



آیا جزوه را از سایت ما دانلود کرده اید؟

## کتابخانه الکترونیکی PNUEB

پیام نوری ها بستاید

مزایای عضویت در کتابخانه PNUEB :

دانلود رایگان و نامحدود خلاصه درس و جزوه

دانلود رایگان و نامحدود حل المسائل و راهنمای

دانلود کتابچه نمونه سوالات دروس مختلف

پیام نور با جواب

**WWW.PNUEB.COM**

# کتابچه نمونه سوالات چیست:

سایت ما اقتفار دارد برای اولین بار در ایران توانسته است کتابچه نمونه سوالات تمام دروس پیام نور که هر یک حاوی تمامی آزمون های برگزار شده پیام نور (تمامی نیمسالهای موجود **حتی امکان با جواب**) را در یک فایل به نام کتابچه جمع آوری کند و هر ترم نیز آن را آپدیت نماید.

## مراحل ساخت یک کتابچه نمونه سوال

**(برای آشنایی با رحالت بسیار زیاد تولید آن در هر ترم) :**

دسته بندی فایلها - سرج بر اساس کد درس - پسbandن سوال و جواب - پیدا کردن یک درس در نیمسالهای مختلف و پسbandن به کتابچه همان درس - پسbandن نیمسالهای مختلف یک درس به یکدیگر - ولرد کردن اطلاعات تک تک نیمسالها در سایت - آپلود کتابچه و خیلی موارد دیگر..

**همچنین** با توجه به تغییرات کدهای درسی دانشگاه استثنائات زیادی در ساخت کتابچه بوجود می آید که کار ساخت کتابچه را بسیار پیچیده می کند .

## فهرست مطالب

### پیشگفتار

۴	فصل اول . سیستم اعداد
۸	فصل دوم .معماری ریزپردازنده 80286
۱۲	فصل سوم .برنامه نویسی
۱۴	فصل چهارم. دستورالعملهای اساسی
۲۱	فصل پنجم. انشعاب و تکرار
۴۲	فصل ششم.عملیات بیتی
۵۲	فصل هفتم.ماکروها و روالها و وقفه ها
۸۸	فصل هشتم.عملیات پردازش رشته ها
۱۰۰	پاسخنامه سوالات چهار گزینه ای آخر کتاب

## تمرین:

۱. هر بایت از ۸ بیت تشکیل شده است.

۲. هر کیلو بایت معادل ۱۰۲۴ بایت می باشد.

۳. در هر WORD اعداد ۰ تا ۶۵۵۳۵ را می توان جا داد.

۴. در هر بایت اعداد  $-2^7$  تا  $+2^7$  را می توان قرار داد.

۵. در کامپیوتر برای نمایش اعداد منفی از روش مکمل دو استفاده می شود.

۶. در کامپیوتر به جای عمل تفریق از جمع دو مقدار استفاده می گردد.

۷. در هر WORD به تعداد ۶۵۵۳۶ وضعیت مختلفه یک و صفر وجود دارد.

۸. عدد ۲۰ را به سیستم دودویی تبدیل نمایید.

مقدار	تقسیم	نتیجه	باقياندہ
20	2	10	0
0	2	5	0
5	2	2	1
2	2	1	0
1	2	0	1

$$\begin{array}{r} 11101011 + \\ \hline 1 \\ \hline 11101100 \end{array} \leftarrow 00010100 \leftarrow 10100$$

۹. عمل ۲۵-۱۸ را با استفاده از روش مکمل دو انجام دهید.

اول مقدار ۱۸- را به مینا دو برد و بعد با مقدار دودویی ۲۵ جمع می کنیم:

باقیمانده	نتیجه	تقسیم	مقدار
0	9	2	18
1	4	2	9
0	2	2	4
0	1	2	2
1	0	2	1

$$10010 \rightarrow 00010010 \rightarrow \frac{11101101 + 1}{11101110} \rightarrow -18$$

باقیمانده	نتیجه	تقسیم	مقدار
1	12	2	25
0	6	2	12
0	3	2	6
1	1	2	3
1	0	2	1

$$00011001 \rightarrow 25-18 \rightarrow \frac{11101110 + 00011001}{1 00000111} \rightarrow 00000111 \rightarrow 7$$

۱۰. عدد 1101101 را به سیستم دهدی تبدیل نمایید.

$$01101101 \rightarrow 0*2^7 + 1*2^6 + 1*2^5 + 0*2^4 + 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0$$

$$\rightarrow 2^6 + 2^5 + 2^3 + 2^2 + 1 = 109$$

۱۱. عدد 101101.11001 را به سیستم دهدی تبدیل کنید.

$$101101.11001 \rightarrow 1*2^5 + 1*2^3 + 1*2^2 + 1 + 2^{-1} + 2^{-2} + 2^{-3}$$

$$\rightarrow 32 + 8 + 4 + 1 + 0.5 + 0.25 + 0.125 = 45.875$$

۱۲. عدد  $2FABC$  را به سیستم هشت تایی تبدیل نمایید.

$$2FABC \rightarrow 0010\ 1111\ 1010\ 1011\ 1100 \rightarrow (575274)_8$$

۱۳. عدد  $43271$  در سیستم مبنای هشت را به سیستم دودویی تبدیل نمایید.

$$43271 \rightarrow 100\ 011\ 010\ 111\ 001$$

۱۴. مشخص نمایید که اگر MSB یک مقداری یک باشد، آیا آن مقدار منفی است؟

دو حالت دارد که این حالت‌ها را با یک مثال بیان می‌کنیم؛ (برای عدد ۲۵۵)

حالت اول:  $(1111\ 1111)_2$

اگر هر بایت را برای اعداد در بازه‌ی ۰ تا ۲۵۵ در نظر بگیریم، این عدد (255) مثبت است با اینکه آن یک است.

حالت دوم:

اگر هر بایت را برای اعداد در بازه‌ی  $-2^7$  تا  $2^7$  در نظر بگیریم این عدد (-1) منفی است.

۱۵. مشخص نمایید که آیا می‌توان عمل ضرب، تفریق و تقسیم را به عمل جمع تبدیل نمود؟

ضرب: به صورت جمع‌های متواالی  $2+2+2=3*2$

تفریق: به صورت مکمل دو به جمع تبدیل می‌کنیم.

تقسیم: به صورت تفریق بعد تفریق‌ها به جمع تبدیل می‌شود.

۱۶. اگه مقداری منفی باشد:

الف) MSB آن صفر است.  ب) MSB آن یک است.

ج) LSB آن یک است.  د) هیچکدام.

۱۷. اگر مقدار از نوع DOUBLE WORD داشته باشیم به چند بایت حافظه نیاز است؟

الف) ۲  ب) ۳  ج) ۴  د) هیچکدام

۱۸. مقدار 7AB در سیستم شانزده‌دیگر چه مقداری در سیستم دودویی می‌باشد؟

الف) 101110100111  ب) 111101010111

ج) 110110111111  د) هیچکدام

۱۹. مقدار 6 در سیستم دده‌دیگر معادل چه مقداری در سیستم دودویی می‌باشد؟

الف) 11111001  ب) 11111010

ج) 11110111  د) هیچکدام

۲۰. تفاضل کداسکی 'A' و 'a' چیست؟

الف) 30  ب) 32  ج) 40  د) هیچکدام

تمرين:

۱. تعداد ثباتهای هشت بیتی را مشخص نمایید.

تعداد این شتابها ۸ تا می باشد که عبارتند از: AH,AL,BH,BL,CH,CL,DH,DL

## ۲. کار Flag Register را شرح دهید.

ثبات فلگ یا ثبات نشانه، یک ثبات ۱۶ بیتی می باشد که ۱۲ بیت از این ثبات فقط می تواند مورد استفاده برنامه نویسی قرار گیرد، هر کدام از این بیتها نام خاصی دارد و مشخص کننده وضعیت خاصی می باشد.

۳. مکانیزم های انتقال داده ها، اینویسید.

۱. انتقال از یک ثبات به محلی در حافظه      ۲. انتقال از محلی از حافظه به یک ثبات

۳. انتقال یک مقدار ثابت به یک محل در حافظه ۴. انتقال از یک ثبات به یک ثبات دیگر

#### ۵. انتقال مقدار ثابت به نک ثبات

۴. دو مقدار A2FB و CD1A را در مبنای شانزده با هم جمع نموده سپس مقدار فلگ های AF,OF,SF,ZF,PF,CF را مشخص نمایند.

$$\begin{array}{r} \text{A2FB} \rightarrow 101000101111011 + \\ \text{CD1A} \rightarrow 1100110100011010 \\ \hline 1 & 0111000000010101 \end{array}$$

CF=1 دست ۱۵ بک CRARR داریم

$\text{PF}=0$  تهوان زوج ایجاد شده است

SF=0 است علامت صفر است

تمام بسته های نتسجه صفر نیست

$AF=1$  از بیت سوم یک CARRAY خارج شده است

$OF=1$  در بیت ۱۴ CARRAY نداریم ولی در بیت ۱۵ داریم

۵. دو مقدار  $A5B2$  و  $C2BF$  را در مبنای شانزده با هم جمع نموده سپس مقدار فلگهای  $OF, SF, PF, ZF, CF, AF$  را مشخص نمایند.

$$\begin{array}{r} A5B2 \rightarrow 1010010110110010 \\ C2BF \rightarrow 1100001010111111 \\ \hline 1 \quad 0110100001110001 \end{array}$$

$CF=1$  در بیت ۱۵ CARRAY داریم

$SF=0$  برابر با صفر است MSB

$OF=1$  در بیت ۱۴ CARRAY نداریم ولی در بیت ۱۵ داریم

$AF=1$  از بیت سوم یک CARRAY خارج شده است

$ZF=0$  تمام بیت های نتیجه صفر نیس

$PF=1$  توازن فرد ایجاد شده است

#### ۶. موارد استفاده ثباتهای CS, DS, ES, SS را بنویسید.

ثبات CS آدرس شروع ناحیه ای از حافظه که کدهای دستور العمل در آن قرار دارد گذاشته می شود.

ثبات DS آدرس شروع ناحیه ای از حافظه که داده ها در آن قرار دارد گذاشته می شود.

ثبات SS آدرس شروع ناحیه ای از حافظه که پشته در آن ایجاد می شود قرار داده می شود، و در صورتی که در برنامه از دستور العمل های پردازش رشته ها استفاده شود آدرس ابتدای آن ناحیه در حافظه در ثبات ES قرار داده می شود.

