

BÁO CÁO MIDTERM PROJECT

THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Lớp: Thực hành Kiến trúc máy tính

Mã học phần: IT3280

Mã lớp: 130999

Đề tài 5, 11

Nhóm: 4

Phạm Đức Huy 20194588

Nguyễn Thị Thúy 20194685

Sinh viên thực hiện:

Phạm Đức Huy - 20194588

Phần 1: Báo cáo Mini project 11

Đề bài:

Programming an application to convert names from LastName-FirstName to FirstName-LastName.

1. Input name of 2 students with space character, for example “Vu Thi XYZ”. Store them into the memory.
2. Change them from LastName-FirstName to FirstName-LastName, for example “XYZ Vu Thi”
3. Print names to the screen.

Phân tích:

1. Yêu cầu:

Input: Họ và tên 2 sinh viên

Output: 2 tên sau khi đảo lộn tên lên trước, họ và tên đệm xuống dưới

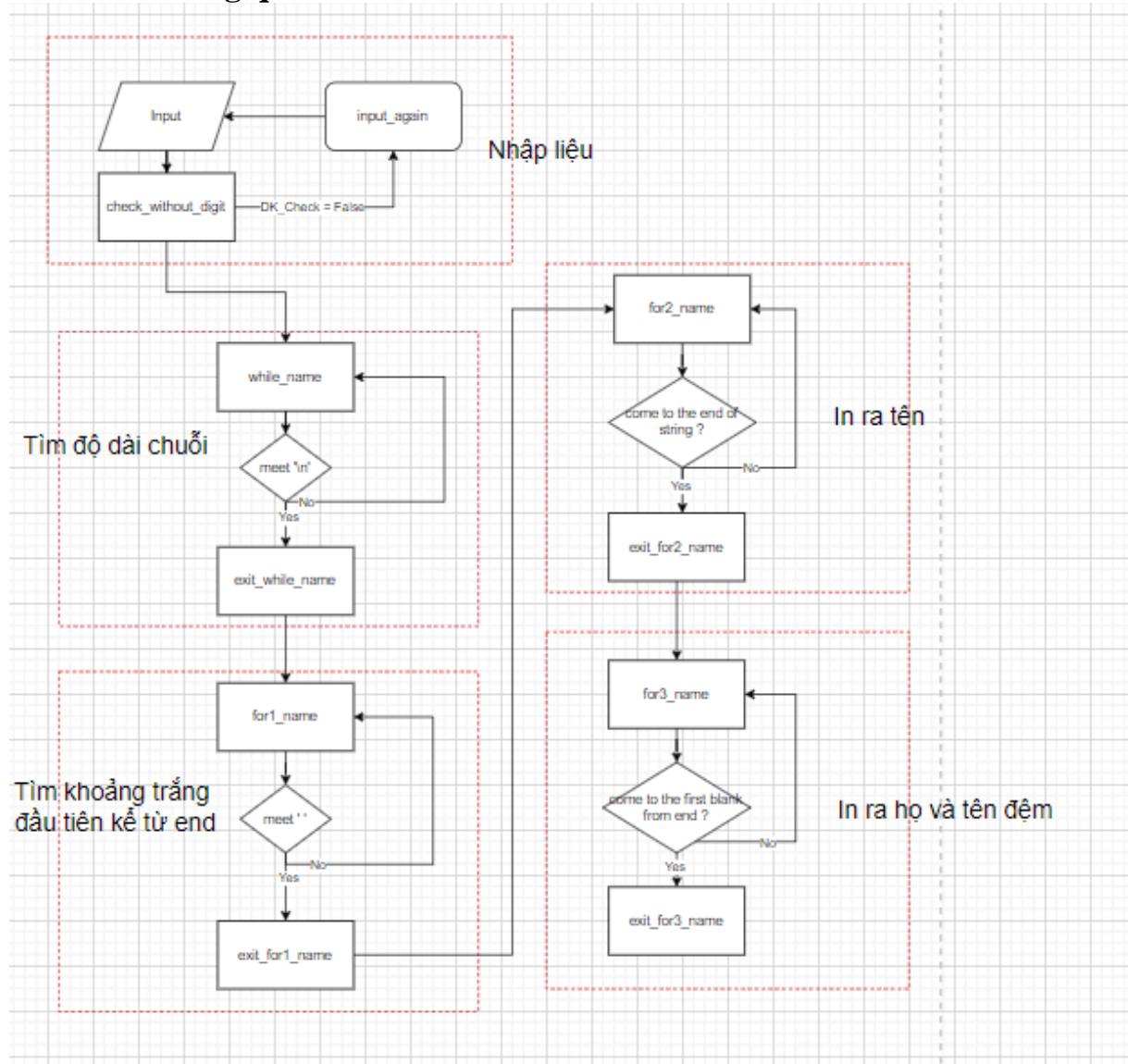
2. Cách thực hiện:

- B1: Yêu cầu nhập vào tên 2 sinh viên
- B2: Kiểm tra điều kiện tất cả ký tự trong tên phải là chữ cái, nếu vi phạm => đưa ra thông báo và yêu cầu nhập lại.
- B3: Xử lý xâu (xâu 1 và xâu 2 tương tự nhau):
 - Tìm độ dài của xâu
 - Tìm khoảng trắng đầu tiên kể từ vị trí kết thúc xâu trở về
 - In ra tên bằng cách in các ký tự từ vị trí khoảng trắng trên đến vị trí kết thúc xâu.
 - In ra họ tên và tên đệm bằng cách in từ đầu đến khoảng trắng trên

Phân tích cách thực hiện:

*Note: Cả 2 xâu họ tên đều được xử lý theo cùng 1 quy trình như phần trình bày dưới đây.

