GROUP#11 2 - ITI

- ARRIOLA, JIN
- GOMEZ, MIGUEL
- GUEVARRA, KYLE

NOTE:

- When code is copied from text editor, the pasted borders or lines becomes obscured.
- When assembling our program, we use “tasm /m2 fin.asm”. not the conventional “tasm fin.asm”

SOURCE CODE

```
.model small

.code

.386

org 100h

start:jmp menu


    a db"旼컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴커$"

    b db"?                PH FOOD                    ?"

    c db"?   旼컴?     旼컴?     旼컴?     旼컴?     ?"

    d db"?   ?9 ?     ?9 ?     ?9 ?     ?9 ?     ?$"

    e db"?   읕컴?     읕컴?     읕컴?     읕컴?     ?"

    f db"?  [t]Tinola     [a]Adobo  [s]Sinigang   [d]Dinuguan    ? "

    g db"냐컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴캑$"

    h db"?                                    ?  "

    i db"?   what? [_]     qty [_]     Buy again:y/n[_]      ? "

    j db"읕컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴컴켸$"

k db" Invalid transaction.../ 'q' to quit       $"
```

```
l db" zero! $"
m db" Invalid command (press 'q' to exit) $"
n db"Sorry! Product unavailable.. / 'q' to quit  $"
o db"                                $"
p db"   Answer should be 'y' or 'n'   $"
q db"Sorry! We're Closed.$"

r db ?
w db ?
x db ?
y db ?
z db ?

carrymo db ?
in1 db ?
in2 db ?
in3 db ?
in4 db ?

ad1 db "input a two-digit number:$"
ad2 db 10,13,"input another two-digit number:$"
ad3 db 10,13, "the sum is :$"
invalidinput db 10, 13, "The input must be a number, try again. or [q] to go to the menu$"


men1 db 10, 13,  "Menu:$"
men2 db "[1] - Programmer's Name:$"
men3 db "[2] - ASCII characters$"
men4 db "[3] - Sum of a two-digit number$"
men5 db "[4] - Store Program$"
men6 db "[Q] - Quit$"
```

```
men7 db "Enter Choice:$"

name1 db "Arriola, Jin$"
name2 db "Gomez, Miguel$"
name3 db "Guevarra, Kyle$"
uulit db 10, 13, "'y' to add again. otherwise it will go to the menu.$"

invamenu db 10, 13, "INVALID INPUT PLEASE TRY AGAIN!!!!!$"
menu proc
    call menu1
    call una
    menu endp

menu1 proc


  lea dx, men1
  call string
  call down
  lea dx, men2
  call string
  call down
  lea dx, men3
  call string
  call down
  lea dx, men4
  call string
  call down
  lea dx, men5
  call string
  call down
```

```
   lea dx, men6
   call string
   call down
   lea dx, men7
   call string
   menu1 endp

una proc
    call input
    cmp al, 31h ;name
    je prog
    cmp al, 32h ;ascii
    je ascii
    cmp al, 33h ;sum
    je sum
    cmp al, 34h ;store
    je store
    cmp al, 'q'
    je tapos
    lea dx, invamenu
    call string
    call down
    call menu
   ret
    una endp

checkinput proc near
       cmp al, 71h    ;quit
       je menu

       cmp al, 30h
```

```
        jb invaewan
        cmp al, 39h
        ja invaewan
        ret
        checkinput endp

invaewan:
        call madamidown
        call madamidown
        lea dx, invalidinput
        call string
         jmp sum



tapos:
      int 20h

ascii:
        call down
        mov cx,256
        mov ah, 2
        mov dl,0
    ulit:
        int 21h
        inc dl
        loop ulit
        call madamidown2
        call menu

madamidown proc
         mov cx, 15
```

```
        int 21h
  yomama:
        call down
        loop yomama
        ret
        madamidown endp

madamidown2 proc
        mov cx, 13
        int 21h
  yomama2:
        call down
        loop yomama2
        ret
        madamidown2 endp

prog:

      call down
      lea dx, name1
      call string
      call down
      lea dx, name2
      call string
      call down
      lea dx, name3
      call string
      call madamidown
      call menu
      call una
```

```
store:
      call main

sum:
       call down
       mov carrymo, 30h
       mov in1, 30h
       mov in2, 30h
       mov in3, 30h
       mov in4, 30h

       lea dx, ad1
       call string

       call input
       call checkinput
       mov in1, al  ;1st number
       call input
       call checkinput
       mov bh, al   ;2nd number


       lea dx, ad2
       call string

       call input
       call checkinput
       mov in3, al  ;3rd
       call input
       call checkinput
       mov cl, al   ;4th
```

```
add bh, cl ;add 2nd and 4th number
sub bh, 30h


mov cl, in3
add in1, cl ;adds 1st and 3rd number
sub in1, 30h

cmp bh, 39h
ja carry    ;if ones is above 39h

cmp in1, 39h
ja hala     ; if tens is above 39h

mov cl, in3
sub in1, 30h
lea dx, ad3
call string
mov dl, in1
call output
mov dl, bh
call output
call down
lea dx, uulit
call string
call input
cmp al, 'y'
je sum
jne menu
```

```
carry:
        sub bh, 0ah
        inc in1

       ; add in1, cl  ;ginawa na to sa sum



       cmp in1, 39h
       ja hala

       lea dx, ad3
       call string
       ; sub in1, 30h
        mov dl,in1  ;tens digit
        call output
        mov dl, bh  ;ones digit
