№1

#include <iostream>

using namespace std;

void printTime(int hour, int minute, int second)

{

cout << hour << ":" << minute << ":" << second << endl;

}

void input(int& hour, int& minute, int& second, bool beginning)

{

if (beginning)

{

cout << "Введите время начала промежутка.\n";

}

else

{

cout << "Введите время конца промежутка.\n";

}

cout << "Час: \n";

cin >> hour;

if (hour > 23 || hour < 0)

{

cout << "Можно ввести значение только в пределах от 0 до 23.\n";

input(hour, minute, second, beginning);

return;

}

cout << "Минута: \n";

cin >> minute;

if (minute > 59 || minute < 0)

{

cout << "Можно ввести значение только в пределах от 0 до 59.\n";

input(hour, minute, second, beginning);

return;

}

cout << "Секунда: \n";

cin >> second;

if (second > 59 || second < 0)

{

cout << "Можно ввести значение только в пределах от 0 до 59.\n";

input(hour, minute, second, beginning);

return;

}

}

int main()

{

setlocale(LC\_ALL,"Russian");

int hour1, minute1, second1;

int hour2, minute2, second2;

input(hour1, minute1, second1, 1);

input(hour2, minute2, second2,0);

printTime(hour1, minute1, second1);

printTime(hour2, minute2, second2);

int durationHour, durationMinute, durationSecond;

if (hour1 <= hour2) {

durationHour = hour2 - hour1;

}else{

durationHour = 24 - (hour1 - hour2);

}

if (minute1 <= minute2) {

durationMinute = minute2 - minute1;

}

else {

durationMinute = 60 - (minute1 - minute2);

if (durationHour == 0) {

durationHour = 23;

}

else {

--durationHour;

}

}

if (second1 <= second2) {

durationSecond = second2 - second1;

}

else {

durationSecond = 60 - (second1 - second2);

--durationMinute;

}

cout << "Продолжительность промежутка: " << durationHour << " часа " << durationMinute << " минут " << durationSecond << " секунд.\n";

return 0;

}

№3

#include <iostream>

#include <conio.h>

using namespace std;

void convert10to16()

{

int number;

cout << "Введите число: ";

cin >> number;

char digits16[] = { '0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F' };

if (number < 16)

{

cout << digits16[number];

}

else

{

int n=0, n1, val1 = number, val2 = val1 % 16;

for (; val1 > 0; ++n)

{

val1 /= 10;

}

val1 = number;

int\* arr = new int[n];

for (int i = 0; ; ++i)

{

arr[i] = val2;

val1 /= 16;

if (val1 < 16)

{

arr[i + 1] = val1;

n1 = i + 1;

break;

}

else

{

val2 = val1 % 16;

}

}

for (int i = n1; i >= 0; --i)

{

cout << digits16[arr[i]];

}

}

}

int digit16to10(char number16)

{

char digits16[] = { '0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

for (int i = 0; i < 16; ++i)

{

if (digits16[i] == number16)

{

return i;

}

}

}

void convert16to10()

{

string number16;

cout << "Введите число: ";

cin >> number16;

if (number16.length() == 1)

{

cout << digit16to10(number16[0]) << endl;

}

else

{

int val1 = 1, number10 = 0;;

for (int i = number16.length() - 1; i >= 0; --i)

{

number10 += digit16to10(number16[i]) \* val1;

val1 \*= 16;

}

cout << number10 << endl;

}

}

int main()

{

setlocale(LC\_ALL, "RU");

cout << "Нажмите N, если хотите перевести число из 10чной СИ в 16ричную СИ.\n";

cout << "Нажмите M, если хотите перевести число из 16ричной СИ в 10чную СИ.\n";

switch (\_getch())

{

case 'N': case 'n':

convert10to16();

break;

case 'M': case 'm':

convert16to10();

break;

default:

break;

}

}

№6

#include <iostream>

using namespace std;

int main()

{

int N;

cin >> N;

int\* arr = new int[N];

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

{

cin >> arr[i];

}

bool IsProgression = true;

int d1, d2;

if (N > 1)

{

d1 = arr[1] - arr[0];

}

else

{

return 0;

}

for (int i = 1; i < N - 1; ++i)

{

d2 = arr[i + 1] - arr[i];

if (d1 == d2 && d1!=0)

{

d1 = d2;

}

else

{

IsProgression = false;

break;

}

}

if (IsProgression)

{

return d1;

}

else

{

return 0;

}

}

№7

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int N;

cin >> N;

int NumOfFloor = 0, a = N, NLeft;

bool toBreak = false;

for (int i = 1; ;)

{

for (int i1 = 0 ; ;++i1)

{

if (a <= 0)

{

NLeft = i + a;

toBreak = true;

break;

}

else if (i1 >= i)

{

break;

}

else

{

a -= i;

++NumOfFloor;

}

}

if (toBreak)

{

toBreak = false;

break;

}

else

{

++i;

}

}

cout << "Номер комнаты: " << N << endl;

cout << "Этаж: " << NumOfFloor << endl;

cout << "Номер комнаты слева: " << NLeft << endl;