Sơ đồ khối tổng quát cách làm:



Giải thích chi tiết từng khối:

1. Khối nhập liệu: Có 3 chương trình chính: input, input_again, check_without_digit

- input:
 - In ra yêu cầu nhập chuỗi với li \$v0 4

```
li $v0, 4
la $a0, mess_input_1
syscall
```
 - Đọc chuỗi từ bàn phím với li \$v0 8, địa chỉ của chuỗi được nhập lưu tại string1

```
li $v0, 8
la $a0, string1
li $a1, 100
syscall
```
- input_again:
 - Đưa ra yêu cầu nhập lại chuỗi với li \$v0 4

```
li $v0, 4
la $a0, mess_nhậpLai
syscall
```
 - Sau đó chương trình sẽ tự động đến hàm input để nhập liệu từ đầu
- check_without_digit và exit_check_without_digit:
 - Ý tưởng: Duyệt từ đầu xâu đến cuối xâu, nếu bắt gặp ký tự không phải chữ cái hay dấu cách thì yêu cầu nhập lại
 - Mã C và giải thích:

```

int i = 0;          // i là chỉ số ($t0)
char* a = address; // a chứa địa chỉ của xâu ($a1)
void check_without_digit(int i)
{
    a = a + i;      // *a = a[i], *a chứa địa chỉ thứ i của xâu ($a0 là a[i])
    if(*a == '\n') // điều kiện dừng là đến cuối xâu bắt gặp '\n'
    {
        exit_check_without_digit(); //đến hàm exit_check_without_digit()
    }
    if((a[i]<= 'z' && a[i] >= 'a') || (a[i] <= 'Z' && a[i] >= 'A') || a[i] == ' ') //kiểm tra điều kiện
    {
        i++; // nếu đúng thì tăng i và tiếp tục xét ở ký tự a[i+1]
        check_without_digit(i);
    }
    else
    {
        input_again(); // nếu không thỏa mãn thì yêu cầu nhập lại
    }
}

void exit_check_without_digit()
{
    // không làm gì cả
}

```

2. Khởi tìm độ dài chuỗi:

- **Ý tưởng:** Duyệt xâu từ đầu đến khi gặp ký tự ‘\n’ thì dừng lại, trong khi duyệt tăng biến i, và khi kết thúc sẽ thu được độ dài chuỗi
- **Mã C và giải thích:**

```

int i = 0;          // i là chỉ số ($t0)
char* a = address; // a chứa địa chỉ của xâu ($a1)
void while_name(int i)
{
    a = a + i;      // *a = a[i], *a chứa địa chỉ thứ i của xâu (a[i] lúc này là $a0)
    if(*a == '\n') // điều kiện dừng là đến cuối xâu bắt gặp '\n'
    {
        exit_while_name_1(i); //đến hàm exit_while_name_1(i) để trả về độ dài
    }
    i++; //tăng i
    while_name_1(i); //tiếp tục xét ở a[i+1]
}

int s; //s là độ dài xâu ($s0)
void exit_while_name(int i)
{
    s = i + 0; //cập nhật s
}

```

*) Trước khi đến khối sau, chương trình in ra chuỗi “Tên người thứ (i) sau khi xử lý:”

```
#---In ra mess_output_1-----  
    li $v0, 4  
    la $a0, mess_output_1  
    syscall
```

3. Khối tìm khoảng trắng đầu tiên kể từ end đổ về

- **Ý tưởng:** Bắt đầu duyệt ngược từ cuối xâu (vị trí độ dài xâu -1), khi bắt gặp ‘ ‘ thì dừng lại. Trong khi duyệt thì giảm dần biến i từ vị trí kết thúc. Sau khi dừng ta sẽ thu được khoảng trắng đầu tiên kể từ end đổ về.
- **Mã C và giải thích:**

```
int i = 0;           // i là chỉ số ($t0)  
char* a = address;  // a chứa địa chỉ của xâu ($a1)  
int s = length;     // s lưu độ dài ($s0)  
i = s - 1; // vì s là độ dài thực nên nếu muốn xét từ ký tự cuối thì i = s -1  
int s2; //là vị trí finish, nơi có khoảng trắng đầu kể từ end đổ về ($s2)  
void for1_name(int i)  
{  
    if(i<0)  
    {  
        exit_for1_name(); // nếu i < 0 thì thoát  
    }  
    a = a + i;          // *a = a[i], *a chứa địa chỉ thứ i của xâu (a[i] lúc này là $a0)  
    i = i -1;          // giảm i  
    if(*a != ' ') // điều kiện dừng là bắt gặp ' '  
    {  
        while_name_1(i); //tiếp tục xét ở a[i-1]  
    }  
    else  
    {  
        s2 = i + 1; //s2 là vị trí cần tìm, cập nhật s2 rồi out khỏi hàm  
        break;  
    }  
}  
}  
void exit_for1_name()  
{  
    // không làm gì  
}
```

