

Câu 1: Tính $(X+Y)*(Z-T)$

```
.ORIG X3000
; câu 1 , người dùng nhập lần lượt các số X,Y,Z,T
; Tính biểu thức sau đây  $(X+Y)*(Z-T)$ 
LD R3, TO_NUM ; chuẩn bị cơ số để chuyển ascii về lại number
LEA R0, NHAP_X
PUTS
IN
ADD R1,R0,R3

LEA R0, NHAP_Y
PUTS
IN
ADD R0,R0,R3
ADD R1,R0,R1 ; tới thời điểm hiện tại ::: R1 = X+Y

LEA R0, NHAP_Z
PUTS
IN
ADD R2,R0,R3

LEA R0, NHAP_T
PUTS
IN
ADD R0,R0,R3

NOT R0,R0 ; LẤY BÙ 2 ĐỂ XUẤT HIỆN -T
ADD R0,R0,#1
ADD R2,R0,R2 ; tới thời điểm hiện tại ::: R2 = Z-T
BRZP PHEP_NHAN ; NẾU R2 = 0 HAY >0 , CUỘC ĐỜI ĐẸP CHẤN
AND R5,R5,#0
ADD R5,R5,#1 ; NHƯNG NẾU R2 < 0 , ĐẶT R5 = 1 NHƯ LÀ BIẾN GÁN CHO SỐ ÂM
NOT R2,R2 ; và lấy bù 2 của R2 để thực hiện phép nhân ( module nhân chỉ
thực hiện được phép nhân 2 số dương)
ADD R2,R2,#1

PHEP_NHAN AND R3,R3,#0 ; xoá thanh ghi R3 để lưu kết quả phép nhân
;thực hiện phép nhân hai thanh ghi R1 và R2 , kết quả được lưu vào R3
AGAIN ADD R3, R3, R1
ADD R2, R2, #-1
BRP AGAIN

; chuyển kết quả về ô R1 =  $(X+Y)*(Z-T)$ 
ADD R1, R3, #0
ADD R5,R5,#0 ; dò xem thanh ghi R5 có bằng 1 hay không
BRNZ IN_MAN_HINH ; nếu R5 không bằng 1 ( tức R2 không âm hay nói cách khác,
kết quả dương thì đến bước in màn hình kết quả)
NOT R1,R1 ; ngược lại phải lấy bù 2 kết quả phép nhân thì R2 là một số hạng
< 0
ADD R1,R1,#1
;bắt đầu in ra màn hình
IN_MAN_HINH
; KIỂM TRA SỐ ÂM
ADD R1,R1,#0
BRZP WITHOUT_NEG ; nếu R1 = 0 hoặc >0 thì bỏ qua bước kiểm tra giá trị âm
LD R0,NEGATIVE
OUT ; in dấu '-' nếu là số âm
NOT R1, R1
ADD R1,R1,#1 ; bù 2 nếu là số âm
```

```

WITHOUT_NEG
AND R2,R2,#0 ; XOÁ R2 LÀM BỘ NHỚ ĐỆM
LD R3, TO_ASCII ; CHUẨN BỊ CƠ SỐ CHUYỂN ASCII ĐỂ IN RA MÀN HÌNH

LD R5 , HUNDREDS ; CHUẨN BỊ SỐ ĐỂ KIỂM HÀNG TRĂM
NOT R5,R5
ADD R5,R5,#1

COUNT_100 ADD R1,R1,R5 ; KIỂM HÀNG TRĂM
BRN STOP_COUNT_100
ADD R2,R2,#1
BRNZP COUNT_100

STOP_COUNT_100
LD R5, HUNDREDS
ADD R1,R1,R5 ; BÙ LẠI 100 DO BỊ TRỪ HỤT

ADD R2,R2,#0
BRZ COUNT_10 ; NẾU HÀNG TRĂM BẰNG 0 THÌ TIẾP TỚI ĐẾM HÀNG CHỤC, KHÔNG OUT RA MÀN HÌNH
ADD R0,R2,R3
OUT

AND R2,R2,#0 ; TẠO MỚI LẠI BIẾN ĐẾM
COUNT_10 ADD R1,R1,#-10 ; kiểm hàng chục
BRN STOP_COUNT_10
ADD R2,R2,#1
BRNZP COUNT_10

STOP_COUNT_10
ADD R1,R1,#10 ; bù lại 10 do trừ hụt

ADD R2,R2,#0 ; nếu hàng chục bằng 0 thì in hàng đơn vị luôn
BRZ IN_DON_VI

ADD R0,R2,R3
OUT

IN_DON_VI ADD R0,R1,R3
OUT
HALT

NHAP_X .STRINGZ "Nhap X: "
NHAP_Y .STRINGZ "Nhap Y: "
NHAP_Z .STRINGZ "Nhap Z: "
NHAP_T .STRINGZ "Nhap T: "
TO_ASCII .FILL x30
TO_NUM .FILL x-30
NEGATIVE .FILL #45
HUNDREDS .FILL #100
.END