        call output
        call down
        lea dx, uulit
        call string
        call input
        cmp al, 'y'
        je sum
        jne menu

hala:
        sub in1, 0ah ; minus tens kasi sobra na sa 39h

        lea dx, ad3  ; print yung "sum is: "
        call string
```

```
mov dl, 31h     ;output yung mga number
call output
mov dl, in1
call output
mov dl, bh
call output
call down
call down

lea dx, uulit  ; tanong kung uulit
call string
call input
cmp al, 'y'
je sum
jne menu
```

```
string proc near
    mov ah,9
    int 21h
    ret
    string endp

down proc near
    mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h
    ret
    down endp

cursor proc near
    mov ah,2
    mov bh,0
    int 10h
    ret
    cursor endp

input proc near
    mov ah,1
    int 21h
    ret
```

```
    input endp

output proc near
    mov ah,2
    int 21h
    ret
    output endp

main proc near
    mov w,39h
    mov x,39h
    mov y,39h
    mov z,39h
    mov r,30h

    mov ax,3
    int 10h

    lea dx,a
    call string
    call down

    lea dx,b
    call string
    call down

    lea dx,c
    call string
    call down

    lea dx,d
```

```
call string
call down

lea dx,e
call string
call down

lea dx,f
call string
call down

lea dx,g
call string
call down

lea dx,h
call string
call down

lea dx,g
call string
call down

lea dx,i
call string
call down

lea dx,j
call string
jmp ag
```

```
yow1: cmp x, 030h ; yung mga "yow" para malaman kung wla na lahat ng stock"
    je yow2
    jne ag69      ; kung hindi sa ag69 pumuta mag loloop palagi sa ag
yow2: cmp y, 030h
    je yow3
    jne ag69
yow3: cmp z, 030h
    je yow4
    jne ag69
yow4: mov dl, 1
    mov dh, 7
    call cursor
    lea dx, q
    call string
    call down
    call down
    call down
    jmp exit

zerona:
      mov dl, 1
      mov dh, 7
      call cursor
      lea dx, n
      call string
      jmp ag

exit7: jmp exit8

ag:
      cmp w,030h  ;para malaman kung out of stock na lahat/ punta sa yow
```

```
        je yow1

ag69:   mov dh,9   ; row    ; crusor ng what?
        mov dl,13 ; column
        call cursor

        call input   ; input para sa pagkain

        mov dl, 1    ; paramawala yung error msg kung meron
        mov dh, 7
        call cursor
        lea dx, o
        call string


        cmp al,'t'
        je tino
        cmp al,'a'
        je ado9
        cmp al,'s'
        je sin9
        cmp al,'d'
        je din9
        cmp al, 'q'
        je exit9

        mov dl, 1
        mov dh, 7
        call cursor
        lea dx, k
        call string
```

```
        jmp ag



ag9: jmp ag

kulang:
        mov dl, 1
        mov dh, 7
        call cursor
        lea dx, k
        call string
        jmp ag

tino:
       cmp w, 030h  ;kung ubos ng tinola stock
       je zerona



        mov dl,27  ; cursor ng qty
        mov dh,9
        call cursor
        call input

        cmp w, al ;kung kulang stock ng w ; if not tuloy lang
        jb kulang

        sub w,al     ; w= tinola
        add w,30h

        mov dl,9  ;cursor ng number ng tinola
        mov dh,3
```

```
        call cursor

        mov dl,w
        call output

        mov dl,49 ; cursor ng buy again
        mov dh,9
        call cursor

        call input
        cmp al, 'y'
        je ag9      ;AYAW PUMUNTA NG AG PAG NAUBOS STOCK
        cmp al, 'n'
        je exit9
        call invb

ado9: jmp ado
zerona8: jmp zerona
sin9: jmp sin
din9: jmp din
kulang9: jmp kulang
exit9: jmp exit8

ado:
        cmp x, 031h  ;kung ubos ng ado stock
        jb zerona9

        mov dl,27  ; cursor ng qty
        mov dh,9
        call cursor
        call input    ;input kung ilan isubtrac
```

```
        cmp x, al ;kung kulang stock ng w ; if not tuloy lang
        jb kulang



        sub x,al      ;x = adobo
        add x,30h     ; kaylangan add 30h para makuha yung number (ascii)

        mov dl,23  ;cursor ng number ng adobo
        mov dh,3
        call cursor

        mov dl,x
        call output

        mov dl,49 ; cursor ng but again
        mov dh,9
        call cursor

        call input
        cmp al, 'y'
        je ag8
        cmp al, 'n'
        je exit8
        call invb

kulang8: jmp kulang9
zerona9: jmp zerona8
ag8: jmp ag9
```

```
sin:
    cmp y, 031h  ;kung ubos ng sinigang stock
    jb zerona9



    mov dl,27  ; cursor ng qty
    mov dh,9
    call cursor
    call input    ;input kung ilan isubtrac

    cmp y, al ;kung kulang stock ng w ; if not tuloy lang
    jb kulang9



    sub y,al     ; y = sin
    add y,30h     ; kaylangan add 30h para makuha yung number (ascii)

    mov dl,35  ;cursor ng number ng sinigang
    mov dh,3
    call cursor

    mov dl,y
    call output

    mov dl,49 ; cursor ng but again
    mov dh,9
    call cursor

    call input
    cmp al, 'y'
    je ag8
```