4. Khối in ra tên

- **Ý tưởng:** In từ vị trí khoảng trắng đầu tiên kể từ end (tìm thấy ở khối 3) cho đến vị trí kết thúc xâu (dựa vào độ dài tìm được ở khối 2)
- **Mã C và giải thích:**

```
int i = 0;           // i là chỉ số ($t0)
char* a = address;  // a chứa địa chỉ của xâu ($a1)
int s = length;     // s lưu độ dài (#s0)
int s2 = finish;    // là vị trí finish, nơi có khoảng trắng đầu kể từ end đổ về ($s2)
i = s2 + 1; // i sẽ xét từ vị trí chữ cái đầu tiên trong tên
void for2_name(int i)
{
    if(i == s)
    {
        exit_for2_name(); // nếu i == s (đến cuối xâu) thì thoát
    }
    a = a + i;           // *a = a[i], *a chứa địa chỉ thứ i của xâu (a[i] lúc này là $a0)
    printf("%c", a[i]) // in ra ký tự thứ i ra màn hình
    i = i + 1; //tiếp tục tăng i
    for2_name(i); //tiếp tục xét ở a[i+1]
}
void exit_for2_name()
{
    print(" "); //in ra ký hiệu space
}
```

5. Khối in ra họ và tên đệm

- **Ý tưởng:** In từ vị trí đầu tiên đến vị trí khoảng trắng đầu tiên kể từ end đổ về (được tìm thấy ở khối 3)
- **Mã C và giải thích:**

```
int i = 0;           // i là chỉ số ($t0)
char* a = address;  // a chứa địa chỉ của xâu ($a1)
int s2 = finish;    // là vị trí finish, nơi có khoảng trắng đầu kể từ end đổ về ($s2)
i = 0; // i sẽ xét từ vị trí chữ cái đầu tiên trong xâu
void for3_name(int i)
{
    if(i == s2)
    {
        exit_for3_name(); // nếu i == s2 (đến vị trí finish, nơi bắt đầu tên) thì thoát
    }
    a = a + i;           // *a = a[i], *a chứa địa chỉ thứ i của xâu (a[i] lúc này là $a0)
    printf("%c", a[i]) // in ra ký tự thứ i ra màn hình
    i = i + 1; //tiếp tục tăng i
    for3_name(i); //tiếp tục xét ở a[i+1]
}
void exit_for3_name()
{
    // không làm gì cả
}
```

Ý nghĩa các thanh ghi:

- Khi kiểm tra điều kiện nhập liệu:
 - \$t0: Chỉ số i
 - \$a1: Địa chỉ tại chỉ số i của chuỗi string (Địa chỉ của a[i])
 - \$a0: Chứa byte của địa chỉ lưu tại a1 (Giá trị của a[i])
 - Lúc đầu khởi tạo \$t2, \$t3, \$t4, \$t5, \$t6 lần lượt là 'z', 'a', 'Z', 'A', ' '.
 - Sau đó các biến tạm \$t được lần lượt được gán 0,1 tương ứng với sự thỏa mãn điều kiện check.
- Khi xử lý các chuỗi:
 - \$t0: Chỉ số i
 - \$a1: Địa chỉ tại chỉ số i của chuỗi string (Địa chỉ của a[i])
 - \$a0: Chứa byte của địa chỉ lưu tại a1 (Giá trị của a[i])
 - \$s0: Độ dài string
 - \$s2: Lưu chỉ số vị trí Finish

Ý nghĩa của các chương trình con:

- Input: Đưa ra yêu cầu nhập họ tên cho từng sinh viên và đọc xâu nhập vào từ bàn phím
- Input_again: In ra yêu cầu nhập lại khi nhập sai định dạng (chứa số hoặc ký tự đặc biệt (ngoài ' '))
- check_without_digit_1 và exit_check_without_digit_1: Kiểm tra trong tên học sinh 1 chỉ có chữ cái hay không
- while_name_1 và exit_while_name_1: Tìm độ dài chuỗi tên 1
- for1_name_1 và exit_for1_name_1: Duyệt từ cuối xâu lên để tìm vị trí Finish (Là vị trí dấu cách cuối cùng của chuỗi tên 1)
- for2_name_1 và exit_for2_name_1: In ra tên (Từ vị trí Finish+1 đến hết)
- for3_name_1 và exit_for3_name_1: In ra họ và tên đệm (Từ đầu đến vị trí Finish)
- check_without_digit_2 và exit_check_without_digit_2: Kiểm tra trong tên học sinh 2 chỉ có chữ cái
- while_name_2 và exit_while_name_1: Tìm độ dài chuỗi tên 2
- for1_name_2 và exit_for1_name_2: Duyệt từ cuối xâu lên để tìm vị trí Finish (Là vị trí dấu cách cuối cùng của chuỗi tên 2)
- for2_name_2 và exit_for2_name_2: In ra tên (Từ vị trí Finish+1 đến hết)
- for3_name_2 và exit_for3_name_2: In ra họ và tên đệm (Từ đầu đến vị trí Finish)