```

Link source github gist:

<https://gist.github.com/nguyenkhoa2001/0480224ced46dafce847749736b021ae>

Câu 2: $(X+Y)*Z-T$

```
.ORIG X3000
; câu 1 , người dùng nhập lần lượt các số X,Y,Z,T
; Tính biểu thức sau đây  $(X+Y)*Z-T$ 
LD R3, TO_NUM ; chuẩn bị cơ số để chuyển ascii về lại number
LEA R0, NHAP_X
PUTS
IN
ADD R1,R0,R3

LEA R0, NHAP_Y
PUTS
IN
ADD R0,R0,R3
ADD R1,R0,R1 ; tới thời điểm hiện tại :::  $R1 = X+Y$ 

LEA R0, NHAP_Z
PUTS
IN
ADD R2,R0,R3 ; thời điểm hiện tại :::  $R2 = Z$  ; ta thực hiện luôn phép nhân
 $(X+Y) * Z = R1 * R2$  ; lưu kết quả vào R4

AND R4,R3,#0 ; XOÁ THANH GHI R4 ĐỂ CHỨA KẾT QUẢ PHÉP NHÂN
AGAIN ADD R4, R4, R1
      ADD R2, R2, #-1
BRP AGAIN

LEA R0, NHAP_T
PUTS
IN
ADD R1,R0,R3
NOT R1,R1 ;lấy bù 2 của T để có '-T'
ADD R1,R1,#1
ADD R1,R1,R4 ;  $R1 = (X+Y)*Z-T$ 

;bắt đầu in ra màn hình
; KIỂM TRA SỐ ÂM
ADD R1,R1,#0
BRZP WITHOUT_NEG ; nếu  $R1 = 0$  hoặc  $>0$  thì bỏ qua bước kiểm tra giá trị âm
LD R0,NEGATIVE
OUT      ; in dấu '-' nếu là số âm
NOT R1, R1
ADD R1,R1,#1 ; bù 2 nếu là số âm

WITHOUT_NEG
AND R2,R2,#0 ; XOÁ R2 LÀM BỘ NHỚ ĐỆM
LD R3, TO_ASCII ; CHUẨN BỊ CƠ SỐ CHUYỂN ASCII ĐỂ IN RA MÀN HÌNH

LD R5 , HUNDREDS ; CHUẨN BỊ SỐ ĐỂ KIỂM HÀNG TRĂM
NOT R5,R5
ADD R5,R5,#1

COUNT_100 ADD R1,R1,R5 ; KIỂM HÀNG TRĂM
BRN STOP_COUNT_100
ADD R2,R2,#1
BRNZP COUNT_100

STOP_COUNT_100
LD R5, HUNDREDS
```

```

ADD R1,R1,R5 ; BÙ LẠI 100 DO BỊ TRỪ HỤT

ADD R2,R2,#0
BRZ COUNT_10 ; NẾU HÀNG TRĂM BẰNG 0 THÌ TIẾP TỚI ĐẾM HÀNG CHỤC, KHÔNG OUT
RA MÀN HÌNH
ADD R0,R2,R3
OUT

AND R2,R2,#0 ; TẠO MỚI LẠI BIẾN ĐẾM
COUNT_10 ADD R1,R1,#-10 ; kiểm hàng chục
BRN STOP_COUNT_10
ADD R2,R2,#1
BRNZP COUNT_10

STOP_COUNT_10
ADD R1,R1,#10 ; bù lại 10 do trừ hụt

ADD R2,R2,#0 ; nếu hàng chục bằng 0 thì in hàng đơn vị luôn
BRZ IN_DON_VI

ADD R0,R2,R3
OUT

IN_DON_VI ADD R0,R1,R3
OUT
HALT

NHAP_X .STRINGZ "Nhap X: "
NHAP_Y .STRINGZ "Nhap Y: "
NHAP_Z .STRINGZ "Nhap Z: "
NHAP_T .STRINGZ "Nhap T: "
TO_ASCII .FILL x30
TO_NUM .FILL x-30
NEGATIVE .FILL #45
HUNDREDS .FILL #100
.END

