

NAME: YAHIA AYARI

MAIL: [s192452@studenti.polito.it](mailto:s192452@studenti.polito.it)

Test solution

; Computer Architectures

; test item1

shift EQU 8

.MODEL small

.STACK

.DATA

INPUT\_BIN DB ?

OUTPUT\_BCD DB 3 DUP (?)

.CODE

xor dx,dx

xor cx,cx

mov AH, 1

int 21h

mov si,shift

MOV DI,0xFF

lab1: dec si

shl dl,1 ;shift left the input 8 bit

mov ax,cx ;save last res into ax

mov cx,00h

adc cx,00h ;add to cx the flag obtained by shifting dl(input)

shl ax,1 ;shift left last res

add cx,ax ;make into cx the new number shifted

mov bl,cl ;make in bl the unit and the ten

mov dh,ch ;make in dh the hundreds

shl bl,4

shr bl,4 ;obtain the unit in bl

cmp si,0 ;to stop the last loop without add\_3 in the last time

je write\_output

cmp bl,5

JGE add\_3\_unit

ten: mov bh,cl

shr bh,4 ;obtain the ten in bh

cmp bh,5

JGE add\_3\_ten

hund: cmp dh,5

JGE add\_3\_hund

jl also

add\_3\_unit: add bl,3h ;add3 to unit

jmp ten

add\_3\_ten: add bh,3h ;add3 to ten

jmp hund

add\_3\_hund: add dh,3h

jmp also

```
also:    mov cl,bh ; put tens in cl

        shl cl,4 ; lasciare place to unit in cl

        add cl,bl ; add unit to cl

        mov ch,dh ; put hundreds in ch

        jmp lab1
```

```
write_output: mov dh,ch ;make hundreds res into dh

            mov bl,cl

            shl bl,4

            sar bl,4 ;make unit res into bl

            mov bh,cl

            sar bh,4 ;make tens res into bh


            mov OUTPUT_BCD,bl

            mov OUTPUT_BCD+1,bh

            mov OUTPUT_BCD+2,dh


            mov ah,2

            mov dl,OUTPUT_BCD[2]

            add dl,30h ;to write a number of the ascii code

            int 21h

            mov dl,OUTPUT_BCD[1]

            add dl,30h

            int 21h

            mov dl,OUTPUT_BCD[0]

            add dl,30h

            int 21h
```

Fin:

mov ah,4ch

int 21H

END

; Computer Architectures

; test item2

shift EQU 8

.MODEL small

.STACK

.DATA

INPUT\_BIN DB ?

OUTPUT\_BCD DB 2 DUP (?)

.CODE

xor dx,dx

xor cx,cx

mov si,shift

MOV DI,0xFF

lab1: dec si

shl dl,1 ;shift left the input 8 bit

mov ax,cx ;save last res into ax

mov cx,00h

adc cx,00h ;add to cx the flag obtained by shifting dl(input)

shl ax,1 ;shift left last res

add cx,ax ;make into cx the new number shifted

mov bl,cl ;make in bl the unit and the ten

mov dh,ch ;make in dh the hundreds

shl bl,4

shr bl,4 ;obtain the unit in bl

cmp si,0 ;to stop the last loop without add\_3 in the last time

je write\_output

cmp bl,5

JGE add\_3\_unit

ten: mov bh,cl

shr bh,4 ;obtain the ten in bh

cmp bh,5

JGE add\_3\_ten

hund: cmp dh,5

JGE add\_3\_hund

jl also

add\_3\_unit: add bl,3h ;add3 to unit

jmp ten

add\_3\_ten: add bh,3h ;add3 to ten

jmp hund

add\_3\_hund: add dh,3h

jmp also

also: mov cl,bh ; put tens in cl

shl cl,4 ; lasciare place to unit in cl

add cl,bl ; add unit to cl

mov ch,dh ; put hundreds in ch

jmp lab1

write\_output: mov dh,ch ;make hundreds res into dh

mov bl,cl ;make unit an tens res into bl

mov OUTPUT\_BCD,bl

mov OUTPUT\_BCD+1,dh

mov ah,2

mov dl,OUTPUT\_BCD[1]

add dl,30h ;to write a number of the ascii code

int 21h

mov dl,OUTPUT\_BCD[0]

shr dl,4 ;affich ten

add dl,30h

int 21h

mov dl,OUTPUT\_BCD[0]

sal dl,4

shr dl,4 ;affich unit

add dl,30h

int 21h



Fin:

mov ah,4ch

int 21H

END

; Computer Architectures

; test item3

shift EQU 16

.MODEL small

.STACK

.DATA

INPUT\_BIN DW ?

OUTPUT\_BCD DB 5 DUP (?)

.CODE

xor dx,dx

xor cx,cx

mov si,shift

MOV Di,0xFFFF

lab1: dec si

shl di,1 ;shift left the input 16 bit

mov ax,cx ;save last res into ax (unit,ten,hund,mille)

mov bh,dl ;save last res into bh (ten\_mille)