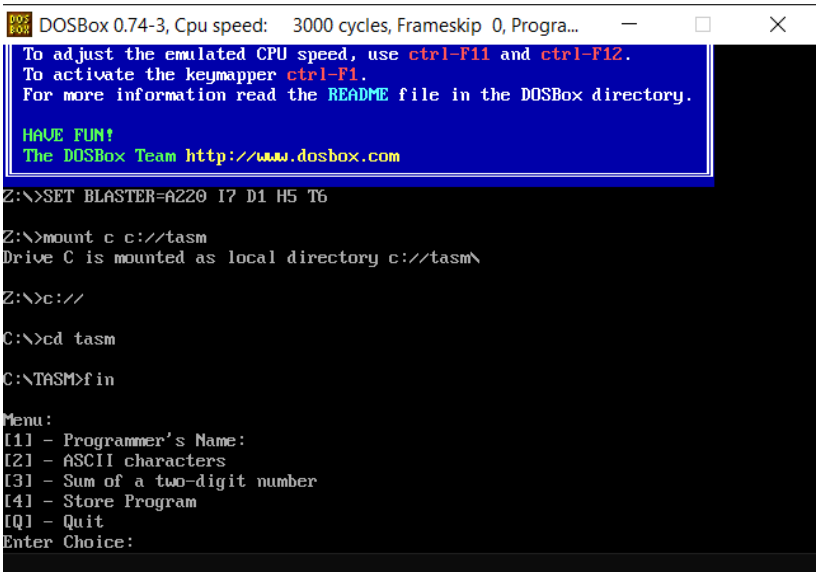
```
        cmp al, 'n'
        je exit
        call invb

exit8: jmp exit
ag7: jmp ag8

din:
        cmp z, 031h  ;kung ubos ng dinuguan stock
        jb zerona9



        mov dl,27  ; cursor ng qty
        mov dh,9
        call cursor
        call input    ;input kung ilan isubtrac

        cmp z, al ;kung kulang stock ng w ; if not tuloy lang
        jb kulang8



        sub z,al      ; z = sin
        add z,30h     ; kaylangan add 30h para makuha yung number (ascii)

        mov dl,48  ;cursor ng number ng dinuguan
        mov dh,3
        call cursor

        mov dl,z
        call output
```

```
        mov dl,49 ; cursor ng but again
        mov dh,9
        call cursor

        call input
        cmp al, 'y'
        je ag7
        cmp al, 'n'
        je exit
        call invb

exit:
        call down
        call down
        call menu
        main endp

invb proc far      ; pag invalid input sa buy again
        mov dl, 1
        mov dh, 7
        call cursor
        lea dx, p
        call string
        mov dl, 49
        mov dh, 9
        call cursor
        call input
        cmp al, 'y'
        je ag7
        cmp al, 'n'
```

