Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

 Лабараторная работа 11

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Бітавыя аперацыі»

Выканала:

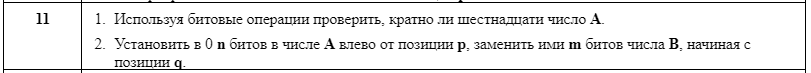
Студэнтка 1 курса 6 группы

Лускіна Вольга Аляксандраўна

Выкладчык: асс. Андронава М.В.

2023, Мінск

Варыянт 11



Заданне 1

#include <iostream>

#include <Windows.h>

using namespace std;

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int A; char tmp[33]; // Масіў для ліка ў двайковай сістэме

cout << "Увядзіце лік: " << endl; cin >> A;

\_itoa\_s(A, tmp, 2); // Ператварэнне ліка A ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "Лік у двайковай сістэме: " << tmp << endl;

if ((A & 15) == 0) {

cout << "Лік кратны 16" << endl; // Праверка, ці з'яўляецца лік А кратным 16

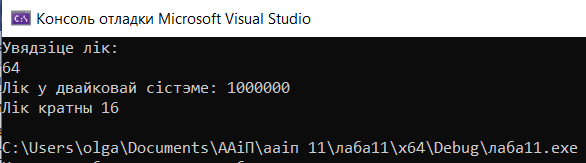
}

else {

cout << "Лік не кратны 16" << endl;

}

}



Заданне 2

#include <iostream>

#include <Windows.h>

using namespace std;

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char tmp[33]; // Масіў для ліка ў двайковай сістэме

int A, B, n, p, m, q, maskA = 0, maskB = 0;

cout << "Увядзіце лік A= "; cin >> A;

\_itoa\_s(A, tmp, 2); // Ператварэнне ліку A ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "A= " << tmp << endl;

cout << "Увядзіце колькасць змяняемых бітоў ліка A: " << endl;

cin >> n;

cout << "Увядзіце пачатак пазіцыі: " << endl;;

cin >> p;

cout << "Увядзіце лік B= "; cin >> B;

\_itoa\_s(B, tmp, 2); // Ператварэнне ліка В ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "B= " << tmp << endl;

cout << "Увядзіце колькасць змяняемых бітоў B" << endl;

cin >> m;

cout << "Увядзіце пачатак пазіцыі: " << endl;

cin >> q;

for (int i = p - 1; i < p + n - 1; i++) { // Цыкл для стварэння маскі maskA шляхам ўстаноўкі бітаў у дыяпазоне ад p да p + n - 1

maskA += pow(2, i); // З дапамогай pow(2, n) атрымліваем лік, у якога толькі адзін біт роўны 1, а ўсе астатнія нулю. А нам патрэбна атрымаць лік, у якога ўсе n малодшых бітоў роўныя 1, таму адымаем 1

}

A = (A & ~maskA); // Ужыванне інвертыраваннай маскі maskA да ліку A для ачысткі адпаведных бітаў

p = q - p; // Падлік зрушэння р

if (p < 0) { // Калі зрушэнне p адмоўнае, зрушваем маску maskA направа на абсалютная значэнне p

maskA = maskA >> abs(p);

}

else if (p > 0) { // Калі зрушэнне p станоўчае, зрушваем маску maskA налева на p

maskA = maskA << p;

}

for (int i = q - 1; i < q - 1 + m; i++) { // Цыкл для стварэння маскі maskВ шляхам ўстаноўкі бітаў у дыяпазоне ад q - 1 да q - 1 + m

maskB += pow(2, i);

}

B = (B & ~maskB); // Выкарыстанне інвертыраваннай маскі maskB да ліку В для ачысткі адпаведных бітоў

B = (B | maskA); // Выкарыстанне маскі maskA да ліку В шляхам выканання лагічнай аперацыі АБО

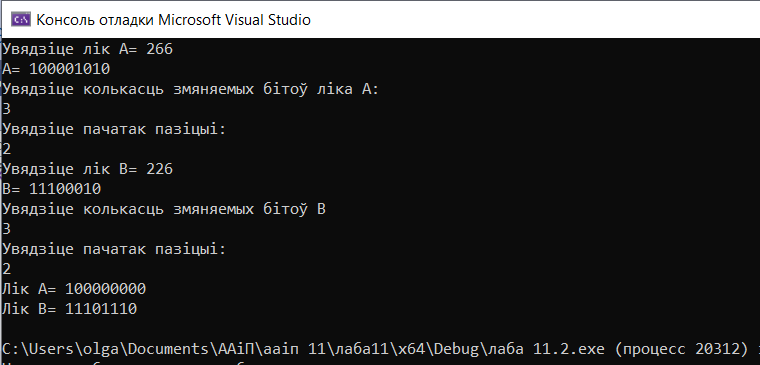
\_itoa\_s(A, tmp, 2); // Ператварэнне ліку А ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

