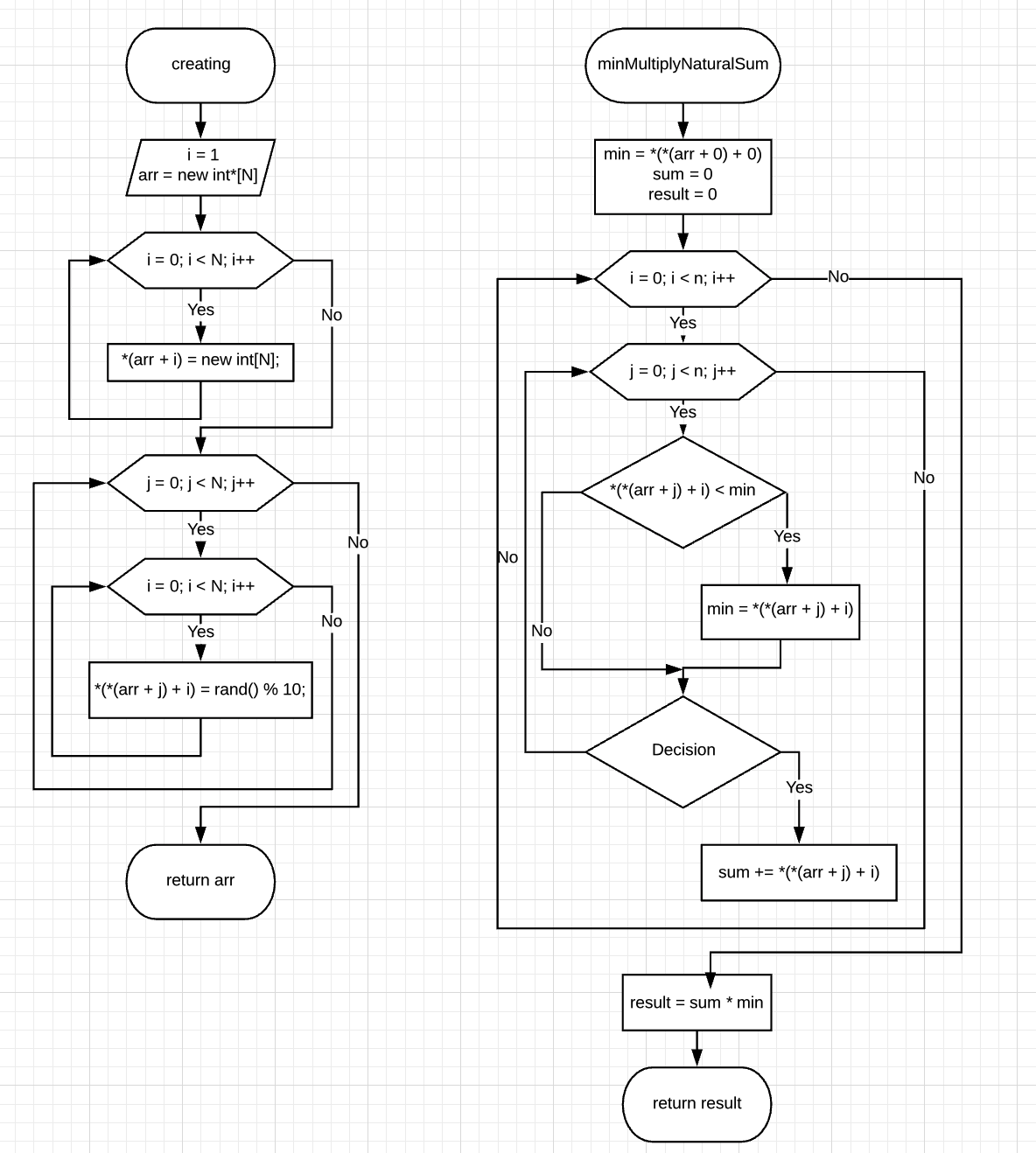
int main() {

int \*\*arr= creating(10);

getEachLineAvrg(arr, 10, 10);

}

# Знайти добуток мінімального елемента матриці на суму її додатних елементів



#include <iostream>

using namespace std;

int\*\* creating(int N) {

int i;

int\*\* arr = new int\* [N];

for (i = 0; i < N; i++) {

arr[i] = new int[N];

}

for (int j = 0; j < N; j++) {

for (i = 0; i < N; i++) {

\*(\*(arr + j) + i) = rand() % 20 - 10;

}

}

return arr;

}

int minMultiplyNaturalSum(int\*\* arr) {

int min = \*(\*(arr));

int sum = 0;

int result = 0;

for(int i = 0; i < n; i++) {

for(int j = 0; j < n; j++) {

if(\*(\*(arr + j) + i) < min) {

min = \*(\*(arr + j) + i);

}

if(\*(\*(arr + j) + i) > 0) {

sum += \*(\*(arr + j) + i);

}

}

}

result = sum \* min;

return result;