```

Link source github gist:

<https://gist.github.com/nguyenkhoa2001/2f20ed590a7a1503246243841a29560d>

Câu 3: $X*Y - Z + T$

```
.ORIG X3000
; câu 1 , người dùng nhập lần lượt các số X,Y,Z,T
; Tính biểu thức sau đây  $X*Y-Z+T$ 
LD R3, TO_NUM ; chuẩn bị cơ số để chuyển ascii về lại number
LEA R0, NHAP_X
PUTS
IN
ADD R1,R0,R3

LEA R0, NHAP_Y
PUTS
IN
ADD R0,R0,R3 ; thời điểm hiện tại thanh ghi R1 chứa X, thanh ghi R0 chứa Y,
thực hiện phép nhân  $R1*R0$  , lưu kết quả vào thanh ghi 4

AND R4,R3,#0 ; XOÁ THANH GHI R4 ĐỂ CHỨA KẾT QUẢ PHÉP NHÂN
AGAIN    ADD R4, R4, R1
        ADD R0, R0, #-1
BRP AGAIN

LEA R0, NHAP_Z
PUTS
IN
ADD R2,R0,R3
NOT R2,R2
ADD R2,R2,#1 ; bù 2 của Z thu được -Z
ADD R1 , R2, R4 ;  $R1 = X*Y - Z$ 

LEA R0, NHAP_T
PUTS
IN
ADD R2,R0,R3
ADD R1,R1,R2 ;  $R1 = X*Y - Z + T$ 

;bắt đầu in ra màn hình
; KIỂM TRA SỐ ÂM
ADD R1,R1,#0
BRZP WITHOUT_NEG ; nếu  $R1 = 0$  hoặc  $>0$  thì bỏ qua bước kiểm tra giá trị âm
LD R0,NEGATIVE
OUT      ; in dấu '-' nếu là số âm
NOT R1, R1
ADD R1,R1,#1 ; bù 2 nếu là số âm

WITHOUT_NEG
AND R2,R2,#0 ; XOÁ R2 LÀM BỘ NHỚ ĐỆM
LD R3, TO_ASCII ; CHUẨN BỊ CƠ SỐ CHUYỂN ASCII ĐỂ IN RA MÀN HÌNH

LD R5 , HUNDREDS ; CHUẨN BỊ SỐ ĐỂ KIỂM HÀNG TRĂM
NOT R5,R5
ADD R5,R5,#1

COUNT_100 ADD R1,R1,R5 ; KIỂM HÀNG TRĂM
BRN STOP_COUNT_100
ADD R2,R2,#1
BRNZP COUNT_100
```

```

STOP_COUNT_100
LD R5, HUNDREDS
ADD R1,R1,R5 ; BÙ LẠI 100 DO BỊ TRỪ HỤT

ADD R2,R2,#0
BRZ COUNT_10 ; NẾU HÀNG TRĂM BẰNG 0 THÌ TIẾP TỚI ĐẾM HÀNG CHỤC, KHÔNG OUT
RA MÀN HÌNH
ADD R0,R2,R3
OUT

AND R2,R2,#0 ; TẠO MỚI LẠI BIẾN ĐẾM
COUNT_10 ADD R1,R1,#-10 ; kiểm hàng chục
BRN STOP_COUNT_10
ADD R2,R2,#1
BRNZP COUNT_10

STOP_COUNT_10
ADD R1,R1,#10 ; bù lại 10 do trừ hụt

ADD R2,R2,#0 ; nếu hàng chục bằng 0 thì in hàng đơn vị luôn
BRZ IN_DON_VI

ADD R0,R2,R3
OUT

IN_DON_VI ADD R0,R1,R3
OUT
HALT

NHAP_X .STRINGZ "Nhap X: "
NHAP_Y .STRINGZ "Nhap Y: "
NHAP_Z .STRINGZ "Nhap Z: "
NHAP_T .STRINGZ "Nhap T: "
TO_ASCII .FILL x30
TO_NUM .FILL x-30
NEGATIVE .FILL #45
HUNDREDS .FILL #100
.END

```

Link source github gist:

<https://gist.github.com/nguyenkhoa2001/f0281ca8622eaaf5a2d38f83ebdcbb0c>