Kết quả thực hiện:

TH1: Cả 2 tên bình thường, gồm 3 chữ:

- Input:

```
Nhap ten nguoi thu 1: Pham Duc Huy
Nhap ten nguoi thu 2: Nguyen Thi Thuy
```

- Output:

```
Ten nguoi thu 1 sau xu li: Huy Pham Duc
Ten nguoi thu 2 sau xu li: Thuy Nguyen Thi
```

TH2: Tên có > 3 chữ:

- Input:

```
Nhap ten nguoi thu 1: Pham Duc Duong Huy
Nhap ten nguoi thu 2: Nguyen Thi Thu Thanh Thuy
```

- Output:

```
Ten nguoi thu 1 sau xu li: Huy Pham Duc Duong
Ten nguoi thu 2 sau xu li: Thuy Nguyen Thi Thu Thanh
```

TH3: Tên có chứa số:

- Input:

```
Nhap ten nguoi thu 1: Pham Duc Huy
Nhap ten nguoi thu 2: Ngulen T90 T0192
```

- Output:

Màn hình in ra lỗi và yêu cầu nhập lại từ đầu:

```
Nhap sai cu phap
Yeu cau nhap lai
Nhap ten nguoi thu 1: |
```

TH4: Tên có chứa ký tự đặc biệt:

- Input:

```
Nhap ten nguoi thu 1: Pham Duc Huy
Nhap ten nguoi thu 2: Ng!!?n T*& T/.)
```

- Output:

Màn hình in ra lỗi và yêu cầu nhập lại từ đầu:

```
Nhap sai cu phap
Yeu cau nhap lai
Nhap ten nguoi thu 1: |
```


Mã nguồn:

```
1  # Group 4: Phạm Đức Huy - Nguyễn Thị Thuý
2  .data
3      string1: .space 100
4      string2: .space 100
5      mess_input_1: .asciiz "Nhập tên người thứ 1: "
6      mess_input_2: .asciiz "Nhập tên người thứ 2: "
7      mess_output_1: .asciiz "Tên người thứ 1 sau xử lý: "
8      mess_output_2: .asciiz "Tên người thứ 2 sau xử lý: "
9      mess_nhậpLại: .asciiz "Nhập sai cú pháp\nYêu cầu nhập lại\n"
10     space: .asciiz " "
11     newLine: .asciiz "\n"
12 .text
13
14 j input
15
16 input_again:
17     li $v0, 4                                # In ra yêu cầu nhập lại
18     la $a0, mess_nhậpLại
19     syscall
20
21 #---Nhập và đọc chuỗi-----
22 input:
23     li $v0, 4                                # in ra yêu cầu nhập chuỗi
24     la $a0, mess_input_1
25     syscall
26
27     li $v0, 8                                # đọc chuỗi từ bàn phím
28     la $a0, string1
29     li $a1, 100
30     syscall
31
32     li $v0, 4                                # in ra yêu cầu nhập chuỗi
33     la $a0, mess_input_2
34     syscall
35
36     li $v0, 8                                # đọc chuỗi từ bàn phím
37     la $a0, string2
38     li $a1, 100
39     syscall
40
41 li $t0, 0                                    # i=0
42 check_without_digit_1:
43     la $a1, string1                          # load address string vào a1
44     add $a1, $a1, $t0                        # $a1 chưa địa chỉ cơ sở của string
45                                             # thời điểm bắt đầu: $a1 chưa địa chỉ
46                                             # tại vị trí đầu tiên của string
47     lb $a0, 0($a1)
48
49     #li $v0, 1                                # in integer
50     #syscall
51
52     beq $a0, 0x0a, exit_check_without_digit_1 # nếu phần tử a[1] = '\n' thì exit (0x0a = '\n')
53     li $t2, 'z'
54     li $t3, 'a'
55     li $t4, 'Z'
56     li $t5, 'A'
57     li $t6, ' '
58
59     sle $t8, $a0, $t6                        #2 dòng này để xét a0 = ' '
60     sle $t9, $t6, $a0
61
62     sle $t1, $a0, $t2                        # a0 <='z' -> t1 = 1
63     sle $t2, $t3, $a0                        # a0 >='a' -> t2 = 1
```