}

int main() {

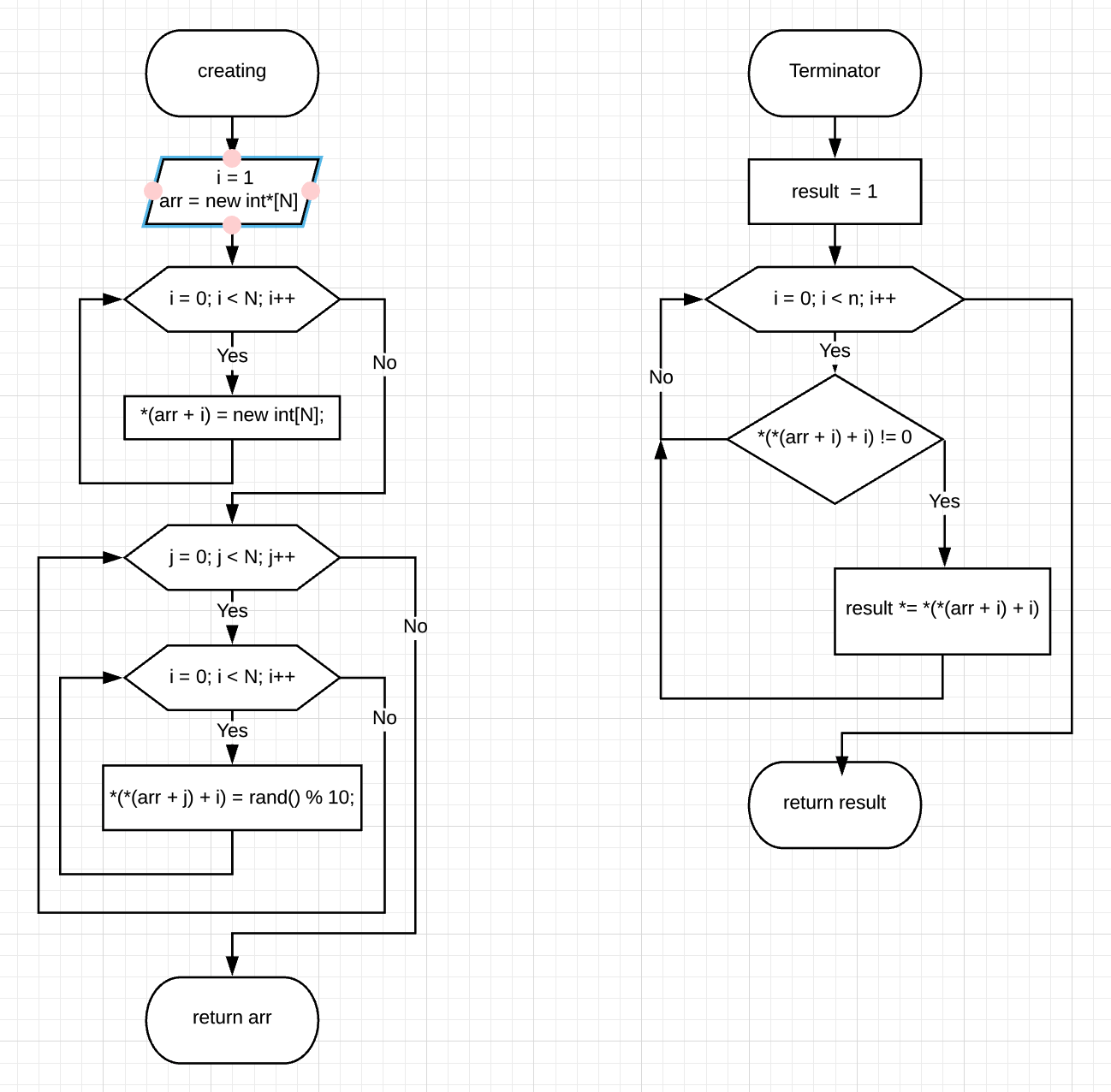
int \*\*arr= creating(10);

cout << "Min number multiply on sum is - " << minMultiplyNaturalSum(arr) << endl;

return 0;

}

# У квадратній матриці знайти добуток ненульових елементів діагоналі



#include <iostream>

using namespace std;

int\*\* creating(int N) {

int i;

int\*\* arr = new int\* [N];

for (i = 0; i < N; i++) {

arr[i] = new int[N];

}

for (int j = 0; j < N; j++) {

for (i = 0; i < N; i++) {

\*(\*(arr + j) + i) = rand() % 10;

}

}

return arr;

}

int diagonalMultiply(int\*\* arr) {

int result = 1;

for(int i = 0; i < n; i++) {

if(\*(\*(arr + i) + i) != 0)

result \*= \*(\*(arr + i) + i);

}

return result;