#### ۷. صفات دستورالعمل و موارد استفاده آنرا بنویسید.

طول این صفت ۶ بایت می‌باشد و از این صفت برای قرار دان تعدادی دستورالعمل که از حافظه به ریزپردازنده منتقل می‌شود استفاده می‌گردد. اینکار سرعت اجرای دستورالعملها را بیشتر می‌نماید.

#### ۸. دو مقدار 54C2 و 3271 در سیستم مبنای شانزده را از هم کم نموده سپس مقدار فلگهای AF, OF, SF, ZF, PF, CF را مشخص نمائید.

$$(54C2)_{16} - (3271)_{16} \quad 3271 \rightarrow 0011001001110001 \rightarrow \text{مکمل دو}$$

$$\begin{array}{r} 1100110110001110 \\ - 1100110110001111 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 54C2 \rightarrow 0101010011000010 \\ - 1100110110001111 \\ \hline 1 \quad 0010001001010001 \end{array}$$

CF=1

در بیت ۱۵ CARRY داریم

PF=1

توازن فرد ایجاد شده است

SF=0

برابر با صفر است MSB

ZF=0

تمام بیت‌های نتیجه صفر نیست

AF=1

از بیت سوم یک CARRY خارج شده است

#### ۹. آیا تمام دستور العمل‌ها روی فلگ‌ها اثر دارد؟

خیر، برای مثال دستورالعمل MOV بر روی هیچ فلگی تأثیر ندارد.

### ۱۰. کار BUS را بنویسید.

ریزپردازنده از طریق BUS به حافظه متصل می گردد. BUS در هر لحظه شانزده بیت را از حافظه به ریزپردازنده و یا بر عکس منتقل می نماید. داده ها، داده های کنترلی و آدرس می توانند توسط BUS بین حافظه و ریزپردازنده جا به جا گردند.

www\*.PnuEB\*.com

تمرین:

۱. کدامیک از کلمات زیر نام مجاز در اسمبلی می باشد.

BOOK	VARIABLE	\$5AC
2NAME	FOR	.2A_A7
+PLUS	NAME3	BOOK2

اسامی غیر مجاز: +NAME و +PLUS

۲. فیلد های یک دستورالعمل را نام ببرید.

هر دستورالعمل اسمبلی شامل چهار فیلد می باشد: فیلد ملاحظات، فیلد عملوند، فیلد عملیات، فیلد اسم

۳. انواع ثابت ها را بنویسید و در مورد هر یک مثالی بدھید.

1000 B : Binary

40D یا 40 : Decimal

OFFH یا 32H : Hexadecimal

24Q یا 60 : Octal

'B', "Milad" : Character

SING DD 0.332E-1 : Floating point

۴. تعداد حداقل و حداکثر عملوندها در یک دستورالعمل را بدھید.

بعضی از دستورالعمل ها فاقد عملوند می باشد، سایر دستورالملها یک یا دو عملوند دارند که با کاما از هم جدا می شوند.

۵. مدهای آدرس دهی را نام ببرید. و در مورد هر کدام مثالی بدهید.

MOV BL,20

۱.آدرس دهی بدون واسطه

MOV AX,CX

۲.آدرس دهی ثبات

MOV AX,TPPS

۳.آدرس دهی مستقیم

MOV AX,[BX]

۴.آدرسی دهی غیر مستقیم ثبات

MOV AX,[BX]+4

۵.آدرس دهی مبنا

MOV AX,[BX][DI]

عآدرس دهی اندیس مستقیم

MOV AX,[BX][DI+2]

۷.آدرس دهی اندیس مبنا

۶.قانون نامگذاری مشخصه را بیان نمائید.

نام در زبان اسمنلی حداقل می تواند شامل ۳۱ کاراکتر باشد. کاراکترها شامل حروف Z تا A و ارقام ۰ تا ۹ و سیمبولها مخصوص @ ? . \$ \_ می باشد. موارد ذیل بایستی در نام گذاری رعایت گردد:

۱. اسم نمی تواند با یک رقم شروع گردد.

۲. اسم نبایستی یکی از کلمات ذخیره شده در اسمنلی باشد.

۳. در صورتیکه از ۰ در نام استفاده گردد، بایستی اولین کاراکتر نام باشد.

۷. در مورد ثابت ها در مبنای ۱۶ از چه پسوندی استفاده می گردد.

شامل ارقام ۰ تا ۹ می باشد و حروف F تا A می باشد که در انتهای آنها حرف H اضافه می گردد.

۸. آیا استفاده از پسوند در مورد ثابت های مبنای ۱۰ اختیاری است؟

می توان حرف D را به طور اختیاری به آخر اضافه نمود.

تمرین:

۱. دستور العمل **ADD** روی کدام فلگ اثر دارد؟

AF,CF,ZF,OF,SF,PF

۲. دستور العمل **DEC** روی کدام فلگ اثر ندارد؟

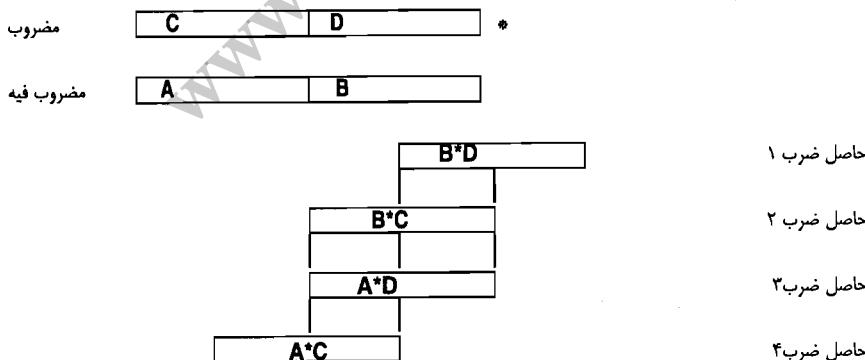
CF

۳. در دستور العمل **IDIV** مقادیر کدام فلگ تعریف نشده می باشد؟

AF,CF,ZF,OF,SF,PF

۴. الگوریتمی را ارایه دهید که دو مقدار از نوع **double word** را در هم ضرب نماید.

در شکل زیر **A,B,C,D** کدام به صورت ۲ بایت در نظر گرفته شده است.



حاصل ضرب نهایی (۶۴ بیتی) = مجموع

۵. قطعه برنامه ای برای الگوریتم تمرین ۴ ارایه کنید.

```

datasg segment para 'data'
HI_MCND    DW  ?
LO_MCND    DW  ?
HI_PP1     DW  ?
LO_PP1     DW  ?
HI_PP2     DW  ?
LO_PP2     DW  ?
HI_PP3     DW  ?
LO_PP3     DW  ?
HI_PP4     DW  ?
LO_PP4     DW  ?
datasg ends

codesg segment para 'code'
main proc far
assume ds:datasg,cs:codesg
PUSH DS
PUSH DI
MOV  DI,datasg
MOV  DS,DI
MOV  HI_MCND,DX
MOV  LO_MCND,AX
MUL  BX
MOV  HI_PP1,DX
MOV  LO_PP1,AX
MOV  AX,HI_MCND
MUL  BX
MOV  HI_PP2,DX
MOV  LO_PP2,AX
MOV  AX,LO_MCND
MUL  CX
MOV  HI_PP3,DX
MOV  LO_PP3,AX
MOV  AX,HI_MCND
MUL  CX
MOV  HI_PP4,DX
MOV  LO_PP4,AX
MOV  AX,LO_PP1
MOV  BX,HI_PP1
ADD  BX,LO_PP2

```

```

ADC  HI_PP2,0
ADD  BX,LO_PP3
MOV  CX,HI_PP2
ADC  CX,HI_PP3
ADC  HI_PP4,0
ADD  CX,LO_PP4
MOV  DX,HI_PP4
ADC  DX,0
POP  DI
POP  DS

mov  AX,4C00H
INT  21H
main endp
codesg ends

end main

```

۶. دستورالعمل های لازم برای محاسبه مجموع ثباتهای ax,bx,cx,dx را ارایه کنید.

```

ADC  AX,BX
ADC  DX,CX
ADC  AX,DX

```

۷. به چند طریق می توان ثبات AX را صفر نمود؟ دستورالعملهای لازم را ارایه کنید.

```

MOV  AX,0
SUB  AX,AX

```

روش های دیگر در فصل های جلوتر بیان می شود.

۸. دستورالعمل MUL برقه فلکهای اثر ندارد؟

فقط بروی فالک های OF و CF تأثیر دارد.

۹. قطعه برنامه ای بنویسید که مجموع عناصر آرایه ۵ عنصری X از نوع word را محاسبه نماید.

```
X      DW    1,2,,3,4,5
LEA    ;I,X
MOV    AX,0
ADD    AX,[SI]
ADD    SI,2
ADD    AX,[SI]
ADD    SI,2
ADD    AX,[SI]
ADD    SI,2
ADD    AX,[SI]
ADC    AX,[SI+2]
```

۱۰. قطعه برنامه ای بنویسید که مجموع عناصر آرایه ۵ عنصری X از نوع double word را محاسبه نماید.

```
X      DD    1,2,,3,4,5
SUM   DD    0
MOV    DI,2
ADD    AX,X
ADC    DX,X[DI]
ADD    DI,2
ADD    AX,X[DI]
ADC    DX,X[DI+2]
ADD    DI,4
ADD    AX,X[DI]
ADC    DX,X[DI+2]
ADD    DI,4
ADD    AX,X[DI]
ADC    DX,X[DI+2]
ADD    SUM ,AX
ADC    SUM+2,DX
```

۱۱. دستور العمل معادل LEA مشخص کنید.

MOV SI,OFFSET X



۱۲. دستور العمل معادل INC BX مشخص کنید.

ADD BX,1

۱۳. قطعه برنامه زیر را به اسembly تبدیل نمایید.

Long int x,y,z,w;

W=x+y-z+30;

```
X    DD  ?
Y    DD  ?
Z    DD  ?
W    DD  ?
MOV   AX,X
MOV   DX,X+2
ADD   AX,Y
ADC   DX,Y+2
SUB   AX,Z
SBB   DX,Z+2
ADD   AX,30
ADC   DX,0
MOV   W,AX
MOV   W+2,DX
```

۱۴. قطعه برنامه ای بنویسید که حاصل ضرب مقادیر ثباتهای DL,CL,BL,AL را محاسبه نماید.

```
MUL   BL      ;AX=BL*AL
MOV   BX,AX
MOV   AL,CL
MUL   DL      ;AX=AL*DL=CL*DL
MUL   BX      ;DX:AX=AX*BX
```

۱۵. قطعه برنامه زیر را به اسembly تبدیل نمایید.

