Варыянт 3

Заданне 5. В соответствии со своим вариантом разработать программы, использующие битовые операции для решения задач, представленных в таблице.

Результаты одной из программ представить в Отладчике.

|  |
| --- |
| 1. Ввести целое число A. Инвертировать все биты с 2 по 14, включая эти биты. Вывести результат. |
| #include <iostream>  #include <windows.h>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  int a, mask = 16382; //Ствараем смаску выгляда 11111111111110  char tmp[33];  cout << "Увядзіце лік: "; cin >> a;  //Выводзім лік а ў двайковым выглядзе  \_itoa\_s(a, tmp, 2);  cout << "A у двайковай сістэме: " << tmp << endl;  //Інвертуем з 2 па 14 біты  a = a ^ mask;  //Выводзім вынік  \_itoa\_s(a, tmp, 2);  cout << "Інвертаваны лік у двайковым выглядзе: " << tmp << endl;  cout << "У дзесятковым выглядзе: " << a;  return 0;  } |
|  |
|  |
| 2. Инвертировать n битов в числе А влево от позиции p, заменить ими m битов числа В, начиная с позиции q. |
| #include <iostream>  using namespace std;  #include <windows.h>  int main()  {  SetConsoleOutputCP(1251);  int A, maskA, n, p, m, q, B;  char tmp[33];  cout << "Увядзіце лік A: ";  cin >> A;  cout << "Увядзіце пазіцыю, з якой будуць канвертаваны біты (пазіцыя неўключна): ";  cin >> p;  cout << "Увядзіце лік бітаў для канвертавання: ";  cin >> n;  //Ствараем маску для інвертацыі n бітаў у ліку А налева ад пазіцыі p  maskA = 1 << n;  maskA = maskA - 1;  maskA = maskA << p;  \_itoa\_s(A, tmp, 2);  cout << "A у двайковай сістэме: " << tmp << endl;  \_itoa\_s(A ^ maskA, tmp, 2);  cout << "Пераўтвораны лік А:" << tmp << endl;  cout << "Увядзіце пазіцыю, з якой будуць заменены біты ў Б (пазіцыя неўключна): ";  cin >> m;  cout << "Увядзіце колькасць бітаў для замены (менш або роўна колькасці бітаў, канвертаваных у A): ";  cin >> q;  //Праверка, што мы замяняем лік бітаў ва Ў роўнае ці меншае за лік бітаў, змененых у А  if (q > n) {  cout << "Памылка! увядзіце лік q <= n";  return 0;  }  cout << "Увядзіце лік В: "; cin >> B;  \_itoa\_s(B, tmp, 2);  cout << "B у двайковай сістэме:" << tmp << endl;  // Ствараем маску для У: зрушваем маску на p + n - q направа для атрымання патрэбнага ліку бітаў q. затым зрушваем іх на m налева і инвертируем вынік  int maskB = ~((maskA >> (p + n - q)) << m);  // Падрыхтоўвае біты, якімі будуць заменены біты ва Ў. калі для замены ва Ў выкарыстоўваецца менш  // бітаў чым было канвертавана ў А, тое адкіне наперадзе ідучыя біты і возьме паказаную колькасць наступных.  int res = ((A ^ maskA) & maskA) >> p;  res = res & ((1 << q) - 1);  res = res << (m);  \_itoa\_s(((B & maskB) | res), tmp, 2);  cout << "Вынік замены для У:" << tmp << endl;  return 0;  } |
|  |

Варыянт 9

|  |
| --- |
| 1. Извлечь 5 битов числа A, начиная с третьего по счету справа, и вставить их в число B, начиная со 2 по счету справа. |
| #include <iostream>  #include <windows.h>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  int A, maskA, n, p, m, q, B;  char tmp[33];  cout << "Увядзіце лік A: "; cin >> A;  //Ствараем маску для 5 бітаў у ліку А налева ад пазіцыі 2 неўключна  maskA = 1 << 5;  maskA = maskA - 1;  maskA = maskA << 2;  cout << "Увядзіце лік В: "; cin >> B;  \_itoa\_s(A, tmp, 2);  cout << "A у двайковай сістэме: " << tmp << endl;  \_itoa\_s(B, tmp, 2);  cout << "B у двайковай сістэме: " << tmp << endl;  // Ствараем маску для B: инвертируем маску для А і зрушваем на 1 направа  int maskB = ~maskA >> 1;  //Знаходзім вынік карстыючыся маскамі  \_itoa\_s(((B & maskB) | (A & maskA)>>1), tmp, 2);  cout << "Вынік замены для B: " << tmp << endl;  return 0;  } |
|  |
| 2. Установить в 0 n битов в числе А вправо от позиции p. |
| #include <iostream>  #include <windows.h>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  int A, maskA, n, p;  char tmp[33];  cout << "Увядзіце лік A: "; cin >> A;  cout << "Увядзіце пазіцыю: ";  cin >> p;  cout << "Увядзіце лік бітаў: ";  cin >> n;  //Ствараем маску для n бітаў у ліку А спава ад пазіцыі p  maskA = 1 << n;  maskA = maskA - 1;  maskA = maskA << (p-n-1);  maskA = ~maskA;  \_itoa\_s(A, tmp, 2);  cout << "Лік А: " << tmp << endl;  //Знаходзім вынік карстыючыся маскамі  \_itoa\_s(maskA & A, tmp, 2);  cout << "Вынік: " << tmp << endl;  return 0;  } |
|  |

Варыянт 10

|  |
| --- |
| 1. Вывести 6 бит целого числа **А**, начиная со 2-ого битов. |
| #include <iostream>  #include <windows.h>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  int A, maskA, n, p;  char tmp[33];  cout << "Увядзіце лік A: "; cin >> A;  cout << "Увядзіце лік бітаў: "; cin >> n;  //Ствараем маску для n бітаў у ліку А спава ад пазіцыі p  maskA = 1 << n;  maskA = maskA - 1;  maskA = maskA << 1;  \_itoa\_s(A, tmp, 2);  cout << "Лік А: " << tmp << endl;  //Знаходзім вынік карстыючыся маскамі  \_itoa\_s((maskA & A) >> 1, tmp, 2);  cout << "Вынік: " << tmp << endl;  return 0;  } |
|  |
| Инвертировать **n** битов в числе **А** влево от позиции **p**. |
| #include <iostream>  #include <windows.h>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  int A, maskA, n, p, m, q, B;  char tmp[33];  cout << "Увядзіце лік A: ";  cin >> A;  cout << "Увядзіце пазіцыю, з якой будуць канвертаваны біты (пазіцыя неўключна): ";  cin >> p;  cout << "Увядзіце лік бітаў для канвертавання: ";  cin >> n;  //Ствараем маску для інвертацыі n бітаў у ліку А налева ад пазіцыі p  maskA = 1 << n;  maskA = maskA - 1;  maskA = maskA << p;  \_itoa\_s(A, tmp, 2);  cout << "A у двайковай сістэме: " << tmp << endl;  \_itoa\_s(A ^ maskA, tmp, 2);  cout << "Пераўтвораны лік А:" << tmp << endl;  //Знаходзім вынік карстыючыся маскамі  \_itoa\_s((maskA ^ A), tmp, 2);  cout << "Вынік: " << tmp << endl;  return 0;  } |
|  |

Варыянт 12

|  |
| --- |
| 1. Ввести целое число **A**. Инвертировать все биты с 4 по 8, включая эти биты. Вывести полученное число. |
| #include <iostream>  #include <windows.h>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  int A, maskA, n, p, m, q, B;  char tmp[33];  cout << "Увядзіце лік A: ";  cin >> A;  //Ствараем маску для інвертацыі з 4 па 8 бітаў у ліку А  maskA = 1 << 5;  maskA = maskA - 1;  maskA = maskA << 3;  \_itoa\_s(A, tmp, 2);  cout << "A у двайковай сістэме: " << tmp << endl;  \_itoa\_s(A ^ maskA, tmp, 2);  cout << "Пераўтвораны лік А:" << tmp << endl;  return 0;  } |
|  |
| Установить в 0 **n** битов в числе **А** вправо от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**. |
| #include <iostream>  #include <windows.h>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  int A, maskA, n, p, q, m, B;  char tmp[33];  cout << "Увядзіце лік A: "; cin >> A;  cout << "Увядзіце пазіцыю: ";  cin >> p;  cout << "Увядзіце лік бітаў: ";  cin >> n;  //Ствараем маску для n бітаў у ліку А справа ад пазіцыі p  maskA = 1 << n;  maskA = maskA - 1;  maskA = maskA << (p - n - 1);  maskA = ~maskA;  cout << "Увядзіце пазіцыю, з якой будуць заменены біты ў Б (пазіцыя неўключна): ";  cin >> m;  cout << "Увядзіце колькасць бітаў для замены (менш або роўна колькасці бітаў, канвертаваных у A): ";  cin >> q;  //Праверка, што мы замяняем лік бітаў ва Ў роўнае ці меншае за лік бітаў, змененых у А  if (q > n) {  cout << "Памылка! увядзіце лік q <= n";  return 0;  }  cout << "Увядзіце лік В: "; cin >> B;  \_itoa\_s(A, tmp, 2);  cout << "Лік А у двайковай сістэме: " << tmp << endl;  \_itoa\_s(B, tmp, 2);  cout << "B у двайковай сістэме:" << tmp << endl;  //Знаходзім вынік карыстаючыся маскамі  \_itoa\_s(maskA & A, tmp, 2);  cout << "Лік А пасля змены: " << tmp << endl;  int maskB = ~((~maskA >> (p - n - 1 - (q-n))) << m);  // Падрыхтоўвае біты, якімі будуць заменены біты ва B. калі для замены ва B выкарыстоўваецца менш  // бітаў чым было канвертавана ў А, тое адкіне наперадзе ідучыя біты і возьме паказаную колькасць наступных.  int res = (A & maskA) >> (p - n - 1);  res = res & ((1 << q) - 1);  res = res << (m);  \_itoa\_s(((B & maskB) | res), tmp, 2);  cout << "Вынік замены для B:" << tmp << endl;  return 0;  } |
|  |