



T3/2021
Hệ thống máy tính

BÁO CÁO

Bài tập về nhà 1



<i>MSSV</i>	<i>Họ và tên</i>
<i>19120583</i>	<i>Lê Thái Bình Minh</i>



Mục lục

I. Mức độ hoàn thành:	3
II. Bài tập 1:	3
1/ Chuyển hệ 10 sang hệ 2:	3
2/ Chuyển hệ 10 sang hệ 16:	4
3/ Chuyển hệ 2 sang hệ 10:	5
4/ Chuyển hệ 16 sang hệ 10:	6
5/ Chuyển hệ 2 sang hệ 16:	7
III. Bài tập 2:	8
1/ Chuyển đổi số nguyên từ hệ thập phân sang hệ nhị phân dạng bù 2:	8
2/ Chuyển đổi số nguyên từ hệ nhị phân sang hệ thập phân dạng bù 2:	10
3/ Tính toán các dãy bit:	11
3.1/ Cộng 2 dãy bits:	11
3.2/ Trừ 2 dãy bits:	11
3.3/ Nhân 2 dãy bits:	12
3.4/ Chia 2 dãy bits:	13
3.5/ Kết quả chạy các phép tính toán bits:	14
IV. Tài liệu tham khảo	15

I. Mức độ hoàn thành:

STT	YÊU CẦU	MỨC ĐỘ HOÀN THÀNH
1	Chuyển hệ 10 sang hệ 2	Đã hoàn thành
2	Chuyển hệ 10 sang hệ 16	Đã hoàn thành
3	Chuyển hệ 2 sang hệ 10	Đã hoàn thành
4	Chuyển hệ 16 sang hệ 10	Đã hoàn thành
5	Chuyển hệ 2 sang hệ 16	Đã hoàn thành
6	Chuyển hệ 10 sang hệ 2 (bù 2)	Đã hoàn thành
7	Chuyển hệ 2 sang hệ 10 (bù 2)	Đã hoàn thành
8	Các phép tính toán dãy 8 bits – bù 2 (cộng, trừ, nhân, chia)	Đã hoàn thành

II. Bài tập 1:

1/ Chuyển hệ 10 sang hệ 2:

DEC to BIN

- Code:

```
int ConvertDECtoBIN(int k)
{
    int dec = k;
    int tmp = 0;
    int i = 1;
    long int res = 0;
    do
    {
        tmp = dec % 2;
        res += (i * tmp);
        dec /= 2;
        i *= 10;
    } while (dec > 0);
    return res;
}
```

- Kết quả chạy:

```
int ConvertDECtoBIN(int k)
{
    int dec = k;
    int tmp = 0;
    int i = 1;
    long int res = 0;
    do
    {
        tmp = dec % 2;
        res += (i * tmp);
        dec /= 2;
        i *= 10;
    } while (dec > 0);
    return res;
}
```

C:\Users\User\source\repos\HTMT\ConvertNumbers\Debug\ConvertNumbers.exe
Give DEC: 56
DEC to BIN: 111000
Give DEC: _

2/ Chuyển hệ 10 sang hệ 16: DEC to HEX

- Code:

```
vector<char> ConvertDECtoHEX(int k)
{
    int dec = k;
    vector<char> res;
    int i = 0;
    while (dec > 0)
    {
        int tmp = 0;
        tmp = dec % 16;
        if (tmp < 10)
            res.push_back(tmp + 48);
        else res.push_back(tmp + 55);
        i++;
        dec /= 16;
    }
    return res;
}
```

- Kết quả chạy:

```
vector<char> ConvertDECtoHEX(int k)
{
    int dec = k;
    vector<char> res;
    int i = 0;
    while (dec > 0)
    {
        int tmp = 0;
        tmp = dec % 16;
        if (tmp < 10)
            res.push_back(tmp + 48);
        else res.push_back(tmp + 55);
        i++;
        dec /= 16;
    }
    return res;
}
```

C:\Users\User\source\repos\HTMT\ConvertNumbers\Debug\ConvertNumbers.exe

Give DEC: 56

DECtoHEX: 38

Give BIN:

3/ Chuyển hệ 2 sang hệ 10: BIN to DEC

- Code:

```
int ConvertNums(int k, int q)
{
    int give = k;
    int tmp = 1;
    int i = 0;
    int res = 0;
    while (give > 0)
    {
        tmp = give % 10;
        res += (pow(q, i)*tmp);
        i++;
        give /= 10;
    }
    return res;
}
```

- Kết quả chạy:

```
int ConvertNums(int k, int q)
{
    int give = k;
    int tmp = 1;
    int i = 0;
    int res = 0;
    while (give > 0)
    {
        tmp = give % 10;
        res += (pow(q, i)*tmp);
        i++;
        give /= 10;
    }
    return res;
}
```

C:\Users\User\source\repos\HTMT\Conv
Give BIN: 1001001
BINtoDEC: 73
Give HEX: _

4/ Chuyển hệ 16 sang hệ 10: HEX to DEC

- Code:

```
int ConvertHEXtoDEC(string s)
{
    int size = s.size();

    int temp = 1;
    int res = 0;
    for (int i = size - 1; i >= 0; i--)
    {
        if (s[i] >= '0' && s[i] <= '9')
            res += ((s[i] - 48) * temp);
        else res += ((s[i] - 55) * temp);
        temp *= 16;
    }
    return res;
}
```

- Kết quả chạy:

```
int ConvertHEXtoDEC(string s)
{
    int size = s.size();

    int temp = 1;
    int res = 0;
    for (int i = size - 1; i >= 0; i--)
    {
        if (s[i] >= '0' && s[i] <= '9')
            res += ((s[i] - 48) * temp);
        else res += ((s[i] - 55) * temp);
        temp *= 16;
    }
    return res;
}
```

C:\Users\User\source\repos\HTMT\ConvertNumbers\Debug\ConvertNumbers.exe
Give HEX: ABC
HEXtoDEC: 2748
Give BIN: _

5/ Chuyển hệ 2 sang hệ 16: BIN to HEX

- Code:

```
if (tam == "0011") ans += '3';
if (tam == "0100") ans += '4';
if (tam == "0101") ans += '5';
if (tam == "0110") ans += '6';
if (tam == "0111") ans += '7';
if (tam == "1000") ans += '8';
if (tam == "1001") ans += '9';
if (tam == "1010") ans += 'A';
if (tam == "1011") ans += 'B';
if (tam == "1100") ans += 'C';
if (tam == "1101") ans += 'D';
if (tam == "1110") ans += 'E';
if (tam == "1111") ans += 'F';
}
return ans;
}
```

```
string ConvertBINtoHEX(string s)
{
    int size = s.size();
    string s2 = "";
    for (int i = 0; i < s.size(); i++)
        s2 += s[i];

    /// <summary>
    /// Phan code sau dùng để fill các code sao cho đủ các cặp 4bits
    /// </summary>
    /// <param name="s2"></param>
    /// <returns><s2 sau khi gan = uwu thì sẽ là 1 dãy đủ cặp 4bits, vd: 100 => 0100>
    if (size % 4 != 0)
    {
        string uwu = "";
        for (int i = 0; i < (4 - size % 4); i++)
            uwu += '0';
        uwu += s2;
        s2 = uwu;
    }

    string ans;
    for (int i = 0; i < s2.size(); i += 4)
    {
        string tam = "";
        for (int j = i; j < i + 4; j++)
            tam += s2[j];
        if (tam == "0000") ans += '0';
        if (tam == "0001") ans += '1';
        if (tam == "0010") ans += '2';
```

- Kết quả chạy

```

}

string ans;
for (int i = 0; i < s2.size(); i += 4)
{
    string tam = "";
    for (int j = i; j < i + 4; j++)
        tam += s2[j];
    if (tam == "0000") ans += '0';
    if (tam == "0001") ans += '1';
}

```

Microsoft Visual Studio Debug Console

```

HEXtoDEC: 2748
Give BIN: 11111
BINtoHEX: 1F

```

III. Bài tập 2:

Các thao tác xử lý bit trên số nguyên dạng bù 2 – 8 bits

1/ Chuyển đổi số nguyên từ hệ thập phân sang hệ nhị phân dạng bù 2:

- Code:
 - Đối với số nguyên dương:

```

1
2
3 string ConvertDECtoBIN8bits(int k)//ham chuyen thap phan sang nhi phan
4 {
5     int dec = abs(k);
6     int tmp = 0;
7     string res = "";
8
9     while (dec > 0)
10    {
11        string temp = "";
12        tmp = dec % 2;
13        temp = (tmp)+48;
14        res += temp;
15        dec /= 2;
16    }
17    string s2 = "";
18    for (int i = res.size() - 1; i >= 0; i--)
19        s2 += res[i];
20    s2 = fitBits(s2);
21    return s2;
22 }
23

```


- Đối với số nguyên âm:

```

106 void ConvertDECtoBIN_2Complement(int n)//ham chuyen thap phan qua nhi phan voi so am
107 {
108     if (n >= -128 && n <= 127)
109     {
110         if (n < 0)
111         {
112             string a = ConvertDECtoBIN8bits(n);
113             a = fitBits(a);
114             cal_1_Complement(a);
115             if (n < 0)
116                 cal_2_Complement(a);
117             cout << a;
118             cout << "\n";
119         }
120         else cout << ConvertDECtoBIN8bits(n);
121     }
122     else cout << "CAN'T CONVERT!\n";
123 }
124

```

- Kết quả chạy:

CalculateBits.cpp

Source.cpp

Global Scope

```

//19120583
#include "SupeLib.h"

int main()
{
    int n;
    cout << "Nhap so thap phan: ";
    cin >> n;
    cout << "Chuyen qua he nhi phan: ";
    ConvertDECtoBIN_2Complement(n);

    string s;
    cout << "\nNhap so nhi phan: ";
    cin >> s;
    cout << "Chuyen qua he thap phan: ";
    cout << ConvertBINtoDEC(s);
}

```

C:\Users\User\source\repos\HTMT\BT2_Calculate

Nhap so thap phan: 3
Chuyen qua he nhi phan: 00000011
Nhap so nhi phan: 11111101
Chuyen qua he thap phan: -3
Nhap 2 so nhi phan:

2/ Chuyển đổi số nguyên từ hệ nhị phân sang hệ thập phân dạng bù 2:

- Code:

```
123 int ConvertBINtoDEC(string s)//ham chuyen nhi phan sang thap phan
124 {
125     int res = 0;
126     for (int i = 1; i < 8; i++)
127     {
128         int tmp = s[i] - '0';
129         res += (pow(2, 8 - i - 1)) * tmp;
130     }
131     if (s[0] - '0' == 1) res = res + (pow(-2, 7));
132     return res;
133 }
```

- Kết quả chạy:

```
cout << "Chuyen qua he thi phan: ";
ConvertDECtoBIN_2Complement(n);*/

string s;
cout << "\nNhap so nhi phan: ";
cin >> s;
cout << "Chuyen qua he thap phan: ";
cout << ConvertBINtoDEC(s);
//string s;
cout << "\nNhap so nhi phan: ";
cin >> s;
cout << "Chuyen qua he thap phan: ";
cout << ConvertBINtoDEC(s);
```

C:\Users\User\source\repos\HTMT\CalculateBits\Debug\Calcu

Nhap so nhi phan: 10001000
Chuyen qua he thap phan: -120
Nhap so nhi phan: 01010101
Chuyen qua he thap phan: 85
Nhap 2 so nhi phan: _

3/ Tính toán các dãy bit:

3.1/ Cộng 2 dãy bits:

```

40 string Add_Bit(string st, string nd)//Ham cong bit
41 {
42     string res = "";
43     int carry = 0;
44     fitBits(st);
45     fitBits(nd);
46     for (int i = 7; i >= 0; i--)
47     {
48         int firstBit = st[i] - '0';
49         int secondBit = nd[i] - '0';
50         int sum = (firstBit ^ secondBit ^ carry) + '0';
51
52         res = (char)sum + res;
53
54         carry = (firstBit & secondBit) | (secondBit & carry) | (firstBit & carry);
55     }
56
57     if (carry)
58         res = '1' + res;
59     res = fixSize(res);
60     return res;
61 }

```

3.2/ Trừ 2 dãy bits:

```

63 string Sub_Bit(string st, string nd)//ham tru bit
64 {
65     string res = "";
66     string stcp = "";
67     string ndcp = "";
68     stcp = fitBits(st);
69     ndcp = fitBits(nd);
70
71     cal_1_Complement(ndcp);
72     cal_2_Complement(ndcp);
73
74
75     res = Add_Bit(stcp, ndcp);
76     return fixSize(res);
77 }