X,Y,Z,W:1..20;
W:=X+Y\*Z-W+100;

این رابطه به این صورت است  $X+(Y*Z)-W+100$



```

X      DB  ?
Y      DB  ?
Z      DB  ?
W      DB  ?
MOV    AL, Y
MUL    Z          ; AX=Y*Z
ADD    AL, X
SUB    AL, W
ADD    AL, 100

```

۱۶. قطعه برنامه ای بنویسید که حاصلضرب عناصر آرایه چهار عنصری X از نوع بایت را مشخص نماید؟ در صورتیکه با مشکلی رو برو شدید ذکر نمایند.

```

X      DB 2,3,4,5
MOV    DI,1
MOV    AL,X
MUL    X[DI]       ; AX=X[0]*X[1]
MOV    BX,AX
INC    DI
MOV    AL,X[DI]   ; AL=X[2]
INC    DI
MUL    X[DI]       ; AX=X[2]*X[3]
MUL    BX           ; DX:AX=AX*BX

```

۱۷. قطعه برنامه معادل زیر به اسembly بدهید.

```

Int x,y,z,w;
W=x-y+z-200;

```

```

X      DW  ?
Y      DW  ?
Z      DW  ?
W      DW  ?
MOV    AX,X
SUB    AX,Y
ADD    AX,Z
SUB    AX,200

```

۱۸. چنانچه بخواهیم محتوی WORD را به یک AL تبدیل نموده و نتیجه را در AX مقدار بدھیم محتوی AH چیست؟

AH = 0000 0000

۱۹. در مورد تمرین ۱۸ چنانچه MSB ثبات AL با برابر با یک باشد محتوی AH چیست؟

AH = 1111 1111

۲۰. دستور العمل لازم برای انجام محاسبه زیر را مشخص کنید.

```
Int x,y,z,w;
W=x/y*z-10
```

X	DW	?
Y	DW	?
Z	DW	?
W	DW	?
MOV	DX, 0	
MOV	AX, X	
DIV	Y	; AX=X/Y
MUL	Z	; DX:AX=(X/Y) * Z
SUB	AX, 10	

تمرین:

۱. دستورالعمل **CMP** مشابه کدام دستور العمل می باشد؟

مشابه دستورالعمل **SUB** می باشد.

۲. آیا استفاده از دستور العمل **CMP X,Y** مجاز می باشد.

خیر، زیرا هر دو عملوند نمی توانند متغیر باشند.

۳. دستورالعمل **JNA** چه موقعی مورد استفاده قرار می گیرد؟

وقتی که **OPR1<=OPR2** باشد این دستور اجرا می شود(برای عملوند های بدون علامت)

۴. دستورالعمل **LE** چه موقعی مورد استفاده قرار می گیرد؟

وقتی که **OPR1<=OPR2** باشد این دستور اجرا می شود(برای عملوند های علامدار)

۵. تفاوت دستور العمل **LOOP** با **LOOPZ** چیست؟

دستورالعمل **LOOPZ** یا **LOOPE** مانند دستورالعمل **LOOP** عمل می نماید با این تفاوت که شرط

تکرار آن است که مقدار **CX** مخالف صفر و مقدار **ZF** برابر با یک باشد.

۶. اگر در ابتدا محتوی **CX** را برابر با صفر قرار دهیم، حلقه تکرار چند بار اجرا می گردد؟

حلقه 65535 بار تکرار می شود.

۷. دستورالعمل های اسمنبلی معادل قطعه برنامه پاسکال زیر بدھید.

```
S:=0;
FOR I=::1 TO N DO
  S:=S+1;
```

```

S DW 0
N DW ?
MOV AX,S
MOV CX,N
M:
    INC S
    ADD AX,S
LOOP M

```

.۸. قطعه برنامه به زبان اسambilی معادل قطعه برنامه زیر را بدهید.

```

X:ARRAY[1..10] OF INTEGER;
S,I:INTEGER;
S:=0;
I:=1;
WHILE I <=10 DO
BEGIN
    S:=S+X[I];
    I:=I+1;
END;

```

```

X      DW  يک آرایه
S      DW 0
MOV    AX,0
MOV    CX,1
MOV    BX,0
FOR1:
CMP    CX,10
JA     EXIT
ADD    AX,X[BX]
ADD    BX,2
INC    CX
JMP    FOR1
EXIT:
MOV    S,AX

```

۹. قطعه برنامه بنویسید که یک آرایه N عنصری از نوع بايت را در نظر بگیرد، تعداد عناصر مثبت آنرا مشخص نماید در متغیر TED قرار دهد.

X DB یک آرایه

N DB 0

TED DB 0

MOV CL, N

MOV CH, 0

INC CX

MOV AX, 0

MOV BX, -1

NEXT:

INC BX

DEC CX

CMP CX, 0

JZ FINISH

CMP X[BX], AX

JLE NEXT

INC TED

CMP CX, 0

JNZ NEXT

FINISH:

۱۰. آیا می توان قطعه برنامه نوشته که N مقدار را بروش مرتب سازی درجی به صورت

صعودی مرتب نماید؟

```
stacksg segment para stack 'stack'
```

```
dw 20 dup(0)
```

```
stacksg ends
```

```
datasg segment para 'data'
```

```
A DW 5,1,3,4,9,6,7,2,8
```

```
I DB 0
```

```
J DB 0
```

```
M DB 9
```

```
N DB 0
```

www\*.PnUEB\*.com

RNUEB

```
datasg ends
codesg segment para 'code'
```

```
PRINT MACRO N
```

ماکرو برای چاپ یک بایت ;

```
LOCAL A
```

```
LOCAL K
```

```
LOCAL M1
```

```
LOCAL M2
```

```
PUSH AX
```

```
PUSH CX
```

```
MOV CX, 0
```

```
A:
```

```
MOV AX, 0
```

```
MOV AL, N
```

```
MOV BL, 10
```

```
DIV BL
```

```
INC CX
```

```
MOV N, AL
```

```
MOV AL, AH
```

```
MOV AH, 0
```

```
PUSH AX
```

```
MOV AL, N
```

```
CMP AL, 0
```

```
JNZ A
```

```
M1:
```

```
POP AX
```

```
MOV DL, AL
```

```
MOV AH, 02H
```

```
ADD DL, 48
```

```
INT 21H
```

```
LOOP M1
```

```
POP CX
```

```
POP AX
```

```
ENDM
```

```
main proc far ; برنامه اصلی
```

```
assume ss:stacksg, ds:datasg, cs:codesg
```

```
MOV ax, datasg
```

```
MOV ds, ax
```

```
MOV BX, 0
```

```
MOV BL, M
```

www\*.PnueB.com

PnueB

```

MOV I,1
MOV SI,0
FOR1:           قطعه کد مرتب سازی از این قسمت شروع می شود ;
    CMP I,BL
    JNB FINISH
    MOV AL,2
    MUL I
    MOV SI,AX
    MOV AX,A[SI]
    MOV DL,I
    MOV J,DL
FOR2:
    CMP J,0
    JNA L1
    CMP A[SI-2],AX
    JNA L1
    PUSH BX
    MOV BX,A[SI-2]
    MOV A[SI],BX
    POP BX
    SUB SI,2
    DEC J
    JMP FOR2
L1:
    MOV A[SI],AX
    INC I
    JMP FOR1
FINISH:
MOV SI,0
MOV CX,0
MOV CL,M
P:
    MOV AX,A[SI]
    MOV N,AL
    PRINT N
    ADD SI,2
LOOP P
mov AX,4C00H
INT 21H
main endp

```

```
codesg ends
end main
```

خروجی:

```
C:\>BC\BIN>soinsert
123456789
C:\>BC\BIN>
```

الگوریتم مرتب سازی درجی الگوریتمی است که مرتب سازی را با درج در رکوردها در یک آرایه مرتب شده موجود مرتب سازی می کند. در این برنامه آرایه A به صورت پیش فرض داده شده و این برنامه این آرایه از نوع word را از طریق مرتب سازی درجی به صورت صعودی مرتب می نماید. ماکرو print در این برنامه برای چاپ یک عدد به اندازه یک بایت پیاده سازی شده است (ماکرو print با پیاده سازی متفاوت در برنامه های این کتاب بکار رفته است تا با طریقه های متفاوت پیاده سازی این ماکرو آشنا گردید).

قطعه کد مرتب سازی به این صورت است که متغیر m طول آرایه را نگه میدارد و متغیر I برای چرخش اندیس حلقه FOR1 بکار می رود و L برای چرخش اندیس حلقه داخلی FOR2 بکار می رود.

بیمایش عناصر آرایه را انجام می دهد FOR1

عنصر مورد نظر را در زیر آرایه مربوطه در مکان درست درج می کند. FOR2

۱۱. یک قطعه برنامه بدھید که آرایه N عنصری X از نوع بایت را از نظر مکانی وارون نماید.

یک آرایه X DB  
N DB آرایه طول  
MOV AX, 0  
MOV AL, N  
MOV BL, 2  
DIV BL  
MOV AH, 0

```

MOV CX,AX
MOV BX,10
M:
    MOV SI,CX
    SUB BX,SI
    MOV DL,X[BX]
    DEC SI
    XCHG DL,X[AX]
    MOV X[BX],DL
LOOP M

```

۱۲. قطعه برنامه بدهید که آرایه N عنصری X از نوع WORD را در نظر گرفته عناصر مخالف صفر آنرا در آرایه دیگری به نام Y قرار دهد (در صور لزوم انتهای آرایه Y خالی بماند).

```

X DW ?
Y DW ?
N DW ?