mov cx,00h

mov dl,00h

adc cx,00h ;add flag of the shifted input to cx

shl ax,1 ;shift left the res

adc dl,00h ;add flag of the shifted cx in di

add cx,ax ;obtain (unit,ten,hund,mille) into ax

shl bh,1 ;shift ten mille

add dl,bh ;obtain ten\_mille into di

mov bl,cl ;make in bl the unit and the ten

shl bl,4

shr bl,4 ;obtain the unit in bl

cmp si,0 ;to stop the last loop without add\_3 in the last time

je write\_output

cmp bl,5

JGE add\_3\_unit

ten: mov bh,cl

shr bh,4 ;obtain the ten in bh

cmp bh,5

JGE add\_3\_ten

hund: mov al,ch ;put hund and mille into al

shl al,4

shr al,4 ;obtain the hund in al

cmp al,5

JGE add\_3\_hund

mille: mov ah,ch

shr ah,4 ;obtain the mille in ah

cmp ah,5

JGE add\_3\_mille

ten\_mille: cmp dl,5

JGE add\_3\_ten\_mille

jl also

add\_3\_unit: add bl,3h ;add3 to unit

jmp ten

add\_3\_ten: add bh,3h ;add3 to ten

jmp hund

add\_3\_hund: add al,3h

jmp mille

add\_3\_mille: add ah,3h

jmp ten\_mille

add\_3\_ten\_mille: add dl,3h

jmp also

also: mov cl,bh ; put tens in cl

shl cl,4 ; lasciare place to unit in cl

add cl,bl ; add unit to cl

mov ch,ah ; put mille in ch

shl ch,4 ; lasciare place to hund in ch

add ch,al ; add hundreds to ch

jmp lab1

write\_output:

mov bl,cl

shl bl,4

sar bl,4 ;make unit res into bl

mov bh,cl

sar bh,4 ;make tens res into bh

mov al,ch

shl al,4

sar al,4 ;make hund res into al

mov ah,ch

sar ah,4 ;make mille res into ah

mov OUTPUT\_BCD,bl

mov OUTPUT\_BCD+1,bh

mov OUTPUT\_BCD+2,al

mov OUTPUT\_BCD+3,ah

mov OUTPUT\_BCD+4,dl

mov ah,2

mov dl,OUTPUT\_BCD[4]

add dl,30h ;to write a number of the ascii code

int 21h

mov dl,OUTPUT\_BCD[3]

add dl,30h ;to write a number of the ascii code

int 21h

mov dl,OUTPUT\_BCD[2]

```
add dl,30h          ;to write a number of the ascii code
```

```
int 21h
```

```
mov dl,OUTPUT_BCD[1]
```

```
add dl,30h
```

```
int 21h
```

```
mov dl,OUTPUT_BCD[0]
```

```
add dl,30h
```

```
int 21h
```

Fin:

```
mov ah,4ch
```

```
int 21H
```

```
END
```

; Computer Architectures

; test item4

shift EQU 16

.MODEL small

.STACK

.DATA

INPUT\_BIN Dw ?

OUTPUT\_BCD DB 3 DUP (?)

.CODE

xor dx,dx

xor cx,cx

mov si,shift

MOV Di,0xFFFE

lab1: dec si

shl di,1 ;shift left the input 16 bit

mov ax,cx ;save last res into ax (unit,ten,hund,mille)

mov bh,dl ;save last res into bh (ten\_mille)

mov cx,00h

mov dl,00h

adc cx,00h ;add flag of the shifted input to cx

shl ax,1 ;shift left the res

adc dl,00h ;add flag of the shifted cx in di

add cx,ax ;obtain (unit,ten,hund,mille) into ax

shl bh,1 ;shift ten mille

add dl,bh ;obtain ten\_mille into di

mov bl,cl ;make in bl the unit and the ten

shl bl,4

shr bl,4 ;obtain the unit in bl

cmp si,0 ;to stop the last loop without add\_3 in the last time

je write\_output

cmp bl,5

JGE add\_3\_unit

ten: mov bh,cl

shr bh,4 ;obtain the ten in bh

cmp bh,5

JGE add\_3\_ten

hund: mov al,ch ;put hund and mille into al

shl al,4

shr al,4 ;obtain the hund in al

cmp al,5

JGE add\_3\_hund

mille: mov ah,ch

shr ah,4 ;obtain the mille in ah

cmp ah,5

JGE add\_3\_mille



ten\_mille: cmp dl,5

JGE add\_3\_ten\_mille

jl also

add\_3\_unit: add bl,3h ;add3 to unit

jmp ten

add\_3\_ten: add bh,3h ;add3 to ten

jmp hund

add\_3\_hund: add al,3h

jmp mille

add\_3\_mille: add ah,3h

jmp ten\_mille

add\_3\_ten\_mille: add dl,3h

jmp also

also: mov cl,bh ; put tens in cl

shl cl,4 ; lasciare place to unit in cl

add cl,bl ; add unit to cl

mov ch,ah ; put mille in ch

shl ch,4 ; lasciare place to hund in ch

add ch,al ; add hundreds to ch

jmp lab1

write\_output:

mov bl,cl ;unit and ten

mov al,ch ;hundreads and mille

mov OUTPUT\_BCD,bl

mov OUTPUT\_BCD+1,al

mov OUTPUT\_BCD+2,dl

mov ah,2

mov dl,OUTPUT\_BCD[2] ;aff ten\_mille

add dl,30h

int 21h

mov dl,OUTPUT\_BCD[1] ;aff mille

shr dl,4

add dl,30h

int 21h

mov dl,OUTPUT\_BCD[1] ;aff hundreads

shl dl,4

shr dl,4

add dl,30h

int 21h

mov dl,OUTPUT\_BCD[0] ;aff ten

shr dl,4

```
add dl,30h
```

```
int 21h
```

```
mov dl,OUTPUT_BCD[0] ;aff unit
```

```
shl dl,4
```

```
shr dl,4
```

```
add dl,30h
```

```
int 21h
```

Fin:

```
mov ah,4ch
```

```
int 21H
```

```
END
```