61	sle \$t3, \$a0, \$t4	# a0 <='Z' -> t3 =1
62	sle \$t4, \$t5, \$a0	# a0 >='A' -> t4 =1
63	and \$t5, \$t1, \$t2	# t5 = a0 <= 'z' && a0 >= 'a'
64	and \$t6, \$t3, \$t4	# t6 = a0 <= 'Z' && a0 >= 'A'
65	and \$t7, \$t8, \$t9	# t7 = a0 == ' '
66	or \$t8, \$t6, \$t5	
67	or \$t9, \$t7, \$t8	# t9 = t5 t6 t7
68	bne \$t9, 1, input_again	# neu khong thoa man thi yeu cau nhap lai
69	addi \$t0, \$t0, 1	# i++
70		
71	j check_without_digit_1	
72	exit_check_without_digit_1:	# ket thuc vong while
73		
74	li \$t0, 0	# i=0
75	check_without_digit_2:	
76	la \$a1, string2	# load address string vao a1
77	add \$a1, \$a1, \$t0	# \$a1 chua dia chi co so cua string
78		# thoi diem bat dau: \$a1 chua dia chi
79		# tai vi tri dau tien cua string
80	lb \$a0, 0(\$a1)	
81		
82	#li \$v0, 1	# in integer
83	#syscall	
84		
85	beq \$a0, 0x0a, exit_check_without_digit_2	# neu phan tu a[i]='\n' thi exit (0x0a = '\n')
86	li \$t2, 'z'	
87	li \$t3, 'a'	
88	li \$t4, 'Z'	
89	li \$t5, 'A'	
90	li \$t6, ' '	
91	sle \$t8, \$a0, \$t6	#2 dong nay de xet a0 = ' '
92	sle \$t9, \$t6, \$a0	
93	sle \$t1, \$a0, \$t2	# a0 <='z' -> t1 =1
94	sle \$t2, \$t3, \$a0	# a0 >='a' -> t2 =1
95	sle \$t3, \$a0, \$t4	# a0 <='Z' -> t3 =1
96	sle \$t4, \$t5, \$a0	# a0 >='A' -> t4 =1
97	and \$t5, \$t1, \$t2	# t5 = a0 <= 'z' && a0 >= 'a'
98	and \$t6, \$t3, \$t4	# t6 = a0 <= 'Z' && a0 >= 'A'
99	and \$t7, \$t8, \$t9	# t7 = a0 == ' '
100	or \$t8, \$t6, \$t5	
101	or \$t9, \$t7, \$t8	# t9 = t5 t6 t7
102	bne \$t9, 1, input_again	# neu khong thoa man thi yeu cau nhap lai
103	addi \$t0, \$t0, 1	# i++
104		
105	j check_without_digit_2	
106	exit_check_without_digit_2:	# ket thuc vong while
107		
108	#----xu ly xau 1-----	
109	#---Tim do dai chuoai-----	
110	li \$t0, 0	# i=0
111	while_name_1:	# vong lap while tim do dai chuoai string
112	la \$a1, string1	# load address string vao a1
113	add \$a1, \$a1, \$t0	# \$a1 chua dia chi co so cua string
114		# thoi diem bat dau: \$a1 chua dia chi
115		# tai vi tri dau tien cua string
116	lb \$a0, 0(\$a1)	
117		
118	#li \$v0, 1	# in integer
119	#syscall	
120	beq \$a0, 0x0a, exit_while_name_1	# neu phan tu a[i]='\n' thi exit (0x0a = '\n')
121	addi \$t0, \$t0, 1	# i++
122	j while_name_1	
123	exit_while_name_1:	# ket thuc vong while

```

124         add $s0, $t0, $zero                # s0 luu do dai string
125
126 ---Duyet tu cuoi chuoi len dau tim vi tri FINISH---
127         addi $t0, $s0, -1                  # i= s - 1 (=strlen)
128     for1_name_1:
129         blt $t0, $zero, exit_for1_name_1    # if i<0 => exit
130         la $a1, string1                    # load address string vao a1
131         add $a1, $a1, $t0                  # $a1 chua dia chi co so cua string
132                                             # thoi diem bat dau: $a1 chua dia chi
133                                             # tai vi tri cuoi cung cua string
134         lb $a0, 0($a1)
135         addi $t0, $t0, -1                  # i--
136         bne $a0, 0x20, for1_name_1          # if a[i]!=' '
137         addi $s2, $t0, 1                  # $s2 = i + 1 , $s2 se luu vi tri FINISH
138     exit_for1_name_1:
139
140 ---In ra mess_output_1---
141         li $v0, 4
142         la $a0, mess_output_1
143         syscall
144
145 ---In ten tu Finish den het chuoi---
146         addi $t0, $s2, 1                  # i=finish+1
147     for2_name_1:
148         beq $t0, $s0, exit_for2_name_1      # if i == strlen => exit
149         la $a1, string1                    # load address string vao a1
150         add $a1, $a1, $t0                  # $a1 chua dia chi co so cua string
151                                             # thoi diem bat dau: $a1 chua dia chi
152                                             # tai vi tri cua ky tu dau tien trong ten
153                                             # (finish +1) cua string
154         lb $a0, 0($a1)
155         li $v0, 11                          # printf(a[i])
156         syscall
157         addi $t0, $t0, 1                  # i++
158         j for2_name_1
159     exit_for2_name_1:
160         li $v0, 4                          # in ra ki tu space " "
161         la $a0, space
162         syscall
163
164 ---In ho va ten dem tu 0 den FINISH---
165         addi $t0, $zero, 0                # i=0
166     for3_name_1:
167         beq $t0, $s2, exit_for3_name_1      # if i == FINISH => exit
168         la $a1, string1                    # load address string vao a1
169         add $a1, $a1, $t0                  # $a1 chua dia chi co so cua string
170                                             # thoi diem bat dau: $a1 chua dia chi
171                                             # tai vi tri dau tien cua string
172         lb $a0, 0($a1)
173         li $v0, 11                          # printf(a[i])
174         syscall
175         addi $t0, $t0, 1                  # i++
176         j for3_name_1
177     exit_for3_name_1:
178
179
180
181         li $v0, 4                          # in ra ki tu newLine "\n"
182         la $a0, newLine
183         syscall
184