```

```

MOV CX,N
MOV SI,-2
MOV DI,0
INC CX
L1:
    DEC CX
    CMP CX,0
    JZ FINISH
    ADD SI,2
    MOV AX,X[SI]
    CMP AX,0
JZ L1
MOV AX,X[SI]
MOV Y[DI],AX
ADD DI,2
JMP L1
FINISH:

```

۱۳. قطعه برنامه بنویسید که مقدار صحیح و مثبت N را در نظر گرفته فاکتوریل آنرا مشخص نماید.

```

datasg segment para 'data'
 maxlen db 20
 actlen db ?
 num db 20 dup(0)
 N DB 0
 codesg segment para 'code'
 main proc far
 assume ds:datasg,cs:codesg
 MOV ax,datasg
 MOV ds,ax

MOV AH ,0AH      ;قطعه کد برای گرفتن عدد از ورودی ;
LEA DX,MAXLEN
INT 21H

MOV AX,0
LEA SI,num
MOV CL,actlen
MOV CH,0
L1:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP L1
MOV DX,0

MOV N,AL      ;مقدار ورودی در متغیر ذخیره می شود ;
CMP N,0
JBE FINISH
MOV CX,0
MOV CL,N      ;قطعه کد مربوط به محاسبه فاکتوریل ;
MOV AX,1
MUL1:
    CMP AH,0
    JNZ MUL2
    MUL CL

```

```

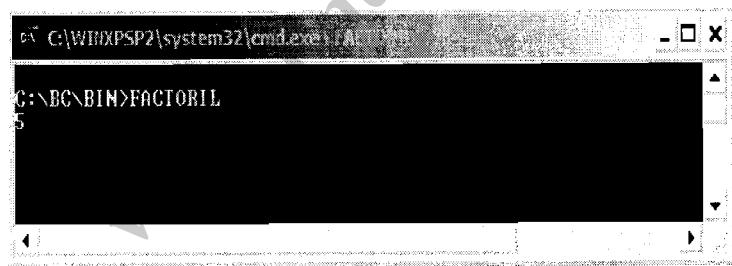
    SUB CL,1
    CMP CL,0
JNZ MUL1
JMP FINISH
MUL2:
    MUL CX
    DEC CX
    CMP CX,0
JNZ MUL2

FINISH:
mov AX,4C00H
INT 21H
main endp
codesg ends

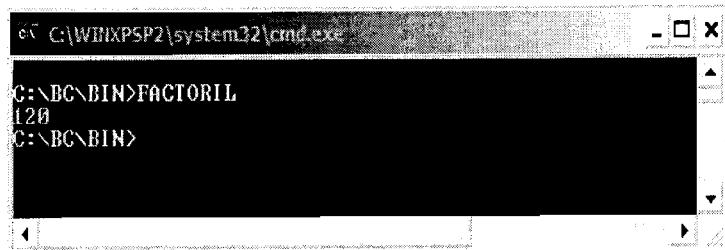
end main

```

ورودی:



خروجی:



در این برنامه قطعه کدی برای گرفتن عدد از ورودی بکار رفته است(در واقع این کد یک رشته را از ورودی گرفته و این رشته را به عدد معادل آن تبدیل می کند).

در این برنامه از دو حلقه MUL1 و MUL2 برای محاسبه فاکتوریل استفاده شده است.

Mul1 ضرب دو بایت را انجام می‌دهد

اگر مقدار حاصلضرب از یک بایت افزایش یابد از ضرب دو word استفاده می‌کنیم.

Mul2 ضرب دو word را انجام می‌دهد

در حالت ضرب دو بایت حاصل فاکتوریل در AX و در حالت ضرب دو word در DX ذخیره می‌شود.

۱۴. قطعه برنامه بنویسید که آرایه N عناصری X از نوع word را در نظر گرفته کوچکترین عنصر آرایه را مشخص نماید.

```

datasg segment para 'data'
A DW 2,1,5,4
N DW ?
datasg ends
codesg segment  ra 'code' e'
main proc far
assume ds:datas cs:codesg sg
MOV ax,datasg
MOV ds,ax
MOV CX,N
MOV SI,0
MOV AX,A[SI]
DEC CX
W:      CMP AX,A[SI+2] ; مقدار مینیمم در ثبات AX قرار دارد
        JBE CONTINUE
        MOV AX,A[SI+2]
CONTINUE:
        ADD SI,2
LOOP W

mov  AX,4C00H
INT  21H
main endp
codesg ends
end main

```

۱۵. قطعه برنامه ای بنویسید که آرایه N عنصری X از نوع double word را در نظر گرفته کوچکترین عنصر آرایه را مشخص نماید.

```

datasg segment para 'data'
A DD 2,1,5,4
N DW 4
datasg ends
codesg segment para 'code'
main proc far
assume ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax
MOV CX,N
MOV SI,0
LEA SI,A
MOV AX,[SI]
MOV DX,[SI+2]
DEC CX

W:
CMP DX,[SI+6]
JB CONTINUE
JE L1
MOV DX,[SI+6]
JMP L2
L1:
CMP AX,[SI+4]
JBE CONTINUE
L2:
MOV AX,[SI+4]
CONTINUE:
ADD SI,4
LOOP W
mov AX,4C00H
INT 21H
main endp
codesg ends
end main

```

با قرار دادن آدرس شروع آرایه در SI و ذخیره اولین عنصر در ثبات های AX:DX، با مقایسه این عنصر با دیگر عناصر توسط دستورات CMP و JMPها کوچکترین عنصر در آرایه را بدست آورده و در ثبات DX:AX ذخیره می کنیم.

۱۶. قطعه برنامه ای بنویسید که آرایه N عنصری X از نوع word را در نظر گرفته اندیس بزرگترین عنصر آرایه را مشخص نماید.

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment istr 'data' a'
A DW 2,1,5,4
M DW 4
N DB 0
datasg ends
codesg segment istr 'code' e'
P MACRO N           ماکرو برای چاپ یک بایت ;
LOCAL A
LOCAL K
LOCAL M1
LOCAL M2
PUSH AX
PUSH CX
MOV CX,0
A:
MOV AX,0
MOV AL,N
MOV BL,10
DIV BL
INC CX
MOV N,AL
MOV AL,AH
MOV AH,0
PUSH AX
MOV AL,N
CMP AL,0
JNZ A

K:
POP AX
MOV BX,0

M1:
MOV DX,0
PUSH CX
MOV CX,10
DIV CX

```

```

POP CX
PUSH AX
PUSH DX
MOV AX,10
MUL BX
MOV BX,AX
POP DX
ADD BX,DX
POP AX
CMP AX,0
JNE M1
MOV AX,BX
M2:
MOV DX,0
PUSH CX
MOV CX,10
DIV CX
POP CX
PUSH AX
MOV AH,02H
ADD DL,48
INT 21H
POP AX
CMP AX,0
JNE M2
POP AX
LOOP K
POP CX
POP AX
ENDM

main proc far ; برنامه اصلی
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV CX,M
MOV SI,0
MOV DI,SI ; اندیس بزرگترین عنصر در DI قرار می گیرد
MOV AX,A[SI]
DEC CX
W:
    CMP AX,A[SI+2]
JAE CONTINUE
    MOV AX,A[SI+2]

```

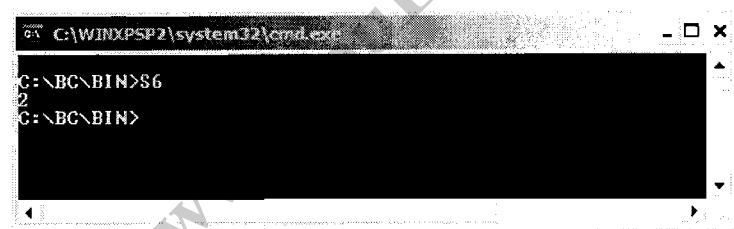
```

    MOV  DI,SI
    ADD  DI,2
CONTINUE:
    ADD SI,2
LOOP W
MOV AX,DI
MOV BL,2
DIV BL
MOV N,AL
P N
mov AX,4C00H
INT 21H
main endp
codesg ends
end main

```

چاپ آرایه از نوع word است اندیس را بر ۲ تقسیم می کنیم ;

: خروجی



عدد ۵، بزرگترین عنصر است؛ با توجه به اینکه اندیس آرایه از صفر شروع می گردد، اندیس عدد ۵ (بزرگترین عنصر) در آرایه ۲ می باشد.

۱۷. قطعه برنامه بنویسید که مجموع زیر را محاسبه نماید.

$$1+2^2+3^2+4^2+\dots+N^2$$

```

datasg segment para 'data'
maxlen db 20
actlen db ?
num db 20 dup(0)
N DB 0
S DW 0
datasg ends
codesg segment para 'code'

```

```
main proc far
assume ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax
```

```
MOV AH, 0AH      ;قطعه کد مربوط به گرفتن عدد از ورودی ;
LEA DX, MAXLEN
INT 21H
```

```
MOV AX, 0
LEA SI, num
MOV CL, actlen
MOV CH, 0
L1:
    MOV BX, 10
    MUL BX
    MOV DL, [SI]
    SUB DL, 48
    MOV DH, 0
    ADD AX, DX
    INC SI
LOOP L1
```

```
MOV N, AL      ;عدد گرفته شده در متغیر مربوطه ذخیره می شود ;
```

```
MOV CX, 0      ;قطعه کد برنامه از اینجا شروع می شود ;
MOV CL, N
```

```
W:
MOV AL, CL
MOV BL, CL
MUL BL
ADD S, AX
LOOP W
```

```
mov AX, 4C00H
INT 21H
main endp
codesg ends
```

```
end main
```

۱۸. قطعه برنامه بنویسید که مشخص نماید آیا عدد N اول می باشد یا خیر؟

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'
N DW 0
ERR DB 'NO$'           ; پیغامی برای خیر
ACC DB  'YES$'          ; پیغامی برای بله
datasg ends
codesg segment para 'code'

main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

PUSH CX
MOV CX,N      ; بافرض اینکه عدد را از ورودی گرفته ایم
CMP CX,1
JZ NO
CMP CX,2
JZ YES
SUB CX ,2
MOV BL,2
M:   MOV AX,N
      DIV BL
      INC BL
      CMP AH,0
LOOPNE M
JZ NO

YES:
MOV AH,09H
LEA DX,ACC
INT 21H
JMP E

NO:
MOV AH,09H
LEA DX,ERR
INT 21H
E:

```

```

POP    CX

MOV  AX,4C00H
INT  21H
main endp
codesg ends
end main

```

۱۹. قطعه برنامه بنویسید که مشخص نماید آیا عدد N کامل می باشد یا خیر؟

```

datasg segment para 'data'
 maxlen db 20
 actlen db ?
 num db 20 dup(0)
 N DW 0
 SUM DW 1
 ERR DB 'NO$'           ;پیغامی برای خیر
 ACC DB 'YES$'          ;پیغامی برای بله ;
datasg ends
codesg segment para 'code'
main proc far
assume ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

MOV AX,0
LEA SI,num
MOV CL,actlen
MOV CH,0
L1:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP   L1

```

عدد گرفته شده در متغیر مربوطه ذخیره می شود ;

```

MOV      N, AX
MOV      CX, N
SUB      CX, 2
MOV      BX, 2
W:       MOV      DX, 0
          MOV      AX, N
          DIV      BX
          CMP      DX, 0
          JNZ      CONTINUE
          ADD      SUM, BX
CONTINUE:
          INC      BX
LOOP    W
MOV      AX, N
CMP      AX, SUM
JZ       YES

MOV      AH, 09H
LEA      DX, ERR
INT      21H
JMP    E

YES:
MOV      AH, 09H
LEA      DX, ACC
INT      21H

E:
mov    AX, 4C00H
INT    21H
main  endp
codesg ends

end main

```

www\*.PnuEB\*.com

 PnuEB

ورودی:

```
C:\WINXPSP2\system32\cmd.exe
C:\>BC\BIN>KAMEL
```

خروجی:

```
C:\WINXPSP2\system32\cmd.exe
C:\>BC\BIN>KAMEL
YES
C:\>BC\BIN>
```

عددی کامل است که مجموع مقسوم علیه های آن به جز خودش، برابر خودش شود. در این برنامه عدد مورد نظر را از ورودی گرفته (البته به صورت رشته و بعد تبدیل آن به عدد معادل) و مقسوم علیه های آن را بدست آورده و آنها را جمع می کنیم و حاصل را با خود عدد مقایسه کرده و در صورت برابری عدد کامل است در غیر این صورت عدد کامل نیست.

