

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



# BÁO CÁO CUỐI KỲ

## MÔN: THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Giảng viên hướng dẫn: Phạm Ngọc Hưng

Sinh viên thực hiện: Nguyễn Thị Minh Châu - 20214997

Lớp: 139365

Hà Nội, tháng 7 năm 2023

# MỤC LỤC

<b>MỤC LỤC</b>	<b>1</b>
<b>KẾT QUẢ THỰC HIỆN</b>	<b>2</b>
Bài 5: Biểu thức trung tố hậu tố	2
Phân tích cách làm và thuật toán	2
Mã nguồn	4
Kết quả chạy mô phỏng	11

## Bài 5: Biểu thức trung tố hậu tố

Viết chương trình tính giá trị biểu thức bất kỳ bằng phương pháp duyệt biểu thức hậu tố.

Các yêu cầu cụ thể:

1. Nhập vào biểu thức trung tố, ví dụ:

$$9 + 2 + 8 * 6$$

2. In ra biểu thức ở dạng hậu tố, ví dụ:  $9\ 2\ +\ 8\ 6\ *\ +$

3. Tính ra giá trị của biểu thức vừa nhập

Các hằng số là số nguyên, trong phạm vi từ  $0 \rightarrow 99$ .

Toán tử bao gồm các phép toán cộng, trừ, nhân, chia lấy thương (/), chia lấy dư (%), đóng mở ngoặc.

### A. Phân tích cách làm và thuật toán.

Để chuyển từ biểu thức trung tố sang biểu thức hậu tố, ta đọc lần lượt từng ký tự trong biểu thức trung tố, giả sử ký tự đọc được là c:

- + Nếu c là số  $\rightarrow$  thêm vào biểu thức hậu tố
- + Nếu c là toán tử  $\rightarrow$  thực hiện thêm toán tử vào stack. Nếu stack rỗng, thêm c vào stack, nếu stack không rỗng, so sánh thứ tự ưu tiên của c với toán tử ở đỉnh stack, nếu độ ưu tiên của  $c \leq$  độ ưu tiên của toán tử ở đỉnh stack, lấy toán tử ở đỉnh ra và thêm vào biểu thức hậu tố, lặp lại việc so sánh trên cho đến khi độ ưu tiên của  $c \geq$  độ ưu tiên của toán tử ở đỉnh stack, sau đó thêm c vào stack.
- + Nếu c là toán tử (  $\rightarrow$  thêm luôn c vào stack.
- + Nếu c là toán tử )  $\rightarrow$  lấy lần lượt các toán tử trong stack ra và thêm vào biểu thức hậu tố cho đến khi gặp toán tử ) . Lưu ý, không thêm toán tử ) vào trong biểu thức hậu tố.
- + Khi duyệt hết biểu thức trung tố, nếu còn toán tử trong stack, thực hiện lấy lần lượt các toán tử ra và thêm vào biểu thức trung tố cho đến khi stack rỗng.

Vì các toán hạng có giá trị từ  $0 \rightarrow 99$  nên khi thêm vào biểu thức hậu tố, cần thêm các dấu cách để ngăn cách giữa các phần tử.

Để tính giá trị biểu thức hậu tố:

- + Thực hiện duyệt biểu thức hậu tố, nếu gặp toán hạng thì thêm toán hạng vào stack, nếu gặp toán tử, lấy hai toán hạng trong stack ra và tính toán phép tính theo toán tử tương ứng, kết quả của phép tính được lưu trở lại vào trong stack. Lặp lại cho đến khi duyệt hết biểu thức hậu tố, kết quả cuối cùng được lưu trong stack.

Thứ tự ưu tiên của các toán tử:

Toán tử	Độ ưu tiên
(	0
+	1
-	1
*	2
/	2
%	2
)	3

## B. Mã nguồn

```
1  .data
2  bttrungto: .space 256
3  bthauto: .space 256
4  nganxep: .space 256
5  arr: .space 256
6
7  string: .asciiz "\n"
8  message1: .asciiz "Bieu thuc trung to: "
9  message2: .asciiz "Bieu thuc hau to: "
10 message3: .asciiz "Ket qua bieu thuc vua nhap: "
11 message4: .asciiz "Nhap vao bieu thuc trung to: "
12 message5: .asciiz "MENU\n1.Chay chuong trinh\n2.Thoat chuong trinh\n"
13 message6: .asciiz "Bieu thuc khong hop le.\n"
14 .text
15 main: la $a0, message5
16      li $v0, 4
17      syscall
18
19      li $v0, 5
20      syscall
21
22      beq $v0, 2, end_main
23
24      jal INPUT
25      nop
26      jal CHECK
27      nop
28      beq $v0, 0, next1
29      nop
30
31      jal OUTPUT1
32      nop
33      jal CONVERT
34      nop
35      jal OUTPUT2
36      jal CALCULATE
37      jal OUTPUT3
38      j main
39      nop
40 next1: la $a0, message6
41      li $v0, 4
42      syscall
43
44      j main
45 end_main:
46      li $v0, 10
47      syscall
48
49 #Nhap bieu thuc trung to
50 INPUT: la $a0, message4
51      li $v0, 4
52      syscall
53
54      li $v0, 8
55      la $a0, bttrungto
56      la $a1, 256
57      syscall
58
```

```

59         jr $ra
60         nop
61         #In bieu thuc trung to ra man hinh
62         OUTPUT1: la $a0, message1
63                 li $v0, 4
64                 syscall
65
66                 la $a0, bttrungto
67                 li $v0, 4
68                 syscall
69
70         jr $ra
71         nop
72         #In bieu thuc hau to ra man hinh
73         OUTPUT2: la $a0, message2
74                 li $v0, 4
75                 syscall
76
77                 la $a0, bthauto
78                 li $v0, 4
79                 syscall
80
81                 la $a0, string
82                 li $v0, 4
83                 syscall
84
85         jr $ra
86         nop
87         #In ket qua ra man hinh
88         OUTPUT3: la $a0, message3
89                 li $v0, 4
90                 syscall
91
92                 lb $a0, -1($s3)
93                 li $v0, 1
94                 syscall
95
96                 la $a0, string
97                 li $v0, 4
98                 syscall
99
100        jr $ra
101        nop
102        #Kiem tra tinh hop le cua bieu thuc vua nhap
103        CHECK: addi $sp, $sp, -4
104                sw $ra, 0($sp)
105                la $a0, bttrungto
106                li $t1, 0
107        load: add $t2, $t1, $a0
108                lb $t3, 0($t2)
109                beq $t3, 10, end_check
110                jal check_number
111                beq $v0, 1, continue
112                jal check_operator
113                beq $v0, 1, continue
114                jal check_space
115                beq $v0, 1, continue
116                j end_check

```

```

---
118 continue:
119     addi $t1, $t1, 1
120     j load
121
122 end_check:
123     lw $ra, 0($sp)
124     addi $sp, $sp, 4
125     jr $ra
126     nop
127 #CONVERT          : chuyen bieu thuc trung to thanh bieu thuc hau to
128 CONVERT:addi $sp, $sp, -4
129     sw $ra, 0($sp)
130     la $s1, bttrungto
131     la $s2, bthauto
132     la $s3, nganxep
133     li $s6, 0
134     li $s7, 0
135     li $t7, -1
136 while1:add $s4, $s1, $s6
137     lb $t3, 0($s4)
138     beq $t3, 10, end_while1
139     jal check_number
140     beq $v0, 1, push_number
141 next3:beq $t3, 10, end_while1
142     jal check_operator
143     beq $v0, 1, check_before_push_operator
144 next2:addi $s6, $s6, 1
145     j while1
146 push_number:add $s4, $s2, $s7
147
148 push_number:add $s4, $s2, $s7
147     sb $t3, 0($s4)
148     addi $s7, $s7, 1
149     addi $s6, $s6, 1
150     add $s4, $s1, $s6
151     lb $t3, 0($s4)
152     jal check_number
153     beq $v0, 1, push_number
154     li $t2, ' '
155     add $s4, $s2, $s7
156     sb $t2, 0($s4)
157     addi $s7, $s7, 1
158     j next3
159 check_before_push_operator:beq $s0, 0, push_operator1
160     beq $t7, -1, push_operator1
161     addi $t1, $t3, 0
162     addi $a0, $s0, 0
163 after_pop:beq $t7, -1, push_operator2
164     add $s4, $t7, $s3
165     lb $t3, 0($s4)
166     jal check_operator
167     ble $a0, $s0, pop_operator
168     j push_operator2
169 pop_operator:beq $t3, '(', before_pop_operator
170     add $s4, $s2, $s7
171     sb $t3, 0($s4)
172     addi $s7, $s7, 1
173     li $t2, ' '
174     add $s4, $s2, $s7

