#include <iostream>

#include <ctime>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int N = 100;

int i, sz, A[N];

int rmn = 0, rmx = 99;

cout << "Введите размер массива ";

cin >> sz;

cout << "Массив А:" << endl;

srand((unsigned)time(NULL));

for (i = 0; i < sz; i++)

{

A[i] = (int)(((double)rand() /

(double)RAND\_MAX) \* (rmx - rmn) + rmn);

cout << A[i] << " ";

}

double o;

int j = 0;

for (i = 0; i < sz; i++)

{

o = A[i] % 7;

if (o == 0)

{

for (int j = i; j <= sz; j++)

A[j] = A[j + 1]; sz--;

}

}

cout << endl << endl << endl;

cout << "Массив после удаления элементов" << endl;

for (i = 0; i < sz; i++)

{

cout << A[i] << " ";

}

cout << endl;

cout << endl << endl << endl;

i = 0;

while (i < sz)

{

if (A[i] % 2 == 1)

{

for (j = sz; j > i; j--)

A[j] = A[j - 1];

sz++;

A[i + 1] = 4;

}

i++;

}

cout << "Массив после добавления элемента со значением 4 после каждого нечетного элемента: " << endl;

for (i = 0; i < sz; i++)

{

cout << A[i] << " ";

}

cout << endl;

cout << endl << endl << endl;

}

[#include](https://www.instagram.com/explore/tags/include/) <iostream>  
[#include](https://www.instagram.com/explore/tags/include/) <ctime>  
[#include](https://www.instagram.com/explore/tags/include/) <vector>  
using namespace std;  
int main()  
{  
setlocale (LC\_ALL, "rus");  
using namespace std;  
int x[200], i, n, k, j, r;  
srand(time(NULL));  
cout << "введите размерность массива: " << endl;  
cin >> n;  
cout << "Massiv:" << endl << " " << endl;  
for (i = 0; i < n; i++)  
{  
x[i] = rand() % 99; cout << x[i] << " ";  
}  
  
cout << endl << "Введите значение которое хотите удалить ";  
cin >> k;  
  
cout << endl << ":" << endl;  
for (i = 0; i < n; i++)  
{  
if (x[i] == k)  
{  
for (j = i; j < n; j++)  
{  
x[j] = x[j + 1];  
} n--; x[j] = -1; i--;  
}  
}  
for (i = 0; i < n; i++)  
  
{  
cout << x[i] << " ";  
}  
  
  
cout << endl << "massiv2: ";  
  
for (i = 0; i < n; i++)  
{  
if (x[i] % 2 == 0)  
{  
n++; for (r = n; r > i - 1; r--)  
{  
x[r] = x[r - 1];

[](https://www.instagram.com/_p1ndoska_/)

[#include](https://www.instagram.com/explore/tags/include/) <iostream>  
[#include](https://www.instagram.com/explore/tags/include/) <ctime>  
using namespace std;  
int main()  
{  
setlocale(LC\_ALL, "rus");  
srand(time(NULL));  
cout << "Введите размерность массива: ";  
int SIZE;  
cin >> SIZE;  
  
int\* arr = new int[SIZE];  
for (int i = 0; i < SIZE; i++)  
{  
arr[i] = rand() % 99;  
}  
for (int i = 0; i < SIZE; i++)  
{  
cout << arr[i] << endl;  
}  
  
int maxValue = arr[0];  
for (int i = 1; i < SIZE; i++)  
{  
if (arr[i] > maxValue)  
{  
maxValue = arr[i];  
}  
}  
cout << "Наибольшее число в массиве: " << maxValue << endl;  
  
cout << "После замены: " << endl;  
for (int i = 0; i < SIZE; i++)  
{  
if (i % 2 == 0)  
{  
arr[i] = maxValue;  
}  
cout << arr[i] << endl;  
}  
}



Битовые операции  
#include <iostream>

using namespace std;

int main()

{

/\*1. Ввести целое число A.Инвертировать все биты с 2 по 14, включая эти биты.Вывести результат.\*/

/\*unsigned a;

char tA[33], tM[33];

cout << "Enter number: ";

cin >> a;

\_itoa\_s(a, tA, 2);

cout << "Binary number: " << tA << endl;

unsigned mask = (1u << 13) - 1;

mask <<= 2;

\_itoa\_s(mask, tM, 2);

cout << "Mask: " << tM << endl;

a ^= mask;

\_itoa\_s(a, tA, 2);

cout << "Inverted number: " << tA;\*/

/\*2. Инвертировать в числе А n битов влево от позиции p, заменить ими m битов числа В, начиная с позиции q.\*/

unsigned a, p, n, bits, m, b, q, c;

char tA[33], tM[33], tB[33], tBit[33], tMB[33], tC[33];

cout << "Enter number a: ";

cin >> a;

\_itoa\_s(a, tA, 2);

cout << "Binary number a: " << tA << endl;

cout << "Enter p (position): ";

cin >> p;

cout << "Enter n (number): ";

cin >> n;

unsigned mask = (1u << n) - 1; // 1u = 00000000000001; (1u << n) = 000000001000000 (n - 1 нулей справа от 1); (1u << n) - 1 = 000000001111111 (просто n единиц)

mask <<= p; //сдвигаем маску на p влево так как нужно получить n битов с позиции р

\_itoa\_s(mask, tM, 2);

cout << "Mask A: " << tM << endl;

a ^= mask; // xor ^ инвертирует биты которые в маске равны 1

\_itoa\_s(a, tA, 2);

cout << "Inverted number: " << tA << endl;

bits = a;

bits >>= p; //сдвигаем обратно на р уже вправо чтобы получить в самом начале наши инвертированные биты

cout << "Enter m: ";

cin >> m;

bits <<= (32 - m ); //таким образом мы записываем в переменную bits только

bits >>= (32 - m ); //наши инвертированные биты (m битов т.к. так надо по условию)

\_itoa\_s(bits, tBit, 2);

cout << "Inverted "<<m<<" bits: " << tBit << "\nEnter number b: ";

cin >> b;

\_itoa\_s(b, tB, 2);

cout << "Binary number b: " << tB << "\nEnter position q: ";

cin >> q;

bits <<= q; //сдвигаем влево на q т.к. надо вставить биты bits с позиции q

c = b;

c ^= bits; // ф-ия xor ^= заменяет m битов в b с позиции q нашими битами bits из числа а

\_itoa\_s(c, tC, 2);

cout << "Changed number b: " << tC;

}