

INDEX

SNO	PRACTICALS	FILE NAME	P.NO
1	Write a program for 32-bit binary (i) Addition (ii) Subtraction (iii) division and (iv) Multiplication	bcdsub.ams subb.asm mult.asm div.asm	3-15
2	Write a program for 32-bit BCD (i) Addition and (ii) Subtraction	bcd.a.asm sbcd.asm	16-22
3	Write a program for Sorting	sort.asm	23-27
4	Write a program for (i) linear search and (ii) Binary search	lins.asm bins.asm	28-36
5	Write a program to (i) Add two arrays (ii) Subtract two array	as2a.asm	37-39
6	Write a program for binary to ascii conversion	btoa.asm	40-42
7	Write a program for ascii to binary conversion	atob.asm	43-47

Q1. Write a program for 32-bit binary

(i) Addition (ii) Subtraction (iii) division and (iv) Multiplication

[CODE](#)

;32 bit binary addition

.model small

.386

.data

num1 DD 00000000H

num2 DD 00000000H

num3 DD 00000000H

msg db 10,13,"Enter the first no.:: \$"

msg1 db 10,13,"Enter the second no.:: \$"

msg2 db 10,13,"The Resultant sum is :: \$"

.code

.startup

MOV AH,09

MOV DX,OFFSET msg

INT 21H

MOV EBX,0

MOV CX,8

AGAIN: MOV AH,01 ;1ST NO. ENTERED

INT 21H

```

CMP AL, 'A'

JGE L2

SUB AL, 30H

SHL EBX, 4

ADD BL, AL

LOOP AGAIN

MOV num1, EBX

MOV AH, 09

MOV DX, OFFSET msg1

INT 21H


MOV EBX, 0

MOV CX, 8

AGAIN1:MOV AH, 01 ;2nd NO. ENTERED

INT 21H

CMP AL, 'A'

JGE L2

SUB AL, 30H

SHL EBX, 4

ADD BL, AL

LOOP AGAIN1

MOV num2, EBX

```

```
mov ax, word ptr num1
mov dx, word ptr num2
add al, dl
daa
mov bl, al
mov al, ah
adc al, dh
daa
mov bh, al
mov word ptr num3, bx
mov ax, word ptr num1+2
mov dx, word ptr num2+2
adc al, dl
daa
mov bl, al
mov al, ah
adc al, dh
daa
mov bh, al
mov word ptr num3+2, bx
mov ebx, num3
mov ah, 09h
```

```
mov dx, offset msg2
```

```
int 21h
```

```
jnc l6
```

```
mov ah, 02h
```

```
mov dl, "1"
```

```
int 21h
```

```
l6: MOV CX, 8
```

```
AGAIN2: ROL EBX, 4
```

```
MOV DL, BL
```

```
AND DL, 0FH
```

```
ADD DL, 30H
```

```
MOV AH, 02
```

```
INT 21H
```

```
LOOP AGAIN2
```

```
L2: .EXIT
```

```
END
```

output

```
Z:\>c:
C:\>cd tasm
C:\TASM>tasm bcsub.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:   bcsub.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  474k

C:\TASM>tlink bcsub.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>bcsub.exe
Enter the first no.:: 11223344
Enter the second no.:: 11112222
The Resultant sum is :: 22335566
C:\TASM>
```

Code:

;32 bit binary subtraction

```
.model small

.386

.stack 100h
.data
num1 DD 00000000H
num2 DD 00000000H
num3 DD 00000000H
msg db 10,13,"Enter the first no.:: $"
msg1 db 10,13,"Enter the second no.:: $"
msg2 db 10,13,"The Resultant Difference is :: $"

.code
.startup

MOV AH,09
MOV DX,OFFSET msg
INT 21H

MOV EBX,0
MOV CX,8

AGAIN: MOV AH,01 ;1ST NO. ENTERED
INT 21H
CMP AL,'A'
JGE L2
SUB AL,30H
SHL EBX,4
ADD BL,AL
LOOP AGAIN

MOV num1,EBX

MOV AH,09
```



```
MOV DX, OFFSET msg1
INT 21H
```

```
MOV EBX, 0
MOV CX, 8
AGAIN1:MOV AH, 01 ;2nd NO. ENTERED
INT 21H
CMP AL, 'A'
JGE L2
SUB AL, 30H
SHL EBX, 4
ADD BL, AL
LOOP AGAIN1
```

```
MOV num2, EBX
```

```
mov ah, 09h
mov dx, offset msg2
int 21h
```

```
mov ax, word ptr num1+2
mov dx, word ptr num2+2
sub al, dl
das
mov dl, al
sbb ah, dh
das
mov dh, al
call dispb
```

```
mov ax, word ptr num1
mov dx, word ptr num2
sub al, dl
das
mov dl, al
sbb ah, dh
das
mov dh, al
```

```

disph proc near
mov cl,4
mov ch,4
disph1:
rol ax,cl
push ax
and al,0fh
add al,30h
cmp al,'9'
jbe disph2
add al,7
disph2:
mov ah,2
mov dl,al
int 21h
pop ax
dec ch
jne disph1
ret
disph endp
.EXIT
L2:
.EXIT
END

```

output

```
Z:\>mount c: c:/
Drive C is mounted as local directory c:/^

Z:\>c:

C:\>cd tasm

C:\TASM>tasm subb.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:  subb.asm
Error messages:  None
Warning messages: None
Passes:         1
Remaining memory: 474k

C:\TASM>tlink subb.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International

C:\TASM>subb.exe

Enter the first no.:: 11223344
Enter the second no.:: 11111111
The Resultant Difference is :: 00112233_
```

Code:

;32bit binary multiplication

.model tiny

.code

mov ax, 04H

mov bx, 02H

mul bl

DISP PROC

PUSH CX

MOV CL, 4

MOV CH, 4

D1:

ROL AX, CL

PUSH AX

AND AL, 0FH

ADD AL, 30H

CMP AL, '9'

JBE D2

ADD AL, 7H

D2:

MOV AH, 02H

MOV DL, AL

INT 21H

POP AX

DEC CH

JNZ D1

POP CX

RET

DISP ENDP

.exit

End

Output

```
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c: c:/
Drive C is mounted as local directory c:/\

Z:\>c:

C:\>cd tasm

C:\TASM>tasm mult.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:    mult.asm
Error messages:     None
Warning messages:   None
Passes:             1
Remaining memory:   476k

C:\TASM>tlink mult.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>mult.exe
0000_
```

CODE:

;32bit binary division

.model tiny

.code

mov ax, 02H

mov bx, 04H

div bl

DISP PROC

PUSH CX

MOV CL, 4

MOV CH, 4

D1:

ROL AX, CL

PUSH AX

AND AL, 0FH

ADD AL, 30H

CMP AL, '9'

JBE D2

ADD AL, 7H

D2:

MOV AH, 02H

MOV DL, AL

INT 21H

POP AX

DEC CH

JNZ D1

POP CX

RET

DISP ENDP

```
.exit  
End
```

output

```
Z:\>SET BLASTER=A220 I7 D1 H5 T6  
  
Z:\>mount C: C:/  
Drive C is mounted as local directory C:/\n  
Z:\>C:  
  
C:\>cd tasm  
  
C:\TASM>tasm div.asm  
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International  
  
Assembling file:  div.asm  
Error messages:  None  
Warning messages: None  
Passes:         1  
Remaining memory: 476k  
  
C:\TASM>tlink div.obj  
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International  
Warning: no stack  
  
C:\TASM>div.exe  
0200_
```