۲۰. قطعه برنامه بنویسید که مقدار  $N^X$  از نوع بایت را گرفته  $N^X$  را محاسبه نماید.

```
datasg segment para 'data'
 maxlen db 20
 actlen db ?
 num db 20 dup(0)
 N DB 0
 X DW 3          مقدار توان در این قسمت قرار می گیرد ;
 datasg ends
 codesg segment para 'code'

 main proc far
 assume ds:datasg,cs:codesg

 MOV ax,datasg
```

```
MOV ds,ax
```

```
MOV AH ,0AH
LEA DX,MAXLEN
INT 21H
```

```
MOV AX,0
LEA SI,num
MOV CL,actlen
MOV CH,0
L1:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP L1
```

```
MOV N,AL      عدد گرفته شده در متغیر مربوطه ذخیره می شود ;
```

```
MOV CX,X      قطعه کد مربوط به برنامه ;
MOV AX,1
MOV BX,0
MOV BL,N
MULBIT:
    CMP AH,0
    JNZ MULWORD
    MUL N
LOOP MULBIT
JMP E
MULWORD:
    MUL BX
LOOP MULWORD
```

E:

```
mov AX,4C00H
INT 21H
main endp
codesg ends
```

end main

www.PnuEB\*.com

PnuEB

۲۱. قطعه برنامه ای بنویسید که آرایه N عنصری X از نوع word را گرفته میانه آنرا مشخص نموده در MID قرار دهد.

```

datasg segment para 'data'
X DW 1,2,3,4,5,6
N DW 6
MID DW 0
datasg ends
codesg segment para 'code'
main proc far
assume ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV SI,0
MOV AX,N
MOV BX,0
MOV BL,2
DIV BL
MUL BL
MOV CX,AX
W:
    INC SI
LOOP W
MOV AX,X[SI]
MOV MID,AX

mov AX,4C00H
INT 21H
main endp
codesg ends

end main

```

## تمرین:

۱. اشکال دستورالعمل های زیر را مشخص نمائید.

```
ROL DL, 2
SHL BX , CX
MOV X, [BX]
```

در دستورالعمل (ROL OPR,CNT) اگر مقدار CNT برابر یک باشد عدد را می نویسیم در غیر این صورت از ثبات CL استفاده می کنیم.

در دستورالعمل (SHL OPR,CNT) تعداد بیت هایی می باشد که به طرف چپ شیفت داده می شود، در صورتی که CNT مخالف یک باشد از ثبات CL استفاده می نماییم.

در دستورالعمل MOV هر دو عملوند نمی توانند متغیر باشد.

۲. مقدار CF و BL پس از اجرا دستورالعمل های ذیل چیست؟

MOV	CL, 4	=> CL -> 0000 0100
MOV	BL, 0BBH	=>BL -> 1011 1011
CLC		=> CF=0
SAL	BL, CL	=>BL -> 1011 0000 , CF=1
XOR	BL , CL	=> BL -> 1011 0100 , CF=0
SAR	BL, CL	=> BL -> 1111 1011 , CF=0

۳. مقدار BX و CF پس از اجرا دستورالعمل های ذیل چیست؟

MOV	CL, 3	=> CL -> 0000 0011
MOV	BX, 2ACh	=> BX ->0000 0010 1010 1100
STC		=>CF=1
ROL	BX, CL	=> BX ->0001 0101 0110 0000 , CF=0

INC CL =>CL → 0000 0100 ,CF=0

RCL BX, CL => BX → 0101 0110 0000 0000 , CF==1

#### ۴. دستورالعملهای ذیل روی چه فلکهای اثر دارند؟

RCR, SHL, MOV, TEST, AND, XOR

	OF	DF	IF	TF	SF	ZF	AF	PF	CF
AND	0	-	-	-	*	*	?	*	0
XOR	0	-	-	-	*	*	?	*	0
TEST	0	-	-	-	*	*	?	*	0
MOV	-	-	-	-	-	-	-	-	-
SHL	*	-	-	-	*	*	?	*	*
RCR	*	-	-	-	-	-	-	-	*

#### ۵. برنامه‌ای بنویسید که یک رشته را گرفته تعداد حروف کوچک آنرا مشخص نماید.

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'
 maxlen db 20
 actlen db ?
 s db 20 dup(0)
datasg ends
codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

```

MOV AH ,0AH قطعه کد مربوط به گرفتن رشته از ورودی ؟

LEA DX,MAXLEN

INT 21H

```

LEA      BX, S
MOV      DI, 0
MOV      CL, actlen
MOV      CH, 0
LOOP1:
        MOV      AL, [BX]
        CMP      AL, 61H
        JB       LBL1
        CMP      AL, 7AH
        JA       LBL1
        INC      DI
LBL1:
        INC      BX
        LOOP    LOOP1

```

```

mov AX, 4C00H
INT 21H
main endp
codesg ends
end main

```

در برنامه با گرفتن رشته از ورودی (توجه کنید این قطعه کد شبیه گرفتن عدد از ورودی است با این تفاوت که رشته گرفته شده به مقدار عددی معادل تبدیل نمی گردد).

در این الگوریتم کاراکتر های رشته با محدوده کداسکی حروف کوچک مقایسه می شود و در صورتی که کاراکتر جزء این محدوده باشد به تعداد حروف کوچک یک واحدافروده می شود (INC DI).

۶. برنامه بنویسید که بیت های شماره فرد ثبات AX را به یک و بیت های زوج ثبات AX را به صفر تبدیل نماید.

```

OR      AX, 0AAAAAH
AND    1 AX, 0AAAAAH

```

۷. برنامه بنویسید که مشخص نماید آیا بیت های 2,5,7 ثبات BL برابر با صفر می باشد یا خیر؟

```
stacksg segment para stack 'stack'
```

```

dw 20 dup(0)
stacksg ends
datasg segment para 'data'

datasg ends
codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax
MOV BL,01011010B ; مقدار ثبات را به صورت پیش فرض وارد کرده ایم
MOV CL,3
MOV DX,2
M:
    CLC
    SHL     BL,CL
    JC      NOT_ZERO
    DEC     DX
JNZ    M

MOV CL,2
SHL BL,CL
JC NOT_ZERO

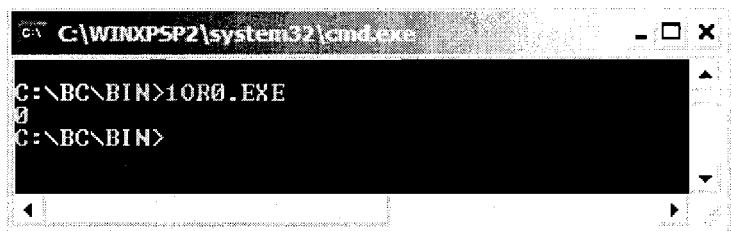
MOV AH,02H
MOV DL,'0'
INT 21H
JMP EXIT

NOT_ZERO:
MOV AH,02H
MOV DL,'1'
INT 21H

EXIT:
mov AX,4C00H
INT 21H
main endp
codesg ends
end main

```

خروجی:



۸. برنامه بنویسید که مشخص نماید آیا بیت های شماره زوج ثبات AX برابر با یک می باشد یا خیر؟

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'

datasg ends
codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax
MOV      AX,0111011101110111B
AND      AX,5555H
CMP      AX,5555H
JZ       YES
NO:

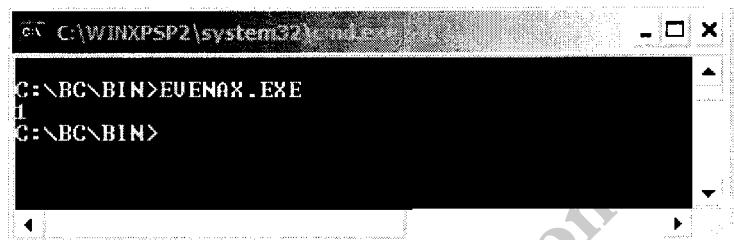
MOV  AH,02H
MOV  DL,'0'
INT  21H
JMP  EXIT

YES:
MOV  AH,02H
MOV  DL,'1'
INT  21H

EXIT:

```

```
mov AX, 4C00H
INT 21H
main endp
codesg ends
end main
```



مقدار ثبات را به صورت پیش فرض وارد کرده ایم و این برنامه را می توان برای مقادیر دیگر برای این ثبات آزمایش کرد.

۹. برنامه ای بنویسید که بیت های ثبات AX را انقدر به طرف چپ شیفت داده تا MSB آن برابر با یک گردد. در صورتیکه در ابتدا MSB برابر با یک می باشد کنترل به منتقل گردد.