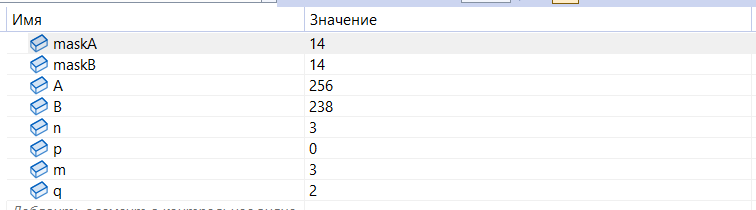
cout << "Лік A= " << tmp << endl;

\_itoa\_s(B, tmp, 2);

cout << "Лік B= " << tmp << endl;

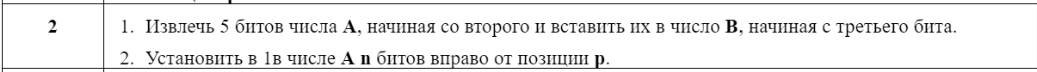
}





Дадатковыя заданні:

Варыянт 2



Заданне 1

#include <iostream>

#include <Windows.h>

using namespace std;

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char tmp[33]; // Масіў для ліка ў двайковай сістэме

int A, B, maskA = 62; // 62 у двайковай сістэме роўнае 00111110. maskA выкарыстоўваецца каб узяць 5 бітоў з ліка А, пачынаючы з другога біта. Нам патрэбна ўзяць 5 бітоў, з доугога па шосты ўключна, мы ўсталёўваем адпаведныя біты ў maskA у 1, каб пасля выкарыстоўваць яго ў аперацыі пабітавага І (&) з лікам A. Астатнія біты maskA ўсталёўваюцца ў 0, каб не ўплываць на выманне патрэбных бітоў

int maskB = ~(maskA << 1); // Вылічэнне maskB шляхам зруху maskA на 1 біт налева і інверсіі ўсіх бітоў

cout << "Лік А="; cin >> A;

cout << "Лік В="; cin >> B;

\_itoa\_s(A, tmp, 2); // Ператварэнне ліка А ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "A=" << tmp << endl;

\_itoa\_s(B, tmp, 2); // Ператварэнне ліка В ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "B=" << tmp << endl;

\_itoa\_s((A & maskA) >> 1, tmp, 2); // Выманне 5 бітоў з ліка А, пачынаючы з другога біта і зрушаючы на 1 направа, ператварэнне ліка А ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "Узятыя біты А: " << tmp << endl;

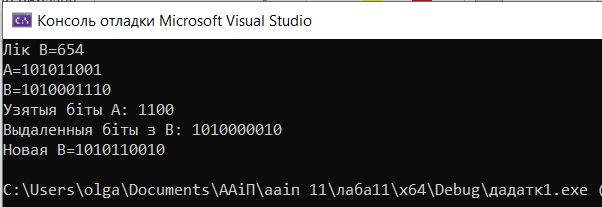
\_itoa\_s(B & maskB, tmp, 2); // Выдаленне пэўных бітоў з ліка В, ператварэнне ліка В ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "Выдаленныя біты з B: " << tmp << endl;

\_itoa\_s(((B & maskB) | ((A & maskA) << 1)), tmp, 2); // Выманне і устаўка бітоў з А у В, ператварэнне выніка ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "Новая B=" << tmp << endl;

}



Заданне 2

#include <iostream>

#include <Windows.h>

using namespace std;

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char tmp[33]; // Масіў для ліка ў двайковай сістэме

int A, B, n, p; // Зменная n потым выконвае і ролю маскі

cout << "Лік А="; cin >> A;