Q2. Write a program for 32-bit BCD
(i) Addition and (ii) Subtraction

Code:

```
;32bit bcd subtraction
```

```
.model small
```

```
.data
```

```
num1 db 10,"number1 : $"
```

```
num2 db 10,"number2 : $"
```

```
diff db 10,"Difference : $"
```

```
.code
```

```
.startup
```

```
mov dx , offset num1
```

```
mov ah, 9h
```

```
int 21h
```

```
mov ax, 3333h
```

```
call disph
```

```
mov ax, 3333h
```

```
call disph
```

```
mov dx, offset num2
```

```
mov ah, 9h
```

```
int 21h
```

```
mov ax, 1111h
```



```
call disph  
mov ax, 1111h  
call disph  
call sub_num  
.exit
```

```
sub_num proc near  
mov dx, 1111h  
mov bx, 3333h  
mov cx, 1111h  
mov ax, 3333h  
sub al, cl  
das  
mov cl, al  
sbb ah, ch  
mov al, ah  
das  
mov ch, al  
mov si, cx  
mov al, bl  
sbb al, dl
```

```
das
mov bl, al
mov al, bh
sbb al, dh
das
mov bh, al
mov dx, offset diff
mov ah, 9h
int 21h
mov ax, si
call disph
mov ax, bx
call disph

ret

sub_num endp
```

```
disph proc near
mov cl, 4
mov ch, 4
disph1:
```

```
rol ax,cl  
push ax  
and al,0fh  
add al,30h  
cmp al,'9'  
jbe disph2  
add al,7  
disph2:  
mov ah,2  
mov dl,al  
int 21h  
pop ax  
dec ch  
jne disph1  
ret  
disph endp  
End
```

output

```
Difference : 19700324
C:\TASM>tasm sbcd.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:  sbcd.asm
Error messages:   None
Warning messages: None
Passes:          1
Remaining memory: 475k

C:\TASM>tlink sbcd.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
sbcd.obj : unable to open file

C:\TASM>tlink sbcd.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>sbcd.exe

number1 : 33333333
number2 : 11111111
Difference : 22222222
C:\TASM>
```

Code:

;32bit bcd additon

```
.MODEL SMALL ;assembler memory model
.STACK 100H
.DATA
MSG1 DB "4 DIGIT BCD SUM IS = $"
BCDSUML DB ?
BCDSUMH DB ?
.CODE
MOV AX,@DATA
MOV DS,AX
XOR AX, AX ; clear register AX
MOV AL, 34H
MOV BL, 98H
ADD AL,BL
DAA; DECIMAL ADJUST AFTER ADDITION
MOV BCDSUML, AL
MOV AL, 12H
MOV BL, 23H
ADC AL,BL
DAA; OPERATES ONLY ON AL
MOV BCDSUMH, AL
MOV AH, BCDSUMH
MOV AL, BCDSUML
PUSH AX
Page 19 of 43MOV DX, OFFSET MSG1
MOV AH, 09H
INT 21H ; INT STANDS FOR INTERRUPT INSTRUCTION 21H IS
INTERRUPT NO FOR DOS SERVICES
POP AX
CALL DISPLAY
MOV AH, 4CH
INT 21H
DISPLAY PROC NEAR ; PROC IS KEYWORD FOR PROCEDURE
MOV CH, 04H
MOV CL, 04H
DISP1:
ROL AX, CL ; ROTATE LEFT 4 TIMES
```

```

PUSH AX ; SAVING ON STACK
AND AL, 0FH
ADD AL, 30H ; 48 IN DECIMAL
CMP AL, '9' ; COMPARE WITH ASCII VALUE OF 9
JBE DISP2
ADD AL, 7
DISP2: MOV DL, AL
MOV AH, 02H
INT 21H
POP AX
DEC CH
JNZ DISP1
RET
DISPLAY ENDP
END

```

```

Z:\>C:
C:\>cd tasm
C:\TASM>\tasm bcda.asm
Illegal command: \tasm.
C:\TASM>tasm bcda.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:  bcda.asm
*Warning* bcda.asm(47) Reserved word used as symbol: DISPLAY
Error messages:   None
Warning messages: 1
Passes:           1
Remaining memory: 475k

C:\TASM>tlink bcda.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International

C:\TASM>bcda.exe
4 DIGIT BCD SUM IS = 3632
C:\TASM>

```

Q.3 Write a program for Sorting?