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'dat'

datasg ends
codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax
MOV      AX, 0020H; مقدار ثبات را به صورت پیش فرض وارد کرده ایم
TEST     AX, AX
JS       EXIT
MOV      CX, 16
LOOP1:
        SHL    AX, 1
        JS    EXIT
```

```
LOOP      LOOP1
```

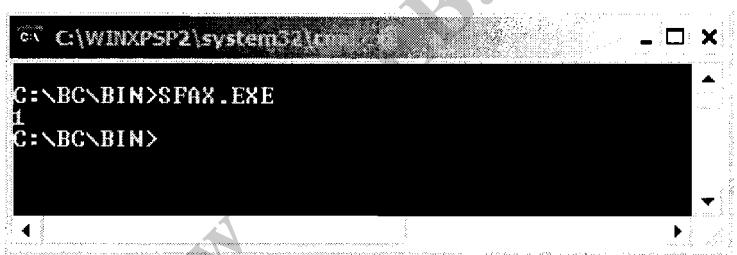
```
JMP    E
```

```
EXIT:
```

```
MOV    AH, 02H
MOV    DL, '1'
INT    21H
```

```
E:
```

```
mov    AX, 4C00H
INT    21H
main  endp
codesg ends
end main
```



#### ۱۰. پس از اجرا دستورالعمل ذیل مقادیر ثباتها چیست؟

MOV	AX, 4BCH	=>AX -> 0000 0100 1011 1100
MOV	DX, 0F2BCH	=> DX -> 1111 0010 1011 1100
XOR	AX, DX	=> AX -> 1111 0110 0000 0000 ,CF=0, OF , SF=1 , ZF=0 , PF=0
SUB	DX, 8	=> DX -> 1111 0010 1011 0100 , CF=0,OF=0 , SF=1,ZF=0,PF=1
NOT	DX	=> DX -> 0000 1101 0100 1011
ADD	AX, 16	=>AX-> 1111 0110 0000 1111 , CF=0,OF,SF=1,ZF=0,PF=0

AND DX, AX => DX ->0000 0100 0000 1011, CF=0, OF=0, SF=0, ZF=0  
, PF=0

CF=0, OF=0, SF=0, ZF=0, PF=0

۱۱. با چه دستوری مقادیر CF, IF, DF برابر با یک می شوند.

به ترتیب STI, STD, STD

۱۲. به چند طریق می توان مقدار ثبات AX را برابر با صفر قرار داد؟

چند روش در فصل های قبل داشتیم:

*MOV AX, 0	*SUB AX, AX	*AND AX, 0
*MOV CX, 16		
M: SHL AX, 1		
LOOP M		
*MOV BL, 0		
DIV BL		
*MOV CL, 16		
SHL AX, CL		
*MOV CL, 16		
SAL AX, CL		
*MOV CL, 16		
SHR AX, CL		
*MOV BX, AX		
NEG BX		
ADD AX, BX		

۱۳. برنامه ای بنویسید که مقدار بیت شماره n ام ثبات AX را مشخص نماید.

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
```

```

datasg segment para 'data'
 maxlen db 20
 actlen db ?
 num db 20 dup(0)
datasg ends
codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH
LEA DX ,MAXLEN
INT 21H

MOV AX,0
LEA SI,num
MOV CL,actlen
MOV CH,0
L1:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP L1

MOV CL,AL      عدد گرفته شده در متغیر مربوطه ذخیره می شود ;
MOV CH,0
INC CL
MOV AX,0020H
MOV DX,AX
CLC
SHR DX,CL      ; یافتن بیت ln
JC ONE

MOV AH,02H
MOV DL,'0'
INT 21H
JMP E
ONE:

```

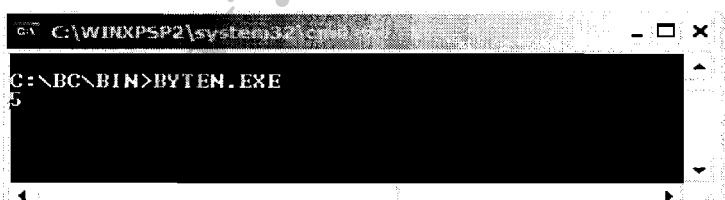
```

MOV AH, 02H
MOV DL, '1'
INT 21H
E:
    mov AX, 4C00H
    INT 21H
    main endp
    codesg ends
end main

```

با ورودی مقدار چندمین بیت ثبات(شماره بیت ها از صفر شروع می گردد) را مشخص می کنیم، سپس برای یافتن بیت مورد نظر آنقدر(یکی بیشتر از ورودی) شیفت به راست می دهیم، بعد از مشخص شدن مقدار بیت مورد نظر، آنرا چاپ می کنیم.

ورودی:



خروجی:



تمرین:

### ۱. تفاوت **procedur** با **MACRO** چیست؟

تفاوت بین روال و ماکرو در این است که ماکرو در برنامه در زمان اجرای برنامه وجود خارجی ندارد بلکه روال در زمان اجرای برنامه وجود داشته و فراخوانی می‌گردد.

### ۲. **macro** ها در چه قسمتی از برنامه قرار می‌گیرند.

**MACRO** ها بایستی در ابتدای برنامه تعریف شوند.

### ۳. **procedure** ها در چه قسمتی از برنامه قرار می‌گیرند؟

در زیر روال برنامه قرار می‌گیرد.

```
MAIN PROC FAR
```

```
.
```

```
MAIN ENDP
```

```
PROC1 PROC NEAR
```

```
RET
```

```
PROC1 ENDP
```

#### ۴. کار عملگرهای ماکرو چیست؟

دو عملگر ماکرو در Macro Assembler عبارتند از `&` ::

عملگر :: باعث می شود که اسمبلر Comments ها را در موقع بسط ماکرو حذف نماید.

عملگر `&` این عملگر دو `text` یا سیمبول را بهم وصل نموده در کنار یکدیگر قرار می دهد.

#### ۵. کار `IFNB, EXITM, REPT` چیست؟ یک مثال برای هر کدام ارائه کنید.

شکل کلی `IFNB` به صورت زیر می باشد. `IFNB` درست است اگر، مقدار argument در موقع فراوانی `IF NOT BLANK` `blank` می باشد. `IFNB` به معنی `IF NOT BLANK` می باشد.

```
ADD_TWO MACRO X1,X2
```

```
IFNB <X1>
```

```
    ADD AX,X1
```

```
ENDIF
```

```
IFNB <X2>
```

```
    ADD AX,X2
```

```
ENDIF
```

```
ENDM
```

از `EXITM` فقط در ماکروها استفاده می گردد. `EXITM` باعث توقف بسط ماکرو می شود بسته به نتایجی که بوسیله directive های شرطی (`IF`) بوجود می آید.

```
IFB <N>
```

```
EXITM
```

```
ENDIF
```

دستورالعمل های بین ENDM و REPT را به تعداد دفعاتی که توسط expression مشخص می گردد تکرار می کند.

REPT expression

ENDM

۶. کار LOCAL چیست؟ یک مثال آرایه دهید.

این دیرکتیو باعث می شود که اسمبلر برای هر عنصر در dummy-list ایجاد یک سیمبول منحصر بفرد بنماید.

LOCAL dummy-list

WAIT MACRO COUNT

LOCAL NEXT

PUSH CX

MOV CX, COUNT

NEXT: LOOP NEXT

POP CX

ENDM

۷. یک MACRO بنویسید که مقادیر ثباتها را ذخیر نماید.

SAVEREG MACRO

PUSH AX

PUSH BX



```

PUSH CX
PUSH DX
PUSH SI
PUSH DI
ENDM

```

### ۸. Interrupt Vector چیست؟

۲۵تا Type Code وجود دارد ریزپردازنده از این استفاده نموده آدرس را در ابتدای حافظه محاسبه نموده و از آدرس دیگری را می خواند. این آدرس جدید Interrupt Vector نام دارد که در حقیقت آدرس برنامه ای است که کار آن وقفه را عینده دار می باشد.

### ۹. در چه قسمتی از حافظه جدول Interrupt Vector قرار دارد؟

در کامپیوترها، ۱۰۲۴ بایت اول حافظه یعنی محلهای حافظه با آدرس ۰ تا 3FF تخصیص به جدولی دارد که این جدول به نام Interrupt vector table معروف است.

### ۱۰. برنامه ای بنویسید که یک پیغام دلخواه را نمایش دهد.

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'
 maxlen db 20
 actlen db ?
 str1 db 20 dup(0)
 n db ' WELCOME TO TPPS!$'           پیغام مورد نظر ;
datasg ends
codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH
LEA DX,MAXLEN

```

```
INT    21H
```

```
LEA    SI,str1
MOV    AL,actlen
MOV    AH,0
ADD    SI,AX
MOV    [SI],'$'
```

```
MOV    AH,09H
LEA    DX,str1
INT    21H
```

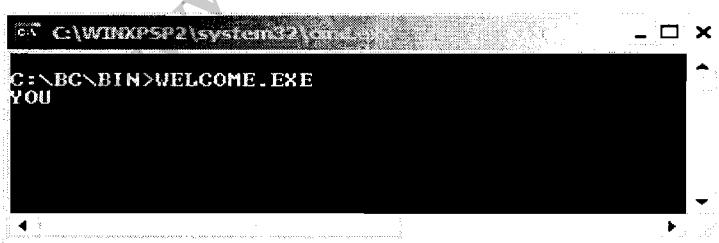
چاپ رشته گرفته شده از ورودی :

```
MOV    AH,09H
LEA    DX,n
INT    21H
```

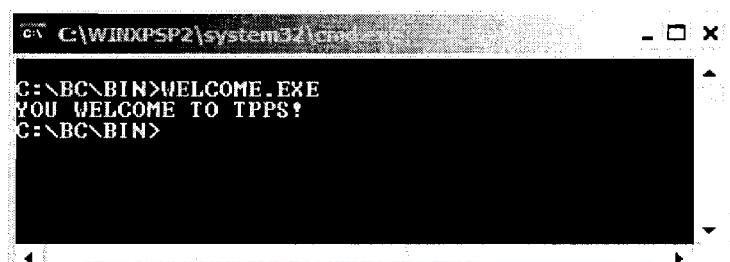
چاپ پیغام دلخواه ;

```
mov   AX,4C00H
INT   21H
main endp
codesg ends
end main
```

ورودی:



خروجی:



در این برنامه یک رشته دلخواه حداکثر به طول ۲۰ از کاربر گرفته و سپس رشته ورودی را با رشته پیش فرض برنامه با هم نشان داده می شود.

۱. روایی بنویسید که دو مقدار صحیح را گرفته مجموع آنها را نمایش دهد.

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'

 maxlen db 20
 actlen db ?
 num db 20 dup(0)
 n1 dw 0
 datasg ends
 codesg segment para 'code'
 main proc far
 assume ss:stacksg,ds:datasg,cs:codesg
 MOV ax,datasg
 MOV ds,ax
```

**CALL SUM ; sum** فراخوانی روال

```
MOV AX,4C00H
INT 21H
main endp
```

**SUM PROC NERA ;** پیاده سازی روال

```
MOV CX,2
L:
MOV AH,0AH
LEA DX,MAXLEN
INT 21H
```

```
MOV AX,0
LEA SI,num
PUSH CX
MOV CL,actlen
MOV CH,0
L1:
MOV BX,10
MUL BX
```

```

MOV  DL, [SI]
SUB  DL, 48
MOV  DH, 0
ADD  AX, DX
INC  SI
LOOP   L1
POP  CX
CMP  CX, 1
JE   E

MOV  n1, AX
DEC  CX

JMP  L