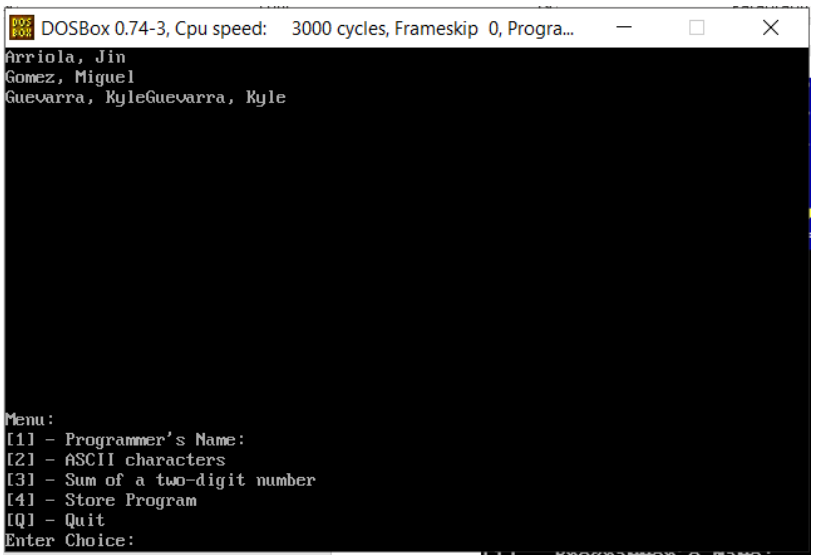
```
        je exit
        call invb
        ret
        invb endp

end start
```

TEST CASES (SCREENSHOTS)