Code:

```
;sorting
```

```
.model small
```

```
.386
```

```
.data
```

```
ARRAY DW 20 DUP (?)
```

```
DATA1 dw 0000H
```

```
NUMB DW 0000H
```

```
msg db 10,13,"Enter the size of the array :: $"
```

```
msg2 db 10,13,"Enter the array :: $"
```

```
msg3 db 10,13,"The sorted array is :: $"
```

```

.code

.startup

MOV AH, 09

MOV DX, OFFSET msg

INT 21H

MOV AH, 01

INT 21H

SUB AL, 30H

MOV AH, 0

MOV CX, AX

MOV DATA1, AX


MOV AH, 09

MOV DX, OFFSET msg2

INT 21H

MOV AH, 0

MOV SI, 0

MOV BX, OFFSET ARRAY

L1: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

MOV DX, SI ; input element of the array

```



```

MOV AH, 01H

INT 21H

SUB AL, 30H

MOV SI, DX

MOV [BX + SI], AX

INC SI

LOOP L1


MOV CX, DATA1

MOV BX, OFFSET ARRAY

MOV DI, CX

L2: MOV CX, DATA1

MOV NUMB, CX      ; Change1

DEC NUMB          ; Change2

MOV CX, NUMB      ; change3

MOV SI, 0

L3: MOV AL, [BX + SI]

CMP AL, [BX + SI + 1]

JL L4

XCHG AL, [BX + SI + 1]

MOV [BX + SI], AL

L4: INC SI

```

```
LOOP L3

DEC DI

JNZ L2


MOV CX, DATA1

MOV SI, 0

MOV BX, OFFSET ARRAY

MOV AH, 09

MOV DX, OFFSET msg3

INT 21H

L5: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

MOV DX, [BX + SI]

INC SI

ADD DL, 30H

MOV AH, 02

INT 21H

LOOP L5

.EXIT

END
```

output

```
Assembling file:  sort.asm
Error messages:  None
Warning messages: None
Passes:         1
Remaining memory: 475k

C:\TASM>tlink sort.obj
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>sort.exe

Enter the size of the array :: 4
Enter the array ::
8
3
5
7
The sorted array is ::
3
5
7
8
C:\TASM>_
```

Q4. Write a program for

(i) linear search and (ii) Binary search

;Linear Search

.model small

.386

.data

ARRAY DW 20 DUP (?)

DATA1 dw 0000H

success db 10,13,"Element is present in the array \$"

fail db 10,13,"Element is not present in the array \$"

msg db 10,13,"Enter the size of the array :: \$"

msg2 db 10,13,"Enter the array :: \$"

msg3 db 10,13,"Enter the element to be searched :: \$"

.code

.startup

MOV AH, 09

MOV DX, OFFSET msg

INT 21H

MOV AH, 01

INT 21H

SUB AL, 30H

MOV AH, 0

MOV CX, AX

MOV DATA1, AX

MOV AH, 09

MOV DX, OFFSET msg2

INT 21H

MOV AH, 0

MOV SI, 0

MOV BX, OFFSET ARRAY

L1: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

MOV DX, SI ; input element of the array

MOV AH, 01H

INT 21H

SUB AL, 30H

;MOV SI, DX

MOV [BX + SI], AX

INC SI

LOOP L1

MOV CX, DATA1

MOV AH, 09

MOV DX, OFFSET msg3

INT 21H

MOV AH, 01 ; Enter element to be searched

INT 21H

SUB AL, 30H

MOV SI, 0

MOV BX, OFFSET ARRAY

L2: CMP [BX + SI], AL ; linear search loop

JZ L3 ; jump if element is found

```
INC SI

LOOP L2


MOV AH, 09H

MOV DX, OFFSET fail ; if the element is not found

INT 21H

MOV AH, 4CH ; to forcefully terminate the program

INT 21H

L3: MOV AH, 09H

MOV DX, OFFSET success ; if the element is found

INT 21H

MOV AH, 4CH

INT 21H

.EXIT

END
```

output

```

C:\TASM>tasm lins.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:   lins.asm
Error messages:   None
Warning messages:  None
Passes:           1
Remaining memory: 474k

C:\TASM>tlink lins.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>lins.exe

Enter the size of the array :: 4
Enter the array ::
2
1
9
6
Enter the element to be searched :: 2
Element is present in the array
C:\TASM>_