```

E:

```
ADD  AX, n1
```

جمع دو مقدار در این خط انجام می شود ;

```
MOV  BX, 0
```

در قطعه کد زیر مقدار جمع دو عدد چاپ می شود ;

L2 :

```

MOV  DX, 0
MOV  CX, 10
DIV  CX
PUSH AX
PUSH DX
MOV  AX, 10
MUL  BX
MOV  BX, AX
POP  DX
ADD  BX, DX
POP  AX
CMP  AX, 0
JNE  L2

```

```
MOV  AX, BX
```

L3 :

```

MOV  DX, 0
MOV  CX, 10
DIV  CX
PUSH AX
MOV  AH, 02H
ADD  DL, 48
INT  21H

```

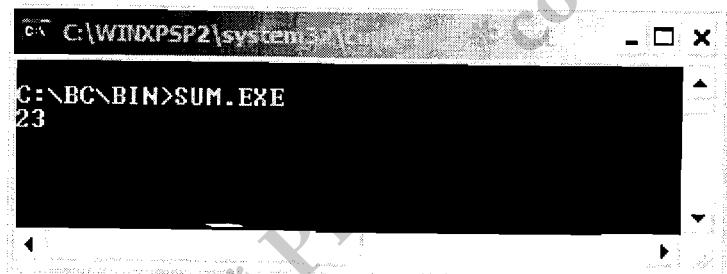
```

    POP  AX
    CMP  AX, 0
JNE   L3
RET
SUM ENDP
codesg ends

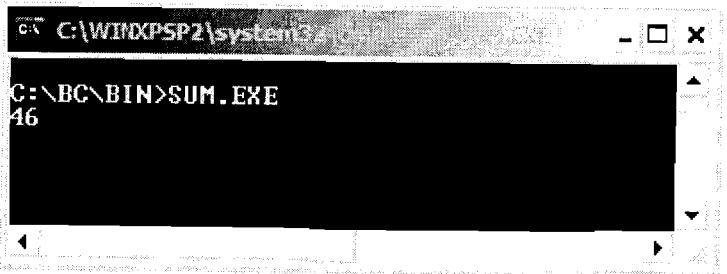
end main

```

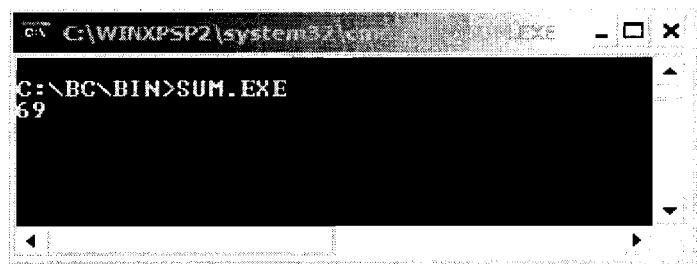
ورودی اول:



ورودی دوم:



خروجی:



در دوال مربوط به جمع دو عدد ابتدا دو عدد از ورودی گرفته (به صورت رشته وسپس به عدد تبدیل می شود) و جمع انجام میشود وسپس آن عدد را چاپ می کنیم (توجه کنید این بار الگوریتم چاپ را در درون این روال نوشته ایم).

۱۲. روایی بنویسید که فاصله زمانی بین فشار دو دکمه را محاسبه نماید.

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'
H DB ?
M DB ?
S DB ?
HS DB ?
datasg ends
codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

CALL PRESS_2KEY

mov AX,4C00H
INT 21H
main endp

PRESS_2KEY      PROC    NEAR
PUSH AX
MOV AH,7

```

INT 21H

MOV AH, 2CH ; گرفتن زمان مربوط به فشار اولین دکمه ;  
INT 21H

MOV H, CH  
MOV M, CL  
MOV S, DH  
MOV HS, DL

MOV AH, 7  
INT 21H

MOV AH, 2CH ; گرفتن زمان مربوط به فشار دومین دکمه ;  
INT 21H

SUB DL, HS  
JNC SUB\_S  
ADD DL, 100  
DEC DH

SUB\_S: SUB DH, S  
JNC SUB\_M  
ADD DH, 60  
DEC CL

SUB\_M: SUB CL, M  
JNC SUB\_H  
ADD CL, 60  
DEC CH

SUB\_H: SUB CH, H  
JNC DONE  
ADD CH, 24

DONE:

RET  
PRESS\_2KEY ENDP

codesg ends

end main

در این روال ابتدا زمان فشار دکمه اول را گرفته، در متغیرهای مربوطه ذخیره می کنیم و بعد از فشار دومین دکمه، زمان را دوباره گرفته، سپس اختلاف دو زمان را محاسبه می کنیم.

۱۳. روالی بنویسید که صفحه مانیتور را پاک نموده و مکان نما را در سطر ۱۰، ستون ۴۰ قرار داده آنگاه کارکتر \* را نمایش دهد.

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'

datasg ends
codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

CALL DISPLAY_CH

mov AX,4C00H
INT 21H
main endp

DISPLAY_CH PROC NAER
MOV AH,00
MOV AL,03
INT 10H

MOV AH,2
MOV BH,0
MOV DH,10
MOV DL,40
INT 10H

MOV AH,2
MOV DL,'*'
INT 21H

```

```
RET
DISPLAY_CH      ENDP
```

```
codesg ends
```

```
end main
```

برای تعیین مکان، مکان نما مقدار سطر را در ثبات DH و مقدار ستون را در ثبات DL قرار می دهیم، با وقفه مربوطه این عمل انجام شده و برای نمایش کاراکتر مورد نظر، آن را در ثبات DL قرار می دهیم.

**۱۴. بیک Macro بنویسید که مقدار N از نوع بایت را گرفته مجموع زیر را محاسبه نماید.**

$$1+2+3+\dots+N$$

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'
 maxlen db 20
 actlen db ?
num db 20 dup(0)
N DB 0
SUM DW 0
datasg ends

codesg segment para 'code'
SUM_N MACRO    N,SUM    ; این ماکرو جمع دو مقدار را محاسبه می کند ;
IFNB <N>
PUSH CX
SUB CX,CX
MOV CL,N
MOV SUM,0
M:
ADD SUM,CX
LOOP M
ENDIF
ENDM

main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax
```

```

MOV AH ,0AH
LEA DX ,MAXLEN
INT 21H

MOV AX,0
LEA SI,num
MOV CL,actlen
MOV CH,0
L1:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP L1
MOV N,AL      ; عدد گرفته شده در متغیر مربوطه ذخیره می شود;

SUM_N N,SUM      ; استفاده از ماکروی مربوطه;

MOV AX,SUM      ; چاپ حاصل عبارت از این قسمت تا پایان کد;
MOV BX,0

L2:
    MOV DX,0
    MOV CX,10
    DIV CX
    PUSH AX
    PUSH DX
    MOV AX,10
    MUL BX
    MOV BX,AX
    POP DX
    ADD BX,DX
    POP AX
    CMP AX,0
JNE L2

MOV AX,BX

```

L3 :

```

MOV DX,0
MOV CX,10

```

```

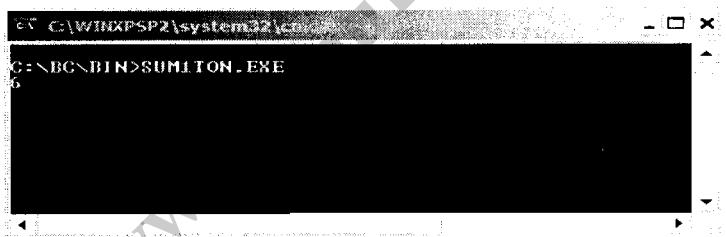
DIV CX
PUSH AX
MOV AH, 02H
ADD DL, 48
INT 21H
POP AX
CMP AX, 0
JNE L3

mov AX, 4C00H
INT 21H

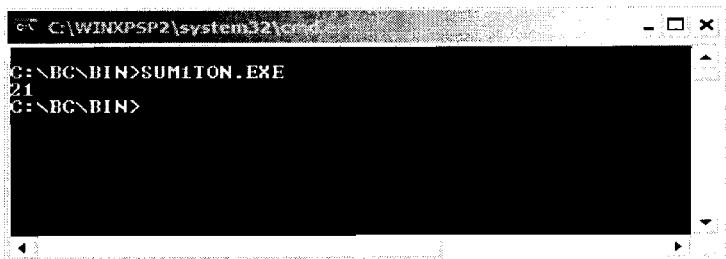
main endp
codesg ends
end main

```

ورودی:



خروجی:



برای محاسبه این مجموع به ماکرو SUM\_N دو پارامتر ارسال شده، که یکی مقدار N و دیگری SUM که مجموع این حاصل جمع را ذخیره کند. در این برنامه با استفاده از یک حلقه مجموع 1 تا N را محاسبه می کند.

۱۵. یک procedure بنویسید که وقت را به صورت یک عدد شش رقمی گرفته مشخص نماید که پس از گذشت ۵ ساعت و ۵۰ دقیقه و ۵۰ ثانیه وقت چیست و آن را نمایش دهد.

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
```

```
datasg segment para 'data'
maxlen db 7
actlen db ?
STRG DB 7 DUP(0)
NUM DB 4 DUP(0)
```

```
H DB ?
M DB ?
S DB ?
C DB 0
N DB 0
ERR DB 'NO$'
ACC DB 'YES$'
datasg ends
```

```
codesg segment para 'code'
```

```
PRINT MACRO N ; ماکرو برای چاپ یک بایت ;
LOCAL A
LOCAL K
LOCAL M1
LOCAL M2
PUSH AX
PUSH CX
MOV CX, 0
A:
MOV AX, 0
MOV AL, N
MOV BL, 10
DIV BL
INC CX
MOV N, AL
MOV AL, AH
MOV AH, 0
PUSH AX
```

```

MOV AL,N
CMP AL,0
JNZ A

M1:
POP AX
    MOV DL,AL
    MOV AH,02H
    ADD DL,48
    INT 21H
LOOP M1

POP CX
POP AX
ENDM

main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH      گرفتن عدد شش رقمی ;
LEA DX,MAXLEN
INT 21H

LEA SI,STRG
MOV AL,actlen
MOV AH,0
ADD SI,AX
MOV [SI],'$'

PUSH DS
POP ES

CLD           جدا سازی دو رقم,دو رقم از عدد گرفته شده ;
MOV CX,1
LEA SI,STRG
LEA DI,NUM

REP MOVSW
MOV . [DI],'$'

MOV AX,0      تبدیل رشته عددگرفته شده به مقدار عددی معادل آن ;
LEA SI,NUM

```

```

MOV CL,2
MOV CH,0
L1:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP    L1