\_itoa\_s(A, tmp, 2); // Ператварэнне ліку A ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "A=" << tmp << endl;

cout << "Увядзіце колькасць змяняемых бітоў ліка A :"; cin >> n;

cout << "Увядзіце пачатак пазіцыі :"; cin >> p;

p = p - n; // Зрух пазіцыі на n

n = pow(2, n) - 1; // З дапамогай pow(2, n) атрымліваем лік, у якога толькі адзін біт роўны 1, а ўсе астатнія нулю. А нам патрэбна атрымаць лік, у якога ўсе n малодшых бітоў роўныя 1, таму адымаем 1

if (p > 0) {

n = n << p; // Калі зрушэнне p станоўчае, зрушваем маску maskA налева на p

}

else {

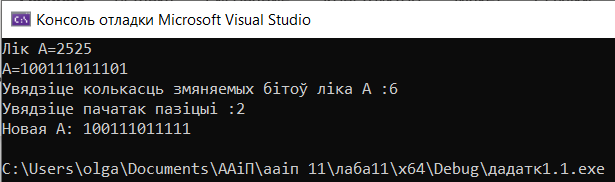
n = n >> abs(p); Калі зрушэнне p адмоўнае, зрушваем маску maskA направа на абсалютная значэнне p

}

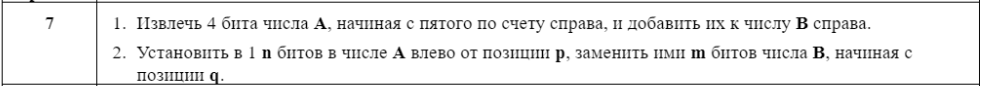
\_itoa\_s(A | n, tmp, 2); // Робім аперацыю пабітовага АБО з маскай да ліка А

cout << "Новая A: " << tmp << endl;

}



Варыянт 7



Заданне 1:

#include <iostream>

#include <Windows.h>

using namespace std;

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char tmp[33]; // Масіў для ліка ў двайковай сістэме

int A, B, maskA = 30; // 30 у двайковай сістэме роўнае 00011110. maskA выкарыстоўваецца каб узяць 5 бітоў з ліка А, пачынаючы з другога біта. Нам патрэбна ўзяць 5 бітоў, з доугога па шосты ўключна, мы ўсталёўваем адпаведныя біты ў maskA у 1, каб пасля выкарыстоўваць яго ў аперацыі пабітавага І (&) з лікам A. Астатнія біты maskA ўсталёўваюцца ў 0, каб не ўплываць на выманне патрэбных бітоў

int maskB = ~(maskA << 1); // Вылічэнне maskB шляхам зруху maskA на 1 біт налева і інверсіі ўсіх бітоў

cout << "Лік А="; cin >> A;

cout << "Лік В="; cin >> B;

\_itoa\_s(A, tmp, 2); // Ператварэнне ліка А ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "A=" << tmp << endl;

\_itoa\_s(B, tmp, 2); // Ператварэнне ліка В ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "B=" << tmp << endl;

\_itoa\_s((A & maskA) >> 1, tmp, 2); // Выманне 4 бітоў з ліка А, пачынаючы з другога біта і зрушаючы на 1 направа, ператварэнне ліка А ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "Узятыя біты А: " << tmp << endl;

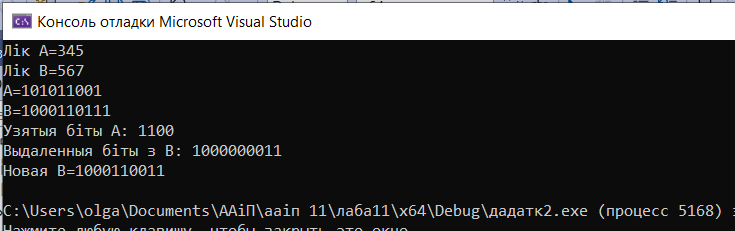
\_itoa\_s(B & maskB, tmp, 2); // Выдаленне пэўных бітоў з ліка В, ператварэнне ліка В ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "Выдаленныя біты з B: " << tmp << endl;

\_itoa\_s(((B & maskB) | ((A & maskA) << 1)), tmp, 2); // Выманне і устаўка бітоў з А у В, ператварэнне выніка ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "Новая B=" << tmp << endl;

}



Заданне 2:

#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char tmp[33]; // Масіў для ліка ў двайковай сістэме

int A, B, n, p, m, q;

cout << "Увядзіце лік A= "; cin >> A;

\_itoa\_s(A, tmp, 2); // Ператварэнне ліку A ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "A= " << tmp << endl;

cout << "Увядзіце колькасць змяняемых бітоў ліка A: " << endl;

cin >> n;

cout << "Увядзіце пачатак пазіцыі: " << endl;;

cin >> p;

cout << "Увядзіце лік B= "; cin >> B;