```

Code:

```
;Binary search
```

```
.model small
```

```
.386
```

```
.data
```

```
ARRAY DW 20 DUP (?)
```

```
DATA1 dw 0000H
```

```
DATA2 dw 0000H
```

```
success db 10,13,"Element is present in the array $"
```

```
fail db 10,13,"Element is not present in the array $"
```

```
msg db 10,13,"Enter the size of the array :: $"
```

```
msg2 db 10,13,"Enter the array :: $"
```

```
msg3 db 10,13,"Enter the element to be searched :: $"
```



```

.code

.startup

MOV AH, 09

MOV DX, OFFSET msg

INT 21H

MOV AH, 01

INT 21H

SUB AL, 30H

MOV AH, 0

MOV CX, AX

MOV DATA1, AX


MOV AH, 09

MOV DX, OFFSET msg2

INT 21H

MOV AH, 0

MOV SI, 0

MOV BX, OFFSET ARRAY

L1: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

MOV DX, SI ; input element of the array

```

MOV AH, 01H

INT 21H

SUB AL, 30H

MOV SI, DX

MOV [BX + SI], AX

INC SI

LOOP L1

MOV AH, 09

MOV DX, OFFSET msg3

INT 21H

MOV AH, 01 ; Enter element to be searched

INT 21H

SUB AL, 30H

MOV DATA2, AX

MOV CX, DATA1

MOV SI, 0

MOV DI, DATA1

MOV BP, 0

MOV BX, OFFSET ARRAY

```

MOV AX, DATA1

L2: MOV SI, DI

ADD SI, BP

MOV AX, SI

MOV DL, 2

DIV DL


MOV AH, 0

MOV DX, 0

MOV SI, AX

MOV DX, DATA2

CMP [BX + SI], DL

JZ L3

CALL L4

LOOP L2

MOV AH, 09H

MOV DX, OFFSET fail ; if the element is not found

INT 21H

MOV AH, 4CH ; to forcefully terminate the program

INT 21H


L3: MOV AH, 09H

```

```
MOV DX, OFFSET success ; if the element is found

INT 21H

MOV AH, 4CH

INT 21H


L4 PROC NEAR

CMP [BX+SI], DL

JL L6

MOV DI, SI

RET

L6: MOV BP, SI

RET

L4 ENDP

.EXIT

END
```

output

```

Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:  bins.asm
Error messages:   None
Warning messages: None
Passes:          1
Remaining memory: 474k

C:\TASM>tlink bins.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>bins.exe

Enter the size of the array :: 5
Enter the array ::
3
6
7
8
9
Enter the element to be searched :: 7
Element is present in the array
C:\TASM>

```

Q5 Write a program for binary to ascii conversion

Code:

;6. Write a program for binary to ascii conversion

```
.model small
```

```
.data
```

```
    array db 8 dup(?)
```

```
    msg db 0dh,0ah,'Program for conversion of binary to
ascii:$'
```

```
    msg1 db 0dh,0ah,'Enter the element to array:$'
```

```
.code
```

.startup

mov dx,offset msg

mov ah,09h

int 21h

mov dx,offset msg1

mov si,0

mov cx,8

again:

mov ah,01h

int 21h

sub al,30h

mov array[si],al

inc si

loop again

mov cx,8

mov al,01h

mov sp,0h

mov si,07h

```

mov bl, 02h

again1:

    mov dl, array[si]

    cmp dl, 01h

    jz  12

here:

dec si

mul bl

loop again1


jmp ext


12:

    add sp, ax

    jmp here


ext:

mov dx, sp

mov ah, 02h

int 21h

end