```

```

185 #-----xy ly xau 2-----
186 #---Tim do dai chuoi-----
187     li $t0,0                                # i=0
188 while_name_2:                               # vong lap while tim do dai chuoi string
189     la $a1, string2                         # load address string vao a1
190     add $a1, $a1, $t0                      # $a1 chua dia chi co so cua string
191                                           # thoi diem bat dau: $a1 chua dia chi
192                                           # tai vi tri dau tien cua string
193     lb $a0, 0($a1)
194     beq $a0, 0x0a, exit_while_name_2        # neu phan tu a[i]='\n' 0x0a = '\n'
195     addi $t0,$t0, 1                         # i++
196     j while_name_2
197 exit_while_name_2:                          # ket thuc vong while
198     add $s0, $t0, $zero                    # s0 luu do dai string
199
200 #---Duyet tu cuoi chuoi len dau tim vi tri FINISH-----
201     addi $t0, $s0, -1                      # i=strlen
202 for1_name_2:
203     blt $t0, $zero, exit_for1_name_2        # if i<0 => exit
204     la $a1, string2                        # load address string vao a1
205     add $a1, $a1, $t0                      # $a1 chua dia chi co so cua string
206                                           # thoi diem bat dau: $a1 chua dia chi
207                                           # tai vi tri cuoi cung cua string
208     lb $a0, 0($a1)
209     addi $t0, $t0, -1                      # i--
210     bne $a0, 0x20, for1_name_2             # if a[i]!=' ' => Ti?p t?c l?p
211     addi $s2, $t0, 1                      # $s2 luu vi tri FINISH
212 exit_for1_name_2:
213
214 #---In ra mess_output_1-----
215     li $v0, 4
216     la $a0, mess_output_2
217     syscall
218
219 #---In ten tu Finish den het chuoi-----
220     addi $t0, $s2, 1                      # i=finish+1
221 for2_name_2:
222     beq $t0, $s0, exit_for2_name_2        # if i == strlen => exit
223     la $a1, string2                       #load address string vao a1
224     add $a1, $a1, $t0                     # $a1 chua dia chi co so cua string
225                                           # thoi diem bat dau: $a1 chua dia chi
226                                           # tai vi tri cua ky tu dau tien trong ten
227                                           # (finish+1) cua string
228     lb $a0, 0($a1)
229     li $v0, 11                             # printf(a[i])
230     syscall
231     addi $t0, $t0, 1                      # i++
232     j for2_name_2
233 exit_for2_name_2:
234     li $v0, 4                             # in ra ki tu space " "
235     la $a0, space
236     syscall
237
238 #---In ho va ten dem tu 0 den FINISH-----
239     addi $t0, $zero, 0                    # i=0
240 for3_name_2:
241     beq $t0, $s2, exit_for3_name_2        # if i == FINISH => finish
242     la $a1, string2                       #load address string vao a1
243     add $a1, $a1, $t0                     # $a1 chua dia chi co so cua string
244                                           # thoi diem bat dau: $a1 chua dia chi
245                                           # tai vi tri dau tien cua string
246     lb $a0, 0($a1)
247     li $v0, 11                             # printf(a[i])
248     syscall
249     addi $t0, $t0, 1                      # i++
250     j for3_name_2
251 exit_for3_name_2:
252

```

Phần 2: Báo cáo Mini project 5

Đề bài:

Write a program to get decimal numbers, display those numbers in binary and hexadecimal.

Phân tích:

3. Yêu cầu:

- Input: số nguyên hệ thập phân
- Output: số nguyên dạng nhị phân và hệ thập lục phân

4. Cách thực hiện:

- Nhập số nguyên hệ thập phân:
 - Đưa ra hộp thoại yêu cầu nhập số nguyên
 - Kiểm tra giá trị nhập vào, trong các trường hợp:
 - Nhập vào ký tự không phải số nguyên (chữ cái, số thực, ký tự đặc biệt, ...)
 - Nhập số lớn hơn số phạm vi cho phép (không thuộc $[-2^{31}; 2^{31} - 1]$)
 - ⇒ Đưa ra thông báo lỗi nhập và yêu cầu nhập lại
- In ra số nguyên dạng nhị phân
- In ra số nguyên dạng Hexa

Ý tưởng:

- Phần in nhị phân
 - Tạo mask với bit 1 ở bit cao nhất, tất cả bit còn lại là 0. Dịch bit dần về phía phải, mỗi khi AND với số đã nhập : nếu trả về 0 chứng tỏ vị trí đó bit là 0 ($1 \& 0 = 0$), nếu trả về khác 0 chứng tỏ vị trí đó bit là 1 ($1 \& 1 = 1$). In từng bit theo thứ tự từ bit cao đến thấp
- Phần in thập lục phân
 - Chuyển 4 bit cao về cuối rồi so sánh với 0xF. Lúc này giá trị trả về sẽ từ 0 -> 16. Do trong ascii ký tự 0 -> 9 ở vị trí 48 -> 57, ký tự A->F từ 65 -> 70 nên ta chia thành 2 trường hợp:
 - + nếu giá trị trả về từ 0 -> 9 thì ta thêm 48 rồi lưu nó vào kết quả
 - + nếu giá trị trả về từ 10 -> 15 thì ta thêm (65-10 = 55) vào kết quả

Ý nghĩa các thanh ghi:

- Khi nhập và kiểm tra đầu vào:
 - \$t0: Số đã nhập
 - \$s6: Status value

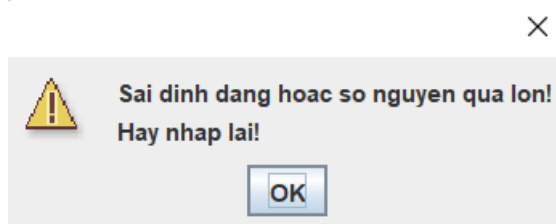
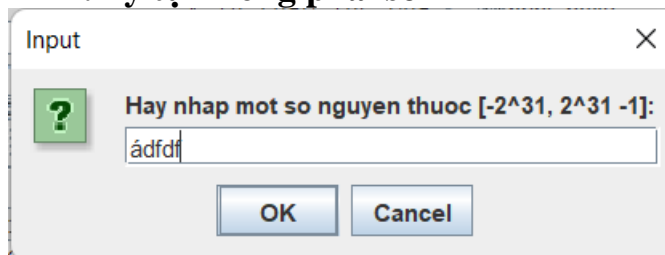
- Khi in số nhị phân:
 - \$t0: Số đã nhập
 - \$t2: Mask để check vị trí có bit 1
 - \$t1: 1 hoặc 0. Giá trị của bit tại từng vị trí
 - \$t3: Biến đếm
- Khi in số thập lục phân:
 - \$t0: Số đã nhập
 - \$t2: String đáp án
 - \$t1: Giá trị mã ASCII tương ứng biểu diễn ở thập lục phân
 - \$t3: Biến đếm

Ý nghĩa của các chương trình con:

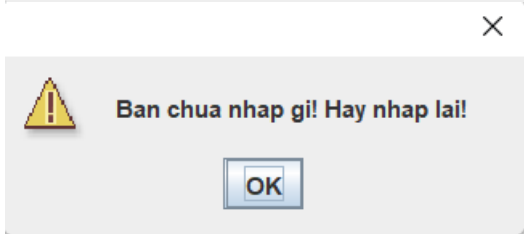
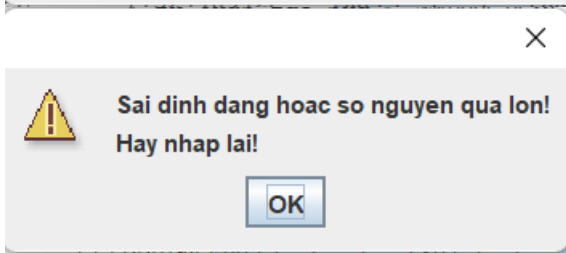
- main: hàm main
- endMain: thoát chương trình
- input_error & saiDinhDang & koNhap: tạo cảnh báo lỗi
- end_error: kết thúc tạo cảnh báo
- print_bin: khởi tạo hàm in nhị phân
- loop1: hàm lặp từng số trong print_bin
- print1: in kết quả nhị phân
- print_hexa: khởi tạo hàm in thập lục phân
- loop2: hàm lặp từng số trong print_hexa
- sum: chuyển sang mã ascii của ký tự từ 0 -> 9
- end: lặp lại vòng lặp vào in ký tự vào string kết quả
- exit: in kết quả và thoát hàm in thập lục phân

Kết quả thực hiện:


TH1: ký tự không phải số



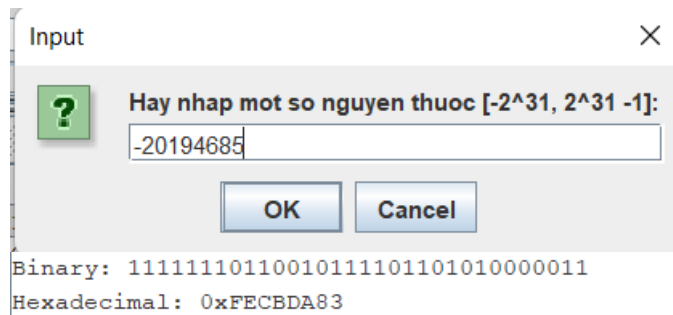
TH2: không nhập gì

[illegible]

Input ✕

 Hay nhập một số nguyên thuộc $[-2^{31}, 2^{31} - 1]$:

TH6: số nguyên âm



Mã nguồn:

```

1  #-----
2  # Project 5: Write a program to get decimal numbers, display those numbers in binary and hexadecimal.
3  #-----
4  .data
5      xauBit: .asciiz "\nBinary: "
6      xauHexa: .asciiz "\nHexadecimal: 0x"
7      nhapSo: .asciiz "Hay nhap mot so nguyen thuoc [-2^31, 2^31 -1]: "
8      loiDinhDang: .asciiz "Sai dinh dang hoac so nguyen qua lon!\nHay nhap lai!"
9      loiChuaNhap: .asciiz "Ban chua nhap gi! Hay nhap lai!"
10     dangHexa: .space 8
11 .text
12 main:
13     # Nhap va luu so
14     la      $a0, nhapSo          # Hien thi hop thoai nhap
15     li      $v0, 51
16     syscall
17     add     $s6, $0, $a1         # s6 = a1 = status value
18                                     # s6 == 0: OK
19     bnez    $s6, input_error     # status != 0 jump input_error
20     add     $t0, $a0, $0         # $t0 = so da nhap
21     jal     print_bin           # in nhi phan
22     jal     print_hexa         # in thap luc pha
23     j       main               # lap lai main cho toi khi an huy
24 endMain:
25     li      $v0, 10             # thoat
26     syscall
27 # ===== kiem tra dau vao =====#
28 input_error:
29     li      $a1, 2              # hop thoai canh bao
30     li      $v0, 55
31     beq     $s6, -2, endMain     # status == -2 => thoat
32     beq     $s6, -1, saiDinhDang # status == -1 => toi error1
33     beq     $s6, -3, koNhap      # status == -3 => toi error2
34 saiDinhDang:
35     la      $a0, loiDinhDang     # loi dinh dang
36     syscall
37     j       end_error
38 koNhap:
39     la      $a0, loiChuaNhap     # loi chua nhap
40     syscall
41     j       end_error
42 end_error:
43     j       main
44 # =====Print Binary===== #
45 print_bin:
46     la      $a0, xauBit         # "Binary"
47     li      $v0, 4
48     syscall
49
50     addi    $t2, $0, 1          # $t2 = 1
51     add     $t1, $0, $0         # $t1 = 0
52     sll     $t2, $t2, 31        # dich $t2 31 bit
53     addi    $t3, $0, 32        # bien dem
54 loop1:

```



```

55         and    $t1, $t0, $t2          # thuc hien AND so dau vao voi $t2
56         beq    $t1, $0, print1        # AND = 0 thi nhay toi print1
57         addi   $t1, $0, 1              # AND != 0 thi gan $t1 = 1
58 print1:
59         move   $a0, $t1
60         li     $v0, 1
61         syscall                                # in gia tri của $t1
62         srl    $t2, $t2, 1              # dich $t2 sang phai 1 bit
63         subi   $t3, $t3, 1              # t3--
64         bne    $t3, $0, loop1           # t3 != 0 => tiep tuc
65         jr     $ra                        # tro lai ham main
66 # =====Print Hexa===== #
67 print_hexa:
68         la     $a0, xauHexa              # "Hexadecimal"
69         li     $v0, 4
70         syscall
71         la     $t2, dangHexa              # store dap an
72         li     $t3, 8                    # bien dem
73 loop2:
74         beqz   $t3, exit                  # thoat neu dem = 0
75         rol    $t0, $t0, 4                # chuyen 4 bit cao ve cuoi
76         and    $t1, $t0, 0xf              # AND voi 1111
77         ble    $t1, 9, sum                # neu <= 9 thi vao sum
78         addi   $t1, $t1, 55                # neu > 9 thi +55 (A -> F)
79         j      end
80 sum:
81         addi   $t1, $t1, 48                # them 48 (0-> 9)
82 end:
83         sb     $t1, 0($t2)                # luu ky tu vao ket qua
84         addi   $t2, $t2, 1                # ketqua[i+1]
85         subi   $t3, $t3, 1                # counter--
86         j      loop2
87 exit:
88         la     $a0, dangHexa
89         li     $v0, 4
90         syscall
91         jr     $ra
92

```