

«به نام خدا»

استاد: دکتر میثم عبداللهی

نام: فاطمه زهرا بخشنده

شماره دانشجویی: 98522157

گزارش تمرین چهارم:

توضیحات:

فایل مربوط به هر سوال تمرین، در این فایل زیپ قرار داده شده است. سوال اول با استفاده از C و سوال 2 و 3 با emu8086 با روش Simplified-Format نوشته شده اند.

توضیحات الگوریتم هر سوال:

سوال اول:

تابع `int86` برای ایجاد وقفه 8086 استفاده می شود. با این تابع میتوان انواع وقفه را اجرا کرد. شماره وقفه به صورت یک عدد هگز به عنوان آرگومان اول تابع داده میشود.

تابع `intdos` نیز برای ایجاد وقفه 8086 استفاده می شود، اما به طور خاص یک وقفه از نوع DOS را اجرا میکند که عدد آن 21 هگز است. مانند زمانی که در اسمبلی رجیستر هارا ست کرده و دستور `INT 21` را میزنیم که به معنی ایجاد وقفه با عدد 21 هگز است.

در این سوال به دو روش فوق تاریخ کنونی را گرفته و از تابع تبدیل میلادی به شمسی، استفاده کرده، سپس آن را به فرمت خواسته شده چاپ می کنیم.
فایل `converter.h` در کنار فایل اصلی برنامه قرار دارد.

در زمان استفاده از `int86` نوع وقفه را نیز به عنوان اولین ورودی تابع به صورت هگز به آن پاس میدهم.

خروجی با استفاده از int86:

```
Turbo C++ Version 3.00 Copyright (c) 1992 Borland International
main.c:
Turbo Link Version 5.0 Copyright (c) 1992 Borland International

    Available memory 4106664
gregorian date using int86: 2021/12/20
persian date using int86: 1400/9/29

Press any key to continue.
```

خروجی با استفاده از intdos:

```
Turbo C++ Version 3.00 Copyright (c) 1992 Borland International
main.c:
Turbo Link Version 5.0 Copyright (c) 1992 Borland International

    Available memory 4106664
gregorian date using intdos: 2021/12/20
persian date using intdos: 1400/9/29

Press any key to continue.
```

سوال دوم:

در این سوال، باید دو عدد از ورودی درخواست کرده و یک منو نشان دهد تا کاربر یک گزینه را انتخاب کند. ابتدا پیام های که می خواهیم در قسمت های مختلف به کاربر نشان دهیم ذخیره کرده، و دیتاهای مورد نیاز مانند عدد اول و دوم را نیز تعریف می کنیم.

```
.MODEL SMALL
.STACK 64
;-----
.DATA
    ASK1 DB 'Enter numbers1: ','$'
    ASK2 DB 'Enter numbers2: ','$'
    MENU DB 'Enter one of the following numbers:', '$'
    OP0 DB '0: Add', '$'
    OP1 DB '1: Sub(absolute, value)', '$'
    OP2 DB '2: Multiply', '$'
    OP3 DB '3: Divide(larger number by the smaller number)', '$'
    OP4 DB '4: Exit', '$'
    CHOICE DB ?
    NUM1 DW ? ; number 1
    NUM2 DW ? ; number 2
    TMP DB ?
```

عدد اول و دوم را از ورودی دریافت کرده و به ترتیب در NUM1 و NUM2 ذخیره میکنیم. اعداد هرکدام یک بایت هستند و طوری از ورودی دریافت کردم که تمام اعداد هگز یک بایتی را میتوانند شامل شوند. چون اعداد یک بایتی و در نتیجه دوروقمی هستند، برای دریافت هر عدد نیاز به گرفتن 2 کاراکتر از ورودی است. (اگر کاراکتر دریافت شده از 9 بزرگتر بود، 7 را از آن کم میکنیم تا حروف A تا F هگز ساخته شوند). پس از دریافت هر رقم عدد را نیم بایت به سمت چپ شیفت داده و با آن رقم جمع میکنیم. اینگونه هر عدد به طور کامل از ورودی دریافت میشود.

```

029          ;input number 1 and store it in NUM1
030
031 INPUT1:  MOV AH,09
032          MOV DX,OFFSET ASK1
033          INT 21H
034
035 STORE1:  MOV AH,1          ;input function
036          INT 21H
037          SUB AX,130H
038          CMP AL,9
039          JA YES1
040          JMP NO1
041 YES1:    SUB AX,7
042 NO1:    SHL AL,4
043          MOV AH,0
044          MOV NUM1,AX
045          MOV AH,1
046          INT 21H
047          SUB AX,130H
048          CMP AL,9
049          JA YES2
050          JMP NO2
051 YES2:    SUB AX,7
052 NO2:    ADD NUM1,AX

```

پس از آن، پس از نشان دادن منو، کاربر یک عدد بین 0 تا 4 وارد کرده و طبق آن یکی از اعمال خواسته شده انجام میشود:

```

CHOOSE:  MOV AL,CHOICE
          CMP AL,30H
          JZ MYADD
          CMP AL,31H
          JZ MYSUB
          CMP AL,32H
          JZ MYMUL
          CMP AL,33H
          JZ MYDIV
          JMP EXIT

MYADD:    MOV AX,NUM1
          ADD AX,NUM2
          MOV BX,AX
          MOV CX,100H
          JMP PRINT1

```

```

MYSUB:  MOV AX, NUM1
        CMP AX, NUM2
        JB  SWAP
        SUB AX, NUM2
        JMP CONT

SWAP:   MOV AX, NUM2
        SUB AX, NUM1

CONT:   MOV BX, AX
        MOV CX, 10H
        JMP PRINT1

MYMUL:  MOV AX, NUM1
        MUL NUM2
        MOV BX, AX
        MOV CX, 1000H
        JMP PRINT1

MYDIV:  MOV DX, 0
        MOV AX, NUM1
        CMP AX, NUM2
        JB  SWAP2
        DIV NUM2
        JMP CONT2

SWAP2:  MOV AX, NUM2
        DIV NUM1

CONT2:  MOV BX, AX
        MOV TMP, DX
        MOV CX, 10H
        JMP PRINTD

```

برای انجام SUB و DIV باید ابتدا عدد بزرگتر را پیدا کرده سپس عمل را انجام دهیم. پس از انجام آن عمل، باید نتیجه آن را در خروجی چاپ کند. برای اینکار نیز به صورت رقم به رقم عدد را چاپ میکنیم. برای هر کاراکتر اگر 0 تا 9 باشد آن را با 48 جمع کرده و اگر A تا F بود آن را با 55 جمع میکنیم، تا آن را به صورت اسکی بنویسیم.

```

1000 PRINT1: MOV AX, BX
1001          MOV DX, 0
1002          DIV CX
1003          MOV BX, DX      ; remainder
1004          CMP AX, 9
1005          JA  ABOVE
1006          ADD AX, 48
1007          JMP BEL
1008 ABOVE:  ADD AX, 55
1009 BEL:    MOV DX, AX
1010          MOV AH, 02
1011          INT 21H
1012          SHR CX, 4
1013          JNZ PRINT1
1014          JMP EXIT

```

در آخر، با اجرای برنامه و وارد کردن اعداد مختلف و گزینه های متفاوت از منو، نتیجه های زیر را میبینیم:

```
Enter numbers1: F8
Enter numbers2: 13

Enter one of the following numbers:
0: Add
1: Sub(absolute value)
2: Multiply
3: Divide(larger number by the smaller number)
4: Exit
0
10B
```

```
Enter numbers1: 5D
Enter numbers2: FD

Enter one of the following numbers:
0: Add
1: Sub(absolute value)
2: Multiply
3: Divide(larger number by the smaller number)
4: Exit
1
A0
```

```
Enter numbers1: DA
Enter numbers2: 7E

Enter one of the following numbers:
0: Add
1: Sub(absolute value)
2: Multiply
3: Divide(larger number by the smaller number)
4: Exit
2
6B4C
```

```
Enter numbers1: 2F
Enter numbers2: D5

Enter one of the following numbers:
0: Add
1: Sub(absolute value)
2: Multiply
3: Divide(larger number by the smaller number)
4: Exit
3
04
19
```

برای تقسیم، دو عدد در خروجی نمایش داده میشود، خط اول خارج قسمت، و خط دوم باقی مانده تقسیم است.

سوال سوم:

در این سوال، ابتدا پیام ها و دیتاهای مورد نیاز را تعریف میکنیم.

```
.MODEL SMALL
.STACK 64
.DATA
ASK1 DB 'Enter row: ','$'
ASK2 DB 'Enter column: ','$'
ASK3 DB 'Enter the length of the side: ','$'
X DW ?
Y DW ?
LEN DW ?
;side length
```

باید 3 عدد 3 رقمی از کاربر درخواست کنیم. به ترتیب شماره سطر راس قائمه، شماره ستون راس قائمه، و طول ساق مثلث. دریافت اعداد رقم به رقم انجام شده، و مانند سوال قبل است. با این تفاوت که این بار اعداد سه رقمی از ورودی دریافت کردیم.

پس از دریافت ورودی ها، ابتدا صفحه را پاک کرده و mode گرافیکی را تنظیم میکنیم. سپس سه ضلع مثلث را به ترتیب رسم میکنیم.

```
CLEAR: MOV AX,0600H ;SCROLL THE SCREEN
        MOV BH,07 ;NORMAL ATTRIBUTE
        MOV CX,0000 ;FROM ROW=00,COLUMN=00
        MOV DX,184FH ;TO ROW=18H,COLUMN=4FH
        INT 10H ;INVOKE INTERRUPT TO CLEAR SCREEN
        MOV AH,00 ;SET MODE
        MOV AL,13H ;MODE = 13
        INT 10H ;INVOKE INTERRUPT TO CHANGE MODE

        MOV CX,Y ;START LINE AT COLUMN = Y AND
        MOV DX,X ;ROW= X

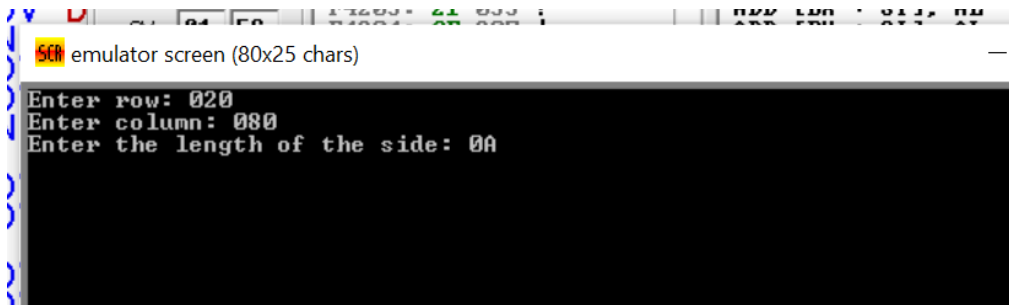
LINE1: MOV AH,0CH ;AH=0CH TO DRAW A LINE
        MOV AL,01 ;PIXELS = WHITE
        INT 10H ;INVOKE INTERRUPT TO DRAW LINE
        INC DX ;INCREMENT VERTICAL POSITION
        MOV BX,X
        ADD BX,LEN
        CMP DX,BX ;DRAW LINE UNTIL ROW = X + LEN
        JNZ LINE1

LINE2: INT 10H ;INVOKE INTERRUPT TO DRAW LINE
        INC CX ;INCREMENT HORIZONTAL POSITION
        DEC DX ;DECREMENT VERTICAL POSITION
        CMP DX,X ;DRAW LINE UNTIL ROW = X
        JNZ LINE2

LINE3: INT 10H ;INVOKE INTERRUPT TO DRAW LINE
        DEC CX ;INCREMENT VERTICAL POSITION
        CMP CX,Y ;DRAW LINE UNTIL COLUMN = Y
        JNZ LINE3
```

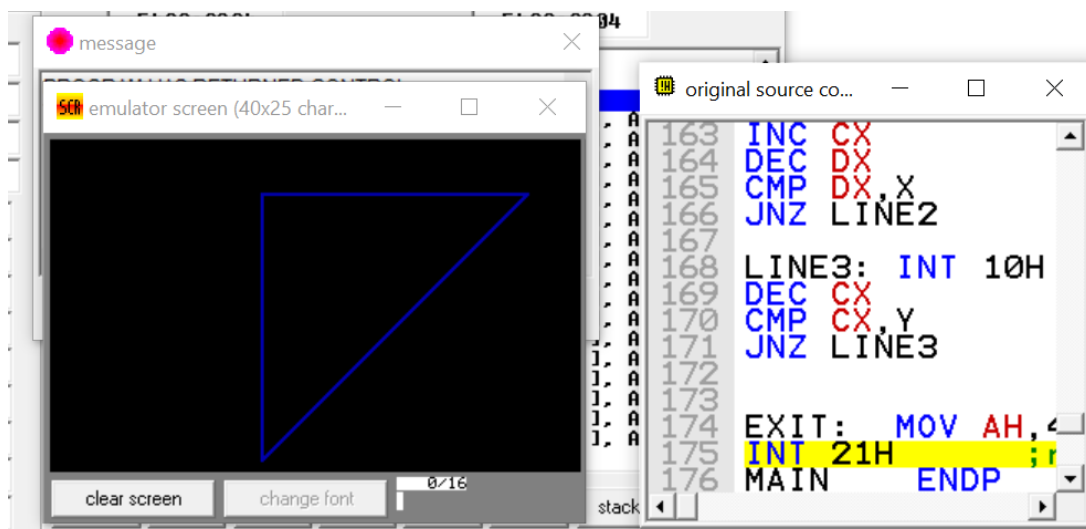
در ابتدا خط عمودی اول را از سطر و ستونی که از ورودی گرفتیم شروع به کشیدن میکنیم، که همان راس قائمه است. سپس از ادامه آن، خط بعدی که ضلع وتر مثلث هست را راس میکنیم. و پس از آن ضلع افقی را رسم کرده و دوباره به راس قائمه می رسمیم.

با اجرای برنامه پیام های زیر نمایش داده میشوند:



```
emulator screen (80x25 chars)
Enter row: 020
Enter column: 080
Enter the length of the side: 0A
```

با وارد کردن داده های ورودی در محدوده مجاز، خروجی را میتوانیم مشاهده کنیم. در اینجا مقدار سطر را 020، ستون را 080 و مقدار طول ضلع را 0A0 وارد کردم.



```
message
emulator screen (40x25 char...)
original source co...
163 INC CX
164 DEC DX
165 CMP DX, X
166 JNZ LINE2
167
168 LINE3: INT 10H
169 DEC CX
170 CMP CX, Y
171 JNZ LINE3
172
173
174 EXIT: MOV AH, 4C
175 INT 21H
176 MAIN ENDP
```

این هم از مثلث قائم الزاویه متساوی الساقین:)