```

```

175         sb $t2, 0($s4)
176         addi $s7, $s7, 1
177         addi $t7, $t7, -1
178         j after_pop
179 before_pop_operator:addi $t7, $t7, -1
180         j push_operator2
181 push_operator1:addi $t7, $t7, 1
182         add $s4, $s3, $t7
183         sb $t3, 0($s4)
184         j next2
185 push_operator2:beq $t1, '}', after_pop2
186         addi $t7, $t7, 1
187         add $s4, $s3, $t7
188         sb $t1, 0($s4)
189         j next2
190 after_pop2:
191         add $s4, $t7, $s3
192         lb $t3, 0($s4)
193         addi $t7, $t7, -1
194         beq $t3, '(', push_operator3
195         add $s4, $s2, $s7
196         sb $t3, 0($s4)
197         addi $s7, $s7, 1
198         li $t2, ' '
199         add $s4, $s2, $s7
200         sb $t2, 0($s4)
201         addi $s7, $s7, 1
202         j after_pop2
203 push_operator3:
204         j next2
205 end_while1:beq $t7, -1, rt3
206         add $s4, $s3, $t7
207         lb $t3, 0($s4)
208         addi $t7, $t7, -1
209         add $s4, $s2, $s7
210         sb $t3, 0($s4)
211         addi $s7, $s7, 1
212         li $t2, ' '
213         add $s4, $s2, $s7
214         sb $t2, 0($s4)
215         addi $s7, $s7, 1
216         j end_while1
217 rt3:    li $t2, '\0'
218         add $s4, $s2, $s7
219         sb $t2, 0($s4)
220         lw $ra, 0($sp)
221         addi $sp, $sp, 4
222         jr $ra
223         nop
224
225 #CACULATE
226 # $s2 : bieu thuc hau to
227 # $s3 : ngan xep
228 CALCULATE:addi $sp, $sp, -4
229         sw $ra, 0($sp)
230         li $s6, 0
231         li $s7, -4
232         la $t2, arr

```



```

233 while2:add $s4, $s2, $s6
234         lb $t3, 0($s4)
235         beq $t3, 0, end_while2
236         jal check_number
237         li $t9, -1
238         li $s5, 0
239         beq $v0, 1, before_convert_num
240 next4:beq $t3, 0, end_while2
241         jal check_operator
242         beq $v0, 1, ccl
243 next5:addi $s6, $s6, 1
244         j while2
245 before_convert_num:add $s4, $t2, $s5
246         sb $t3, 0($s4)
247         addi $s5, $s5, 1
248         addi $t9, $t9, 1
249         addi $s6, $s6, 1
250         add $s4, $s2, $s6
251         lb $t3, 0($s4)
252         jal check_number
253         beq $v0, 1, before_convert_num
254         jal convert_num
255         j next4
256 ccl: addi $s3, $s3, -2
257         lb $a0, ($s3)
258         lb $a1, 1($s3)
259         jal CAL
260         j next5
261 end_while2:
262         lw $ra, ($sp)
263         addi $sp, $sp, 4
264         jr $ra
265 #Check xem co phai ki tu space khong?
266 # $v0 = 0 -> khong la space
267 # $v0 = 1 -> la space
268 check_space:
269         bne $t3, ' ', check_space_false
270 check_space_true:
271         li $v0, 1
272         jr $ra
273         nop
274 check_space_false:
275         li $v0, 0
276         jr $ra
277         nop
278 #Check xem co phai toan tu khong?
279 # $v0 = 0 -> khong phai la toan tu
280 # $v0 = 1 -> la toan tu, $s0 luu tru thu tu uu tien cua toan tu
281 check_operator:
282         addi $sp, $sp, -28
283         sw $s2, 24($sp)
284         sw $s3, 20($sp)
285         sw $s4, 16($sp)
286         sw $s5, 12($sp)
287         sw $s6, 8($sp)
288         sw $s7, 4($sp)
289         sw $t8, ($sp)
290         li $s2, '+'