```

MOV H,AL                  عدد گرفته شده در متغیر مربوطه ذخیره می شود ;

CLD                          تبدیل دو کاراکتر بعدی به عدد معادل آن :

```

MOV CX,1
LEA SI,STRG
ADD SI,2
LEA DI,NUM

```

```

REP MOVSW
MOV [DI],'$'
MOV AX,0
LEA SI,NUM
MOV CL,2
MOV CH,0
L2:

```

```

    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP    L2

```

MOV M,AL                  عدد گرفته شده در متغیر مربوطه ذخیره می شود ;

```

CLD
MOV CX,1
LEA SI,STRG
ADD SI,4

```

بدهست آوردن دو رقم آخر ورودی، گرفته شده :

```

LEA     DI,NUM

REP     MOVSW
MOV     [DI],'$'

MOV AX,0
LEA SI,NUM
MOV CL,2
MOV CH,0
L3:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP   L3
MOV S,AL      عدد گرفته شده در متغیر مربوطه ذخیره می شود;

ADD S,50      قطعه کد برنامه از این قسمت شروع می شود;
CMP S,60
JB MIN

MOV AL,S
SUB AL,60
MOV S,AL
INC M

MIN:
ADD M,55

CMP M,60
JB HOUR

MOV AL,M
SUB AL,60
MOV M,AL
INC H

HOUR:
ADD H,5
CMP H,24
JB DONE

```

```
MOV AL, H
SUB AL, 24
MOV H, AL
```

DONE:

```
MOV AL, H
MOV N, AL
PRINT N
```

```
MOV AH, 02H
MOV DL, ':'
INT 21H
```

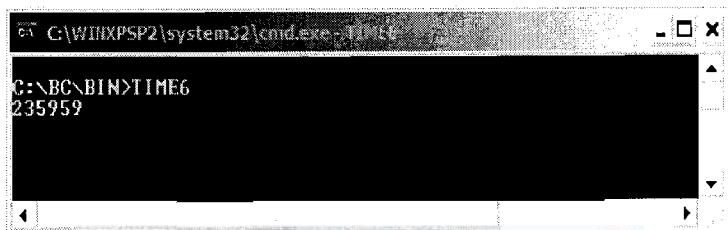
```
MOV AL, M
MOV N, AL
PRINT N
```

```
MOV AH, 02H
MOV DL, ':'
INT 21H
```

```
MOV AL, S
MOV N, AL
PRINT N
MOV AX, 4C00H
INT 21H
main endp
codesg ends
```

end main

ورودی:



خروجی:

```
C:\WINXPSP2\system32\cmd.exe
C:\>TIME6
5:55:49
C:\>
```

ابتدا عدد شش رقمی را به صورت یک رشته دریافت کرده و بعد در سه مرحله این رشته را به سه قسمت دو تایی تبدیل و همزمان عدد مربوطه را بدست آورده و ذخیره می کنیم.

بعد زمان گرفته شده را با ۵ ساعت و ۵۵ دقیقه و ۵۰ ثانیه جمع کرده نتیجه را چاپ می کنیم.

۱۶. یک procedure بنویسید که آرایه N عنصر X از نوع بایت را به صورت نزولی مرتب نماید.

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'
N db طول آرایه
X db آرایه از نوع بایت
datasg ends
codesg segment para 'code'

main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

Call SORT_X

mov AX,4C00H
INT 21H
main endp

SORT_X PROC NEAR
PUSH CX
```

```

PUSH  DI
PUSH  BX
MOV   CH, 0
MOV   CL, N
DEC   CX

LOOP1:
    MOV      DI, CX
    MOV      BX, 0

LOOP2:
    MOV      AL, X [BX]
    CMP      AL, X [BX+1]
    JGE      CONTINUE
    XCHG   AL, X [BX+1]
    MOV      X [BX] , AL

CONTINUE:
    INC      BX
    LOOP    LOOP2

    MOV      CX, DI
    LOOP    LOOP1
    POP      BX
    POP      DI
    POP      CX

RET
SORT_X    ENDP
codeseg ends
end main

```

این برنامه در واقع مرتب سازی حبابی به صورت نزولی است، در این روش، باید چندین مرتبه در طول آرایه حرکت کرد و هر بار، عنصری را با عنصر بعدی مقایسه نمود و در صورتی که عنصر اول از عنصر دوم کوچکتر باشد، جای آنها را عوض کرد (برای مرتب سازی صعودی).

#### ۱۷. یک Macro بنویسید که مینیمم مقدار از نوع word را مشخص نماید.

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'
 maxlen db 20
 actlen db ?
num db 20 dup(0)
N DB 0
MIN DW 1

```

```

datasg ends
codesg segment para 'code'
MIN_N MACRO N,MIN           ;ماکرو برای محاسبه مینیمم
LOCAL L
IFNB <N>
SUB CX,CX
MOV CL,N
M:
MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

MOV AX,0
LEA SI,num

PUSH CX

MOV CL,actlen
MOV CH,0
L:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP    L

POP CX
CMP AX,MIN
JNB CONTINUE
MOV MIN,AX
CONTINUE:
    LOOP    M

ENDIF
ENDM

main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

```

```

MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

```

```

MOV AX,0
LEA SI,num
MOV CL,actlen
MOV CH,0
L1:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP L1

```

عدد گرفته شده در متغیر مربوطه ذخیره می شود ;

```

MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

```

```

MOV AX,0
LEA SI,num

```

PUSH CX

```

MOV CL,actlen
MOV CH,0

```

L:

```

    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI

```

LOOP L

MOV MIN ,AX

```
DEC N

MIN_N    N,MIN

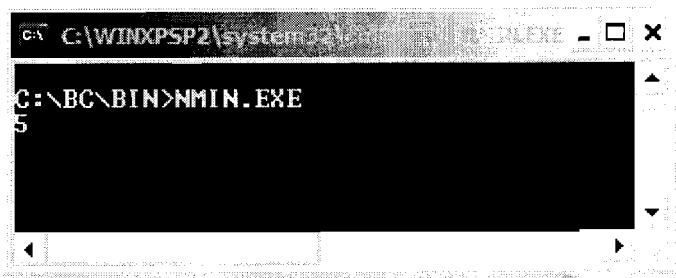
MOV AX,MIN
MOV BX,0

L2:
    MOV DX,0
    MOV CX,10
    DIV CX
    PUSH AX
    PUSH DX
    MOV AX,10
    MUL BX
    MOV BX,AX
    POP DX
    ADD BX,DX
    POP AX
    CMP AX,0
JNE L2

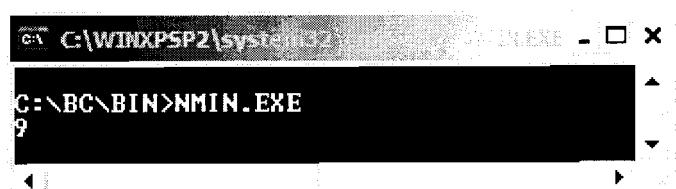
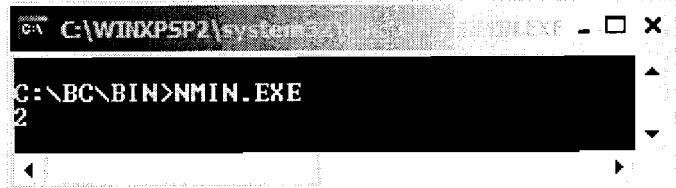
MOV AX,BX

L3:
    MOV DX,0
    MOV CX,10
    DIV CX
    PUSH AX
    MOV AH,02H
    ADD DL,48
    INT 21H
    POP AX
    CMP AX,0
JNE L3
mov AX,4C00H
INT 21H
main endp
codesg ends
end main
```

ورودی:



اعداد ورودی برای مقایسه:



```
C:\WINXPSP2\sys> C:\BC\BIN>NMIN.EXE
2
```

خروجی:

```
C:\WINXPSP2\sys> C:\BC\BIN>NMIN.EXE
2
C:\BC\BIN>
```

ابتدا یک عدد از ورودی گرفته تا تعداد اعداد ورودی را تعیین کرده، سپس با استفاده از عدد گرفته شده اعداد را گرفته و مقایسه کرده و مینیم آنها را به وسیله ماکرو بدست آورده و آنرا چاپ می کنیم.

**۱۸. یک Macro بنویسید که مشخص نماید عدد صحیح و مثبت N از نوع word اول می باشد یا خیر؟**

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'
maxlen db 20
actlen db ?
num db 20 dup(0)
N DW 0
ERR DB 'NO$'          پیام مبتدی بر خیر بودن جواب ;
ACC DB 'YES$'          پیام مبتدی بر درست بودن جواب ;
datasg ends
codesg segment para 'code'
DIGIT MACRO N          ماکرو برای مشخص نمودن عدد اول ;
PUSH CX
MOV CX,N
CMP CX,1
```

```
JZ      NO
CMP    CX, 2
JZ      YES
SUB    CX , 2
MOV    BL, 2
M:     MOV    AX,N
        DIV    BL
        INC    BL
        CMP    AH, 0
LOOPNE M
JZ      NO
YES:
MOV    AH, 09H
LEA    DX,ACC
INT    21H
JMP    E

NO:
MOV    AH, 09H
LEA    DX,ERR
INT    21H
E:
POP    CX
ENDM

main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV    AH ,0AH
LEA    DX,MAXLEN
INT    21H

MOV    AX,0
LEA    SI,num
MOV    CL,actlen
MOV    CH,0
L1:
    MOV    BX,10
    MUL    BX
    MOV    DL,[SI]
    SUB    DL,48
    MOV    DH,0
    ADD    AX,DX
```

```
INC SI
LOOP    L1
```

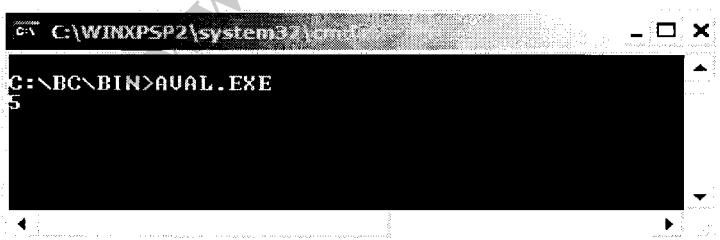
MOV N, AX              عدد گرفته شده در متغیر مربوطه ذخیره می شود ;

**DIGIT N**

```
mov AX, 4C00H
INT 21H
main endp
codesg ends
end main
```

ماکرو **DIGIT** یک پارامتر دارد که مقدار **N** را دریافت می کند برای بررسی اول بودن آن، عددی اول است که به جز خودش و یک به هیچ یