ЛАБОРАТОРНАЯ РАБОТА №1

по курсу «Архитектура вычислительных систем»

по теме:

Представление данных в ЭВМ

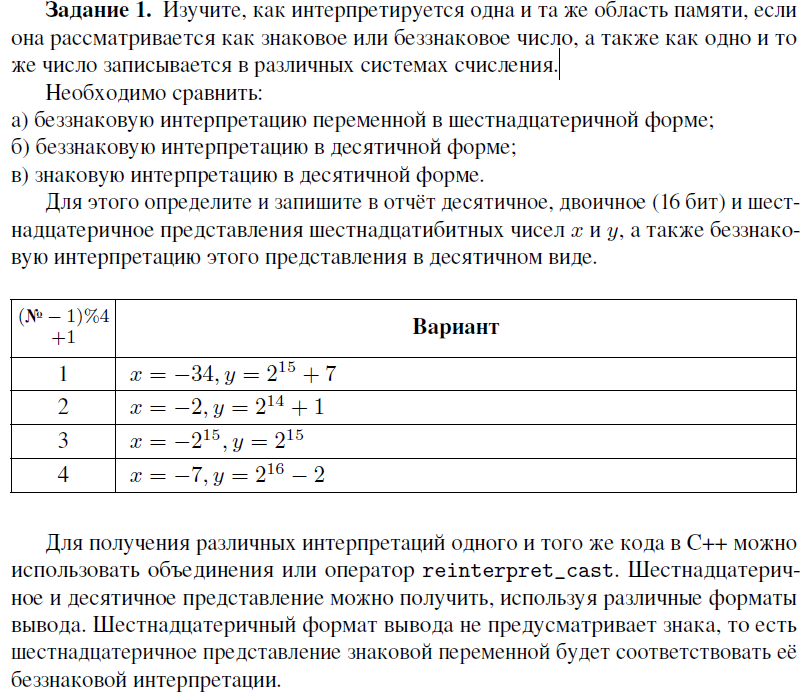
Выполнили:

Студент 3-го курса

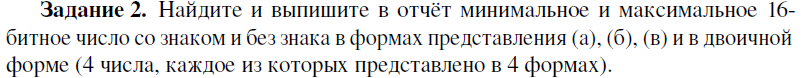
Петров Д.А.

Рукавишников С.?.

Москва 2020



|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  template<typename T>  void print(T Number)  {  cout << "Беззнаковая интерпретация в 16-ой форме: " << hex <<(unsigned short)Number << endl << "Беззнаковая интерпретация в 10-ой форме: " << dec << (unsigned short)Number << endl;  cout << "Знаковая интерпретация в десятичной форме: " << dec << (short)Number << endl;  }  int main()  {  setlocale(LC\_ALL, "Russian");  int x = -34;  int y = pow(2, 15) + 7;  cout << "-34: \n";  print(x);  cout << "\n2^15 + 7: \n";  print(y);  system("Pause");  return 0;  } |
|  |



|  |
| --- |
| #include <iostream>  #include <cmath>  #include <bitset>  using namespace std;  template<typename T>  void PrintTypes(T Number)  {  cout << "Беззнаковая интерпретация в 16-ой форме: " << hex <<(unsigned short)Number << endl << "Беззнаковая интерпретация в 10-ой форме: " << dec << (unsigned short)Number << endl;  cout << "Знаковая интерпретация в десятичной форме: " << dec << (short)Number << endl<<"Двоичная форма:" <<bitset<16>(Number) << endl;  }  int main()  {  setlocale(LC\_ALL, "Russian");  cout << "Максимальное беззнаковое:\n";  PrintTypes(USHRT\_MAX);  cout << "\nМинимальное беззнаковое:\n";  PrintTypes(0);  cout << "\nМаксимальное со знаком:\n";  PrintTypes(SHRT\_MAX);  cout << "\nМинимальное со знаком:\n";  PrintTypes(SHRT\_MIN);  system("Pause");  return 0;  } |
|  |

|  |
| --- |
|  |
| #include <iostream>  #include <cmath>  #include <bitset>  using namespace std;  template<typename T>  void PrintTypes(T Number)  {  cout << "hex: " << hex <<(T)Number << endl ;  cout << "dec: " << dec << (T)Number << endl;  }  template <typename T>  T Conjunction\_(T x,T y) // конъюнкция  {  return(x & y);  }  template <typename T>  T Disjunction\_(T x, T y) // дизъюнкция  {  return(x|y);  }  template <typename T>  T XOR(T x, T y) // сложение по модулю 2  {  return(x ^ y);  }  template <typename T>  T NOT(T value) // отрицание  {  return(~value);  }  template <typename T>  T BitLeft(T x,T y) // отрицание  {  return(x<<y);  }  template <typename T>  T BitRight(T x, T y) // отрицание  {  return(x >> y);  }  template <typename T>  T convert(T a) {    a = (-a);  return a;  }  int main()  {  setlocale(LC\_ALL, "Russian");  cout << endl << "---------------------------------------UNSIGNED OPERATIONS---------------------------------------" << endl;  //binary operations  /\*cout << endl<<"BINARY OPERATIONS"<<endl;\*/  unsigned short int xB1, yB1, xB2, yB2;  xB1 = 0x8008;  yB1 = 0x1111;  xB2 = 0x0005;  yB2 = 0x0002;    cout << endl << "x1 = " <<hex <<xB1 <<" = " <<dec<< xB1<<endl;  cout <<"y1 = " << hex<<yB1 << " = "<<dec<<yB1 << endl;  cout << endl << "x2 = " <<hex<< xB2<<" = "<<dec<<xB2;  cout << endl << "y2 = " <<hex <<yB2<<" = "<< dec<<yB2<< endl;  cout <<endl <<"Conjuction" << endl << "sys 1---------" << endl;  PrintTypes(Conjunction\_(xB1, yB1));  cout << "sys 2---------" << endl;  PrintTypes(Conjunction\_(xB2, yB2));  cout <<endl <<"Disjunction" << endl<<"sys 1---------"<<endl;  PrintTypes((unsigned short int)Disjunction\_(xB1, yB1));  cout << "sys 2---------" << endl;  PrintTypes((unsigned short int)Disjunction\_(xB2, yB2));  cout << endl<<"XOR" << endl;  PrintTypes(XOR(xB1, yB1));  cout << "sys 2---------" << endl;  PrintTypes(XOR(xB2, yB2));  cout <<endl<<"NOT" << endl<<"sys 1---------"<<endl;  PrintTypes(NOT(xB1));  //PrintTypes(NOT(yB1));  cout <<"sys 2---------" << endl;  PrintTypes(NOT(xB2));  //PrintTypes(NOT(yB2));  cout <<endl<<"CONVERT" << endl<<"sys 1---------"<<endl;  PrintTypes(convert(xB1));  //PrintTypes(convert(yB1));  cout <<"sys 2---------" <<endl;  PrintTypes(convert(xB2));  //PrintTypes(convert(yB2));  cout <<endl<<"Bit left" << endl << "sys 1---------" << endl;  PrintTypes(BitLeft(xB1, yB1));  cout << "sys 2---------" << endl;  PrintTypes(BitLeft(xB2, yB2));  cout <<endl<<"Bit right" << endl << "sys 1---------" << endl;  PrintTypes(BitRight(xB1, yB1));  cout << "sys 2---------" << endl;  PrintTypes(BitRight(xB2, yB2));  //  cout << endl << "---------------------------------------SIGNED OPERATIONS---------------------------------------" << endl;  //binary operations  //cout << endl << "BINARY OPERATIONS" << endl;  signed short int xB1\_, yB1\_, xB2\_, yB2\_;  xB1\_ = 0x8008;  yB1\_ = 0x1111;  xB2\_ = 0x0005;  yB2\_ = 0x0002;  cout << endl << "x1 = " << hex << xB1\_ << " = " << dec << xB1\_ << endl;  cout << "y1 = " << hex << yB1\_ << " = " << dec << yB1\_ << endl;  cout << endl << "x2 = " << hex << xB2\_ << " = " << dec << xB2\_;  cout << endl << "y2 = " << hex << yB2\_ << " = " << dec << yB2\_ << endl;  cout << endl << "Conjuction" << endl << "sys 1---------" << endl;  PrintTypes(Conjunction\_(xB1\_, yB1\_));  cout << "sys 2---------" << endl;  PrintTypes(Conjunction\_(xB2\_, yB2\_));  cout <<endl<<"Disjunction" << endl << "sys 1---------" << endl;  PrintTypes(Disjunction\_(xB1\_, yB1\_));  cout << "sys 2---------" << endl;  PrintTypes(Disjunction\_(xB2\_, yB2\_));  cout <<endl<<"XOR" << endl << "sys 1---------" << endl;  PrintTypes(XOR(xB1\_, yB1\_));  cout << "sys 2---------" << endl;  PrintTypes(XOR(xB2\_, yB2\_));  cout <<endl<<"NOT" << endl << "sys 1---------" << endl;  PrintTypes(NOT(xB1\_));  //PrintTypes(NOT(yB1\_));  cout << "sys 2---------" << endl;  PrintTypes(NOT(xB2\_));  //PrintTypes(NOT(yB2\_));  cout <<endl<<"CONVERT" << endl << "sys 1---------" << endl;  PrintTypes(convert(xB1\_));  //PrintTypes(convert(yB1\_));  cout << "sys 2---------" << endl;  PrintTypes(convert(xB2\_));  //PrintTypes(convert(yB2\_));  cout <<endl<<"Bit left" << endl << "sys 1---------" << endl;  PrintTypes(BitLeft(xB1\_, yB1\_));  cout << "sys 2---------" << endl;  PrintTypes(BitLeft(xB2\_, yB2\_));  cout <<endl<<"Bit right" << endl << "sys 1---------" << endl;  PrintTypes(BitRight(xB1\_, yB1\_));  cout << "sys 2---------" << endl;  PrintTypes(BitRight(xB2\_, yB2\_));  system("Pause");  return 0;  } |
| ---------------------------------------UNSIGNED OPERATIONS---------------------------------------  x1 = 8008 = 32776  y1 = 1111 = 4369  x2 = 5 = 5  y2 = 2 = 2  Conjuction  sys 1---------  hex: 0  dec: 0  sys 2---------  hex: 0  dec: 0  Disjunction  sys 1---------  hex: 9119  dec: 37145  sys 2---------  hex: 7  dec: 7  XOR  hex: 9119  dec: 37145  sys 2---------  hex: 7  dec: 7  NOT  sys 1---------  hex: 7ff7  dec: 32759  sys 2---------  hex: fffa  dec: 65530  CONVERT  sys 1---------  hex: 7ff8  dec: 32760  sys 2---------  hex: fffb  dec: 65531  Bit left  sys 1---------  hex: 0  dec: 0  sys 2---------  hex: 14  dec: 20  Bit right  sys 1---------  hex: 0  dec: 0  sys 2---------  hex: 1  dec: 1  ---------------------------------------SIGNED OPERATIONS---------------------------------------  x1 = 8008 = -32760  y1 = 1111 = 4369  x2 = 5 = 5  y2 = 2 = 2  Conjuction  sys 1---------  hex: 0  dec: 0  sys 2---------  hex: 0  dec: 0  Disjunction  sys 1---------  hex: 9119  dec: -28391  sys 2---------  hex: 7  dec: 7  XOR  sys 1---------  hex: 9119  dec: -28391  sys 2---------  hex: 7  dec: 7  NOT  sys 1---------  hex: 7ff7  dec: 32759  sys 2---------  hex: fffa  dec: -6  CONVERT  sys 1---------  hex: 7ff8  dec: 32760  sys 2---------  hex: fffb  dec: -5  Bit left  sys 1---------  hex: 0  dec: 0  sys 2---------  hex: 14  dec: 20  Bit right  sys 1---------  hex: ffff  dec: -1  sys 2---------  hex: 1  dec: 1 |

|  |
| --- |
|  |
| #include <iostream>  #include <typeinfo>  using namespace std;  int main()  {  setlocale(LC\_ALL, "Russian");    unsigned short int t = 0XEE22;  unsigned int t1 = t;  cout << "unsigned short int x:" << "\t10x: " << dec << t << "\t16x: " << hex << t << endl<<endl;  cout << "unsigned int x:" << "\t10x: " << dec << t1 << "\t16x: " << hex << t1 << endl << endl;    signed short int d = 0xEE22;  signed int d1 = d;  cout << "signed short int x:" << "\t10x: " << dec << d << "\t16x: " << hex << d << endl << endl;  cout << "signed int x:" << "\t10x: " << dec << d1 << "\t16x:" << hex << d1 << endl << endl;  system("Pause");  return 0;  } |
|  |

|  |
| --- |
|  |
| #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_ALL, "Russian");    int a= 0x12345678;  char a1[] = "abcd";  char a2[] = "абвг";  wchar\_t str1[] = L"abcd";  wchar\_t str2[] = L"абвг";  system("Pause");  return 0;  } |
| |  |  |  | | --- | --- | --- | | 0x12345678 | 78 56 34 12 | xV4. | | "abcd" | 61 62 63 64 00 | abcd. | | "абвг" | e0 e1 e2 e3 00 | абвг. | | L"abcd" | 61 00 62 00 63 00 64 00 00 00 | a.b.c.d... | | L"абвг" | 30 04 31 04 32 04 33 04 00 00 | 0.1.2.3... | |

|  |
| --- |
|  |
| #include <iostream>  #include <typeinfo>  using namespace std;  template<typename T>  void sizeVar() {  cout <<typeid(T).name()<<" have size = " <<sizeof(T)<<" bytes"<<endl;  }  int main()  {  setlocale(LC\_ALL, "Russian");        //cout << &a;  sizeVar<char>();  sizeVar<bool>();  sizeVar<wchar\_t>();  sizeVar<short>();  sizeVar<int>();  sizeVar<long>();  sizeVar<long long>();  sizeVar<float>();  sizeVar<double>();  sizeVar<long double>();  sizeVar<size\_t>();  sizeVar<ptrdiff\_t>();  sizeVar<void\*>();  system("Pause");  return 0;  } |
|  |

|  |
| --- |
|  |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | Win32 | | Win64 | | Ubuntu32 | | Ubuntu64 | | | 0x123456 | 78 56 34 12 | xV4. | 78 56 34 12 | xV4. | 78 56 34 12 | xV4. | 78 56 34 12 | xV4. | | “abcd” | 61 62 63 64 00 | abcd. | 61 62 63 64 00 | abcd. | 61 62 63 64 00 | abcd. | 61 62 63 64 00 | abcd. | | “абвг” | e0 e1 e2 e3 00 | абвг. | e0 e1 e2 e3 00 | абвг. | d0 b0 d0  b1 d0 b2  d0 b3 | Ð°Ð±Ð²Ð³ | d0 b0 d0 b1 d0  b2 d0 b3 | Ð°Ð±Ð²Ð³ | | L“abcd” | 61 00 62 00 63 00 64 00 | a.b.c.d. | 61 00 62 00 63 00 64 00 | a.b.c.d. | 61 00 00 00 62 00 00 00 63  00 00 00  64 00 00  00 | a...b...c...  d... | 61 00 00 00 62 00  00 00 63 00 00 00  64 00 00 00 | a...b...c...d... | | L“абвг” | 30 04 31 04 32 04 33 04 00 00 | 0.1.2.3. | 30 04 31 04 32 04 33 04 00 00 | 0.1.2.3. | 30 04 00  00 31 04  00 00 32  04 00 00  33 04 00  00 | 0...1...2...  3... | 30 04 00 00 31 04  00 00 32 04 00 00  33 04 00 00 | 0...1...2...3... | |

|  |
| --- |
|  |
|  |
|  |