\_itoa\_s(B, tmp, 2); // Ператварэнне ліка В ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "B= " << tmp << endl;

cout << "Увядзіце колькасць змяняемых бітоў B" << endl;

cin >> m;

cout << "Увядзіце пачатак пазіцыі: " << endl;

cin >> q;

int maskA = ((1 << n) - 1) << (p - n); // Робім лік, у якім толькі n-ы ўстаноўлены ў 1, астатнія роўныя 0. Потым адымаем 1, каб усталяваць усе n малодшых бітоў у 1, а астатнія пакінуць роўнымі 0. Здвігаем на (p - n) пазіцый налева

A = (A & ~maskA); // Пабітавая аперацыя І

int maskB = ((1 << m) - 1) << (q - p);

B = (B & ~maskB);

B |= (A >> (p - n)) & maskA; // Замяняем m бітоў у ліку В

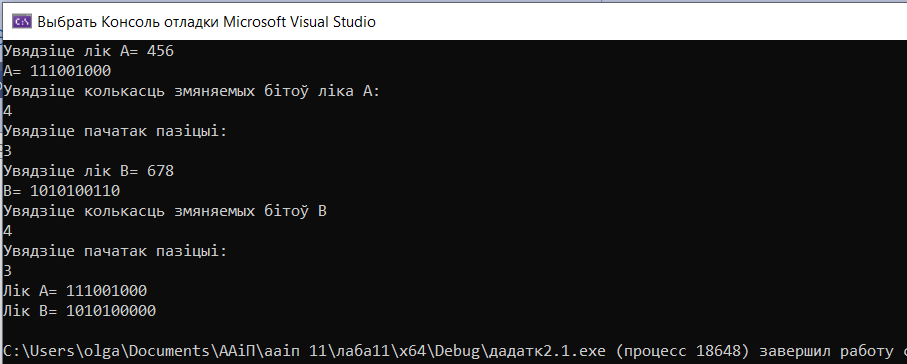
\_itoa\_s(A, tmp, 2); // Ператварэнне ліку А ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "Лік A= " << tmp << endl;

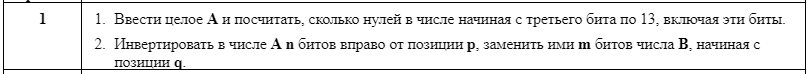
\_itoa\_s(B, tmp, 2);

cout << "Лік B= " << tmp << endl;

}



Варыянт 1



Заданне 1:

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int A;

cout << "Лік А= ";

cin >> A;

char tmp[33];

\_itoa\_s(A, tmp, 2); // Ператварэнне ліка А ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "A: " << tmp << endl;

int maskA = 1; // Маска А для падліка колькасці нулёў з 3 па 13 біт

maskA = maskA << 13;

maskA -= 1;

maskA = maskA >> 2;

maskA = maskA << 2;

A = (maskA & A) >> 2; // Пабітавая аперацыя І, каб пакінуць толькі тыя біты, якія адпавядаюць бітам у maskA, потым зрушваем рэзультат на 2 пазіцыі направа

int counter = 0;

for (int i = 0; i < 11; i++)

{

if ((A & 1) == 0) {

counter++; // Глядзім, ці з'яўляецца малодшы біт ліка А роўным 0

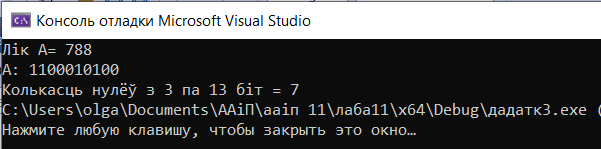
}

A = A >> 1; // Зрушваем направа, эквівалентна дзяленню на 2 і адкідвання астатку

}

cout << "Колькасць нулёў з 3 па 13 біт = " << counter;

}



Заданне 2:

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int A, maskA, n, p;

char tmp[33]; // Масіў для ліка ў двайковай сістэме

cout << "Увядзіце лік А= "; cin >> A;

\_itoa\_s(A, tmp, 2); // Ператварэнне ліку A ў двайковую сістэму злічэння, і захаванне выніку ў масіў tmp

cout << "A= " << tmp << endl;

cout << "Увялзіце колькасць бітоў для інвертыравання: ";

cin >> n;

cout << "Увядзіце пачатак пазіцыі: ";

cin >> p;

maskA = 1 << n; // Маска для інвертыравання n бітоў направа ад пазіцыі p

maskA = maskA - 1;

maskA = maskA << p - n;

\_itoa\_s(maskA, tmp, 2);

\_itoa\_s(A ^ maskA, tmp, 2); // Выкарыстоўваем маску А

cout << "Новая А= " << tmp << endl;

}