}

№8

#include <iostream>

using namespace std;

void oneDigit(int digit, bool bigLetter, bool toIncline)

{

if (!toIncline)

{

if (bigLetter)

{

string oneDgtToWord[] = { "Ноль","Один","Два","Три","Четыре","Пять","Шесть","Семь","Восемь","Девять" };

cout << oneDgtToWord[digit];

}

else

{

string oneDgtToWord[] = { "","один ","два ","три ","четыре ","пять ","шесть ","семь ","восемь ","девять " };

cout << oneDgtToWord[digit];

}

}

else

{

if (bigLetter)

{

string oneDgtToWord[] = { "","","Две ","Три ","Четыре ","Пять ","Шесть ","Семь ","Восемь ","Девять " };

cout << oneDgtToWord[digit];

}

else

{

string oneDgtToWord[] = { "","одна ","две ","три ","четыре ","пять ","шесть ","семь ","восемь ","девять " };

cout << oneDgtToWord[digit];

}

}

return;

}

void twoDigit(int\* digits, bool bigLetter, bool toIncline)

{

if (bigLetter)

{

if (digits[0] == 1)

{

switch (digits[1])

{

case 0:

cout << "Десять ";

break;

case 1:

cout << "Одиннадцать ";

break;

case 2:

cout << "Двенадцать ";

break;

case 3:

cout << "Тринадцать ";

break;

case 4:

cout << "Четырнадцать ";

break;

case 5:

cout << "Пятнадцать ";

break;

case 6:

cout << "Шестнадцать ";

break;

case 7:

cout << "Семьнадцать ";

break;

case 8:

cout << "Восемьнадцать ";

break;

case 9:

cout << "Девятнадцать ";

break;

default:

break;

}

}

else

{

switch (digits[0])

{

case 2:

cout << "Двадцать ";

break;

case 3:

cout << "Тридцать ";

break;

case 4:

cout << "Сорок ";

break;

case 5:

cout << "Пятьдесят ";

break;

case 6:

cout << "Шестьдесят ";

break;

case 7:

cout << "Семьдесят ";

break;

case 8:

cout << "Восемьдесят ";

break;

case 9:

cout << "Девяносто ";

break;

default:

break;

}

if (digits[1] == 0)

{

return;

}

else

{

oneDigit(digits[1], 0, toIncline);

return;

}

}

}

else

{

if (digits[0] == 1)

{

switch (digits[1])

{

case 0:

cout << "десять ";

break;

case 1:

cout << "одиннадцать ";

break;

case 2:

cout << "двенадцать ";

break;

case 3:

cout << "тринадцать ";

break;

case 4:

cout << "четырнадцать ";

break;

case 5:

cout << "пятнадцать ";

break;

case 6:

cout << "шестнадцать ";

break;

case 7:

cout << "семьнадцать ";

break;

case 8:

cout << "восемьнадцать ";

break;

case 9:

cout << "девятнадцать ";

break;

default:

break;

}

}

else

{

switch (digits[0])

{

case 2:

cout << "двадцать ";

break;

case 3:

cout << "тридцать ";

break;

case 4:

cout << "сорок ";

break;

case 5:

cout << "пятьдесят ";

break;

case 6:

cout << "шестьдесят ";

break;

case 7:

cout << "семьдесят ";

break;

case 8:

cout << "восемьдесят ";

break;

case 9:

cout << "девяносто ";

break;

default:

break;

}

if (digits[1] == 0)

{

return;

}

else

{

oneDigit(digits[1], 0, toIncline);

return;

}

}

}

}

void threeDigit(int\* digits, bool bigLetter, bool toIncline)

{

if (bigLetter)

{

switch (digits[0])

{

case 1:

cout << "Сто ";

break;

case 2:

cout << "Двести ";

break;

case 3:

cout << "Триста ";

break;

case 4:

cout << "Четыреста ";

break;

case 5:

cout << "Пятьсот ";

break;

case 6:

cout << "Шестьсот ";

break;

case 7:

cout << "Семьсот ";

break;

case 8:

cout << "Восемьсот ";

break;

case 9:

cout << "Девятьсот ";

break;

default:

break;

}

}

else

{

switch (digits[0])

{

case 1:

cout << "сто ";

break;

case 2:

cout << "двести ";

break;

case 3:

cout << "триста ";

break;

case 4:

cout << "четыреста ";

break;

case 5:

cout << "пятьсот ";

break;

case 6:

cout << "шестьсот ";

break;

case 7:

cout << "семьсот ";

break;

case 8:

cout << "восемьсот ";

break;

case 9:

cout << "девятьсот ";

break;

default:

break;

}

}

if (digits[1] != 0)

{

int twodigits[] = { digits[1],digits[2] };

twoDigit(twodigits, 0, toIncline);

}

else

{

oneDigit(digits[2], 0, toIncline);

}

}

void underMillion(int\* digits, int amtofdigits, bool bigLetter)

{

int n = amtofdigits - 3;

int\* tdigits = new int[n];