}

int main() {

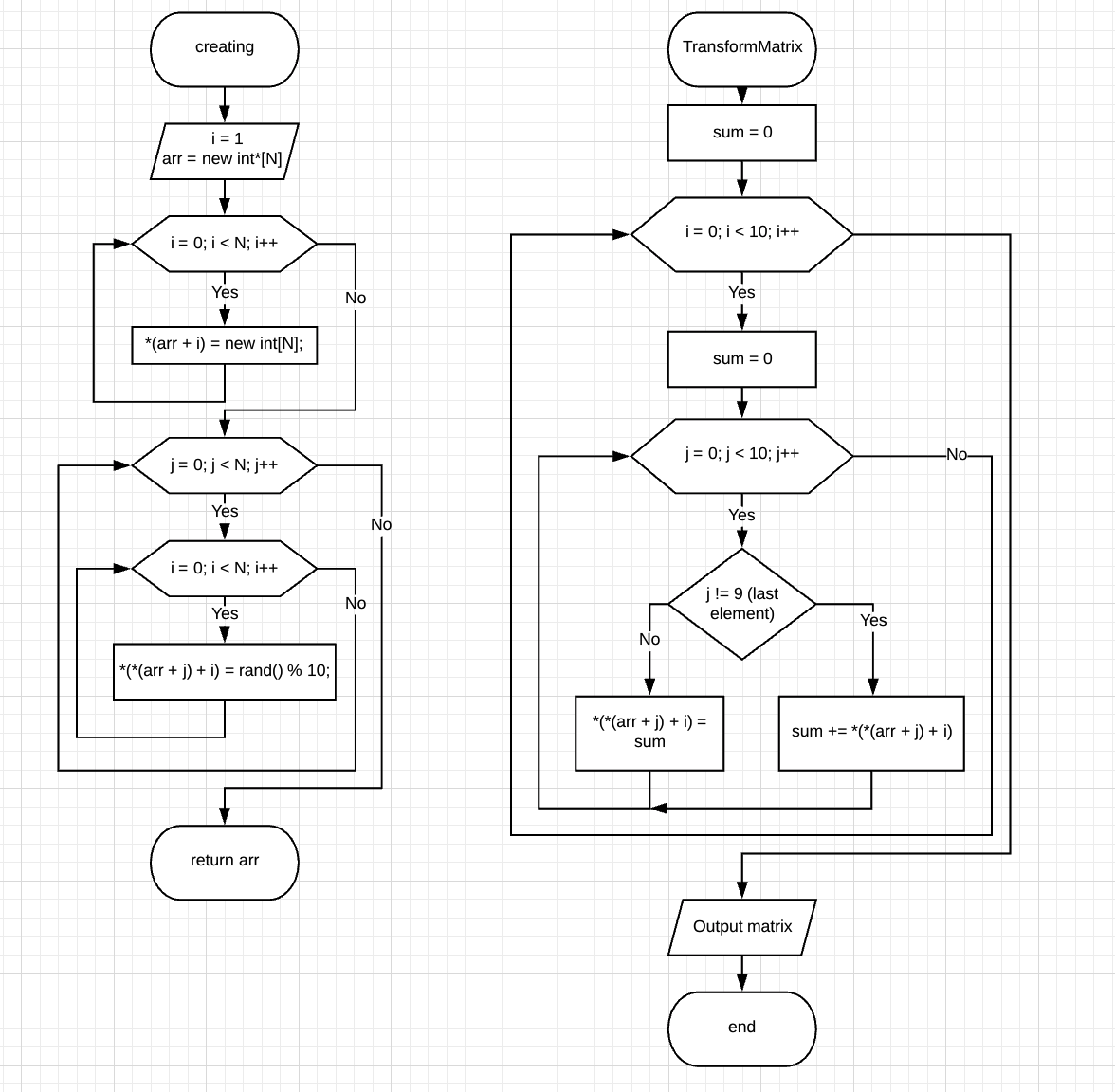
int \*\*arr= creating(10);

cout << "Multiply of non NULL diagonal numbers is - " << diagonalMultiply(arr) << endl;

return 0;

}

Останніх елементи матриці кожного ряду, замінити на суму елементів ряду.



#include <iostream>

using namespace std;

int\*\* creating(int N) {

int i;

int\*\* arr = new int\* [N];

for (i = 0; i < N; i++) {

arr[i] = new int[N];

}

for (int j = 0; j < N; j++) {

for (i = 0; i < N; i++) {

\*(\*(arr + j) + i) = rand() % 10;

}

}

return arr;

}

void transformMatrix(int\*\* arr) {

int sum = 0;

for(int i = 0; i < 10; i++) {

sum = 0;

for(int j = 0; j < 10; j++) {

if(j != 9) {

sum += \*(\*(arr + j) + i);

}

else {

\*(\*(arr + j) + i) = sum;

}

}

}

}

void printMatrix(int\*\* arr) {

for(int i = 0; i < 10; i++) {

for(int j = 0; j < 10; j++) {

cout << \*(\*(arr + j) + i) << " ";

}

cout << endl;

}

}

int main() {

int \*\*arr= creating(10);

transformMatrix(arr);

printMatrix(arr);

return 0;

}