```

output

```
Z:\>mount c: c:/
Drive C is mounted as local directory c:/\

Z:\>c:

C:\>cd tasm

C:\TASM>tasm btoa.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:   btoa.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  475k

C:\TASM>tlink btoa.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>btoa.exe

Program for conversion of binary to ascii:010010011
```

Q6. Write a program for ascii to binary conversion

```
;Write a program for ascii to binary conversion.
```

```
.model small
```

```
.data
```

```
    msg db 0dh,0ah,'Program for converting ASCII to  
Binary:$'
```

```
    msg1 db 0dh,0ah,'Enter the element :$'
```

```
.code
```



```

.startup

mov dx,offset msg

    mov ah,09h

    int 21h

    mov dx,offset msg1

    mov ah,09h

    int 21h

    mov ah,01h

    int 21h

    mov bl,al

    mov dl,0Ah

    mov ah,02h

    int 21h

    mov cx,8

again:

    shl bl,1

    jc 12

    jnc 13

    loop again

12:

    mov dl,31h

    mov ah,02h

```

```
    int 21h  
    jmp 14  
13:  
    mov dl,30h  
    mov ah,02h  
    int 21h  
    jmp 14  
14:  
    loop again  
  
End
```

output

```

7
The sorted array is ::
3
5
7
8
C:\TASM>tasm atob.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:  atob.asm
Error messages:   None
Warning messages: None
Passes:          1
Remaining memory: 475k

C:\TASM>tlink atob.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

C:\TASM>atob.exe

Program for converting ASCII to Binary:
Enter the element :9
00111001_

```

Q7. Write a program to add and subtract two arrays

Code:

```
;ARRRAY addition & subtraction
```

```
.model small
```

```
.stack 1000h
```

```
.data
```

```

arr1 db 05h, 06h, 07h, 09h, 09h

arr2 db 01h, 01h, 01h, 01h, 01h

diff db '$$'

sum db '$$$'

space db '$'

endl db 0ah,'$'

.code

printSpace proc stdcall

    mov ah, 09h

    mov dx, offset space

    int 21h

    ret

printSpace endp

printSum proc stdcall

    mov ah, 09h

    mov dx, offset sum

    int 21h

    call printSpace

    ret

printSum endp

printSub proc stdcall

    mov ah, 09h

```

```

    mov dx, offset diff

    int 21h

    call printSpace

    ret

printStats endp

addition proc stdcall

    mov cx, 5

    mov si, 0

a:

    xor bx, bx

    mov bl, byte ptr[arr1+si]

    add bl, byte ptr[arr2+si]

cmp bl, 09h

    ja b

c:

    or bx, 3030h


    mov byte ptr[sum], bh

    mov byte ptr[sum+1], bl


    call printSum

    inc si

```

```

        loop a
ret

b:

    xor ax, ax

    mov al, bl

    sub al, 0ah    ; sub bl, 09h    dec bl

    inc ah

    mov bx, ax

    jmp c
addition endp

subtraction proc stdcall

    mov cx, 5

    mov si, 0

s:

    xor ax, ax

    xor bx, bx

    mov al, byte ptr[arr1+si]

    mov bl, byte ptr[arr2+si]

    sub al, bl

```

```

    or al, 30h

    mov byte ptr[diff], al


    call printSub

    inc si

    loop s

    ret

subtraction endp

start:

    mov ax, @data

    mov ds, ax

    mov es, ax

    call addition

    mov ah, 09h

    mov dx, offset endl

    int 21h


    call subtraction

    jmp last

last:

    .exit 0

end start

```

output

```
C:\TASM>tlink as2a.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International

C:\TASM>as2a.exe
18 18 18 18 18
0 0 0 0 0
C:\TASM>tasm as2a.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:  as2a.asm
Error messages:   None
Warning messages: None
Passes:          1
Remaining memory: 474k

C:\TASM>tlink as2a.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International

C:\TASM>as2a.exe
06 07 08 10 10
4 5 6 8 8
C:\TASM>_
```