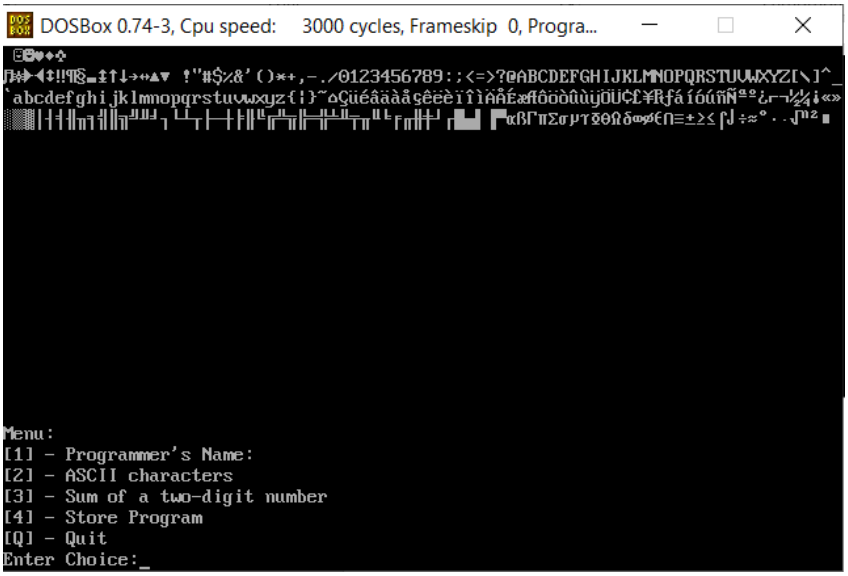
Menu



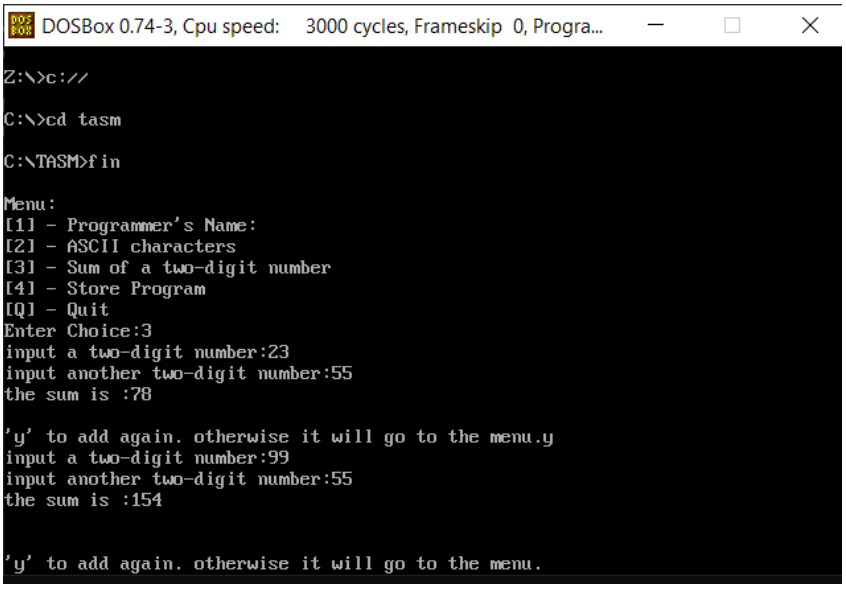
Programmer's Name



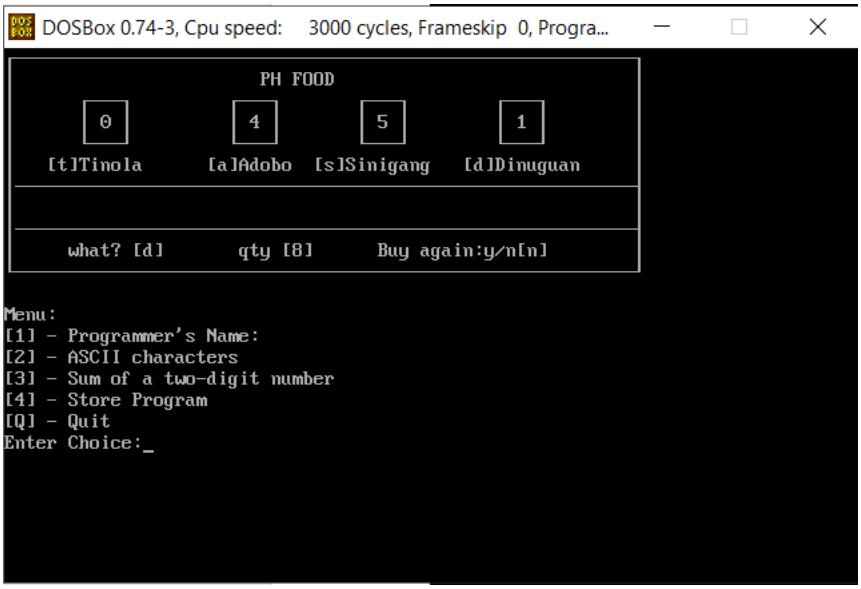
ASCII Characters



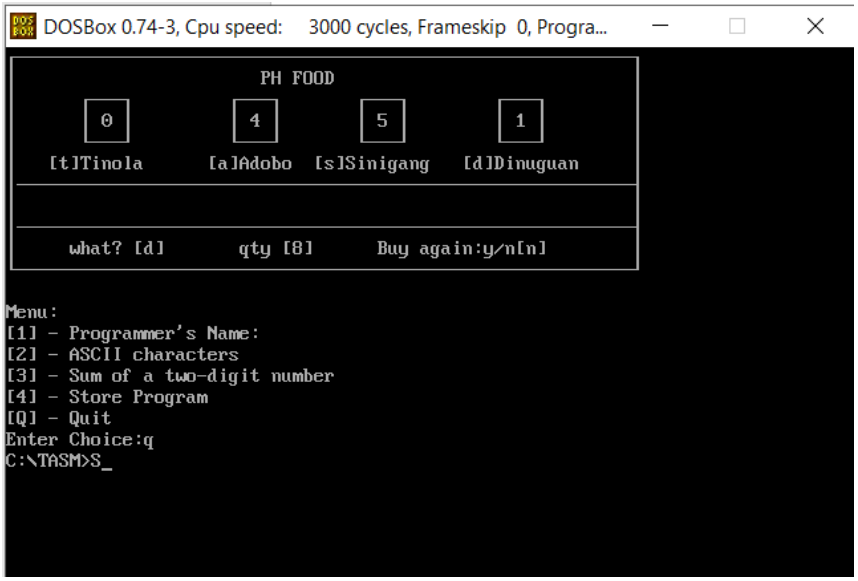
Sum



Store



q to quit



Reflection

Gomez, Miguel

- If possible use your own declared variables when storing string or numbers to lessen the use of general registers. Note that when adding or subtracting declared variables e.g.(add x, y) you will be prompted with an error. You must first mov one of your declared variable in a general register before the assembler will accept the program. E.g.(add x, cl)
- Be mindful when to use procedures or subroutines when creating the logic of your program

Arriola, Jin

- Throughout the activity, we mostly used some of the codes of the programs that we already did in the past, so it was quite convenient to just reuse them and compile them into a single program where we can select each code on the menu. We had to add some procedures and subroutines to make everything seamlessly fit with each other, and for them to be integrated in the menu. Other than that, the only hard part we encountered was when we had to ask a user to input two two-digit numbers and adding them. We only knew how to add single-digit numbers, and it was quite complicated for me to keep track of everything when we had to deal with this problem. Thankfully, as we worked together, and with the help of our Professor's code and reminders, we were able to go through this problem and display the necessary output.
- As I familiarized myself with debug and turbo assembler, I realized just how important assembly language is. While it's not as popular as some high-level programming languages out there, it's definitely underrated when it comes to its speed, optimization, and control over hardware operations. It's quite easier to understand too, as it's more primitive compared to the various commands and functions present in high-level programming languages.

Guevarra, Kyle

- Using the procedure function really helped us lessen our code and work faster because of how much repetitive some functions are needed when we can simply call the procedure to lessen it. It is very convenient when writing long codes and it does not have any limit on how many times you use it or how long the content of the procedure it. You just need to know when and how to use it. Familiarizing myself with the commands and the service numbers made our lives a lot easier since when we need to have an input or display an output, we know the right service command for it.

Member Contribution

We did this activity by sharing Miguel's screen to his group mates. While he writes the code, all of them together brainstormed when creating the program.

Members Evaluation

Gomez, Miguel

- Guevarra, Kyle 10/10
- Arriola, Jin 10/10

Arriola, Jin

- Gomez, Miguel 10/10
- Guevarra, Kyle 10/10

Guevarra, Kyle

-Gomez, Miguel 10/10

- Arriola, Jin 10/10