```

```

291     li $s3, '-'
292     li $s4, '*'
293     li $s5, '/'
294     li $s6, '%'
295     li $s7, '('
296     li $t8, ')'
297     beq $t3, $s2, operator1
298     beq $t3, $s3, operator1
299     beq $t3, $s4, operator2
300     beq $t3, $s5, operator2
301     beq $t3, $s6, operator2
302     beq $t3, $s7, operator0
303     beq $t3, $t8, operator3
304     j check_operator_false
305 operator0: li $s0, 0
306           j check_operator_true
307 operator1: li $s0, 1
308           j check_operator_true
309 operator2: li $s0, 2
310           j check_operator_true
311 operator3: li $s0, 3
312 check_operator_true:
313     li $v0, 1
314     j rt1
315     nop
316 check_operator_false:
317     li $v0, 0
318 rt1:   lw     $s2, 24($sp)
319        lw     $s3, 20($sp)
320        lw     $s4, 16($sp)
321        lw     $s5, 12($sp)
322        lw     $s6, 8($sp)
323        lw     $s7, 4($sp)
324        lw     $t8, ($sp)
325        addi   $sp, $sp, 28
326        jr     $ra
327        nop
328 #Check xem co phai so khong?
329 # $v0 = 0 -> khong la so
330 # $v0 = 1 -> la so
331 check_number:
332     addi   $sp, $sp, -8
333     sw     $t8, 4($sp)
334     sw     $t9, ($sp)
335     li $t8, '0'
336     li $t9, '9'
337
338     beq $t8, $t3, check_number_true
339     beq $t9, $t3, check_number_true
340     bgt $t8, $t3, check_number_false
341     bgt $t3, $t9, check_number_false
342 check_number_true:
343     li $v0, 1
344     j     rt2
345 check_number_false:
346     li $v0, 0
347 rt2:
348     lw     $t9, ($sp)

```

```

349     lw     $t8, 4($sp)
350     addi   $sp, $sp, 8
351     jr     $ra
352 #Tinh gia tri bieu thuc
353 # $a0 : phan tu truoc
354 # $a1 : phan tu sau
355 # $t3 : op
356 # $a0 op $a1
357 CAL:
358 add_op: bne    $t3, '+', minus_op
359         add    $v0, $a0, $a1
360         j      rt_cal
361 minus_op: bne    $t3, '-', mul_op
362         sub    $v0, $a0, $a1
363         j      rt_cal
364 mul_op: bne    $t3, '*', div_op
365         mult   $a0, $a1
366         mflo    $v0
367         j      rt_cal
368 div_op: bne    $t3, '/', divr_op
369         div    $a0, $a1
370         mflo    $v0
371         j      rt_cal
372 divr_op: bne    $t3, '%',      rt_cal
373         div    $a0, $a1
374         mfhi    $v0
375 rt_cal:
376         sb     $v0, ($s3)
377         addi   $s3, $s3, 1
378     jr     $ra
379 #Chuyen doi ky tu thanh so
380 # $t2 : luu ki tu
381 # $s3 : lu so tuong ung
382 # $t9 : bien dem cua mang luu ki tu
383 convert_num:
384         addi   $sp, $sp, -8
385         sw     $t1, ($sp)
386         sw     $t5, 4($sp)
387         bne    $t9, $0, twoNum
388         lb     $t1, 0($t2)
389         addi   $t1, $t1, -48
390         j      push
391 twoNum:
392         lb     $t1, 0($t2)
393         addi   $t1, $t1, -48
394         addi   $t5, $0, 10
395         mult   $t1, $t5
396         mflo    $t1
397         lb     $t5, 1($t2)
398         addi   $t5, $t5, -48
399         add    $t1, $t1, $t5
400 push: sb $t1, 0($s3)
401         addi   $s3, $s3, 1
402         lw     $t5, 4($sp)
403         lw     $t1, ($sp)
404         addi   $sp, $sp, 8
405         j      rt4
406 end_loop2:
407 end_loop1:
408 rt4: jr $ra
409     nop

```

### C. Kết quả chạy mô phỏng

```
MENU
1.Chay chuong trinh
2.Thoat chuong trinh
1
Nhap vao bieu thuc trung to: 12%3+4*(5+6)
Bieu thuc trung to: 12%3+4*(5+6)
Bieu thuc hau to: 12 3 % 4 5 6 + * +
Ket qua bieu thuc vua nhap: 44
MENU
1.Chay chuong trinh
2.Thoat chuong trinh
1
Nhap vao bieu thuc trung to: 2.1*d+3.2
Bieu thuc khong hop le.
MENU
1.Chay chuong trinh
2.Thoat chuong trinh
2

-- program is finished running --
```

Clear