```

3.3/ Nhân 2 dãy bits:

```

135 string Multi_Bit(string& st, string& nd)//ham nhan bit
136 {
137     int size = 8;
138     string C = "0";
139     string A = "00000000";
140     string Q = st;
141     string M = nd;
142     int check1 = ConvertBINtoDEC(st);
143     int check2 = ConvertBINtoDEC(nd);
144     if (check1 * check2 <= 127 && check1 * check2 >= -128)
145     {
146         while (size > 0)
147         {
148             if (Q[7] == '1' && C[0] == '0')
149             {
150                 A = Sub_Bit(A, M);
151             }
152             else
153             if (Q[7] == '0' && C[0] == '1')
154             {
155                 A = Add_Bit(A, M);
156             }
157             //SHR
158             C[0] = Q[7];
159             for (int i = 7; i > 0; i--)
160                 Q[i] = Q[i - 1];
161             Q[0] = A[7];
162             for (int i = 7; i > 0; i--)
163                 A[i] = A[i - 1];
164             size--;
165         }
166
167         string ans = "";
168         ans += A;
169         ans += Q;
170         ans = fixSize(ans);
171         return ans;
172     }
173     else {
174         cout << "\nCant Multiply!";
175         return "";
176     }
177 }
178

```

3.4/ Chia 2 dãy bits:

```

179 string Div_Bit(string& st, string& nd)//ham chia bit
180 {
181     string A = "";
182     int sign = 0;
183
184     if (st[0] == '1')
185     {
186         st = ConvertNegaBintoPosiBin(st);
187         sign++;
188     }
189     if (nd[0] == '1')
190     {
191         nd = ConvertNegaBintoPosiBin(nd);
192         sign++;
193     }
194
195     if (st[0] == '1')
196         A = "11111111";
197     else A = "00000000";
198     int size = 8;
199     string Q = st;
200     string M = nd;
201     while (size > 0)
202     {
203         //SHL
204         for (int i = 0; i < 7; i++)
205             A[i] = A[i + 1];
206         A[7] = Q[0];
207         for (int i = 0; i < 7; i++)
208             Q[i] = Q[i + 1];
209         //Q[7] = '0';

```

```

210     A = Sub_Bit(A, M);
211     if (A[0] == '1')
212     {
213         Q[7] = '0';
214         A = Add_Bit(A, M);
215     }
216     else Q[7] = '1';
217     size--;
218 }
219
220 if (sign == 0 || sign == 2)
221 {
222     cout << "\nDu: " << A << "\nThuong: ";
223     return Q;
224 }
225 else
226 {
227     A = ConvertPosiBintoNegaBin(A);
228     cout << "\nDu: " << A << "\nThuong: ";
229     Q = ConvertPosiBintoNegaBin(Q);
230     return Q;
231 }
232 }

```

3.5/ Kết quả chạy các phép tính toán bits:

```

17
18     string s1, s2;
19     cout << "\nNhap 2 so nhi phan: ";
20     cin >> s1;
21     cin >> s2;
22     cout << "Tong: ";
23     cout << Add_Bit(s1, s2);
24     cout << "\nHieu: ";
25     cout << Sub_Bit(s1,s2);
26     cout << "\nTich: ";
27     cout << Multi_Bit(s1,s2);
28     //cout << "\nThuong: ";
29     cout << Div_Bit(s1, s2);
30
31     return 0;
32 }
33

```

Microsoft Visual Studio Debug Console

```

Nhap so nhi phan: 00000100
Chuyen qua he thap phan: 4
Nhap 2 so nhi phan: 11110001
00000100
Tong: 11110101
Hieu: 11101101
Tich: 1111111111000100
Du: 11111101
Thuong: 11111101
C:\Users\User\source\repos\HTMT\BT2_CalculateBits\
To automatically close the console when debugging
le when debugging stops.
Press any key to close this window . . .

```

IV. Tài liệu tham khảo

Trong quá trình làm bài tập, em đã tham khảo các nguồn tài liệu sau:

1. *Slide trên Moodle.*
2. [Performing Arithmetic on Bitsets | Science and Mathematics \(flylib.com\)](#)
3. Website Geekforgeeks.