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

{

tdigits[i] = digits[i];

}

if (n == 1 && tdigits[0] == 1 && bigLetter)

{

cout << "Тысяча ";

}

else

{

switch (n)

{

case 1:

oneDigit(tdigits[0], bigLetter, 1);

break;

case 2:

twoDigit(tdigits, bigLetter, 1);

break;

case 3:

threeDigit(tdigits, bigLetter, 1);

break;

default:

break;

}

int a = tdigits[n - 1], b=tdigits[n-2];

if (a == 1)

{

cout << "тысяча ";

}

else if ((a > 1 && a < 5 && b == 0) || (a == 2 && b != 1))

{

cout << "тысячи ";

}

else

{

cout << "тысяч ";

}

}

if (digits[n]==0)

{

int tdigits2[] = { digits[n + 1],digits[n + 2] };

twoDigit(tdigits2, 0, 0);

}

else

{

int tdigits2[] = { digits[n], digits[n + 1],digits[n + 2] };

threeDigit(tdigits2, 0, 0);

}

}

void underBillion(int\* digits, int amtofdigits)

{

int n = amtofdigits - 6;

int\* tdigits = new int[n];

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

{

tdigits[i] = digits[i];

}

if (n == 1 && tdigits[0] == 1)

{

cout << "Миллион ";

}

else

{

switch (n)

{

case 1:

oneDigit(tdigits[0], 1, 0);

break;

case 2:

twoDigit(tdigits, 1, 0);

break;

case 3:

threeDigit(tdigits, 1, 0);

break;

default:

break;

}

int a = tdigits[n - 1];

if (a == 1 && tdigits[n-2]!=1)

{

cout << "миллион ";

}

else if (a > 1 && a < 5 && tdigits[n-2]!=1)

{

cout << "миллиона ";

}

else

{

cout << "миллионов ";

}

}

if (digits[n] == 0 && digits[n+1] == 0 && digits[n+2] == 0)

{

int tdigits[3];

for (int i = 0; i < 3; ++i)

{

tdigits[i] = digits[n + 3];

++n;

}

if (tdigits[0] == 0 && tdigits[1] == 0)

{

oneDigit(tdigits[2], 0, 0);

return;

}

else if (tdigits[0] == 0)

{

int tdigits2[] = { tdigits[1], tdigits[2] };

twoDigit(tdigits2, 0, 0);

return;

}

else

{

threeDigit(tdigits, 0, 0);

return;

}

}

if (digits[n] == 0 && digits[n + 1] == 0)

{

int tdigits[4];

for (int i = 0; i < 4; ++i)

{

tdigits[i] = digits[n + 2];

++n;

}

underMillion(tdigits, 4, 0);

}

else if (digits[n] == 0)

{

int tdigits[5];

for (int i = 0; i < 5; ++i)

{

tdigits[i] = digits[n + 1];

++n;

}

underMillion(tdigits, 5, 0);

}

else

{

int tdigits[6];

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

{

tdigits[i] = digits[n];

++n;

}

underMillion(tdigits, 6, 0);

}

}

void convertToWord(int number, int amtofdigits)

{

int\* digits = new int[amtofdigits];

int degree10 = pow(10, amtofdigits - 1), temp = number;

digits[0] = number / degree10;

for (int i = 1; i < amtofdigits; ++i)

{

degree10 = degree10 / 10;

temp = temp - digits[i - 1] \* degree10 \* 10;

digits[i] = temp / degree10;

}

for (int i = 0; i < amtofdigits; i++)

{

cout << digits[i];

}

cout << endl;

switch (amtofdigits)

{

case 1:

oneDigit(digits[0], 1, 0);

return;

case 2:

twoDigit(digits, 1, 0);

return;

case 3:

threeDigit(digits, 1, 0);

return;

default:

break;

}

if (amtofdigits < 7)

{

underMillion(digits, amtofdigits, 1);

}

else if (amtofdigits < 10)

{

underBillion(digits, amtofdigits);

}

cout << endl;

}

int main()

{

setlocale(LC\_ALL, "ru");

int number, amtofdigits=1;

cin >> number;

if (number == 1000000000)

{

cout << "Миллиард " << endl;

return 0;

}

if (number > 1000000000)

{

cout << "Вы вышли за диапазон допустимых чисел." << endl;

return 0;

}

int val = number;

for (;val>9;)

{

val = val / 10;

++amtofdigits;

}

convertToWord(number, amtofdigits);

return 0;

}

№9

#include <iostream>

#include <ctime>

using namespace std;

int main()

{

srand(time(NULL));

int N;

cin >> N;

int\* arr = new int[N];

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

{

arr[i] = rand() % (N + 1);

for (int k = 0; k < i; ++k)

{

if (arr[k] == arr[i])

{

arr[i] = rand() % (N + 1);

k = -1;

}

}

}

for (int i = 0; i < N; ++i) cout << arr[i] << endl;

int value = 0;

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

{

if (arr[i] == value)

{

++value;

i = -1;

}

if (i == N - 1 && arr[i] != value)

{

return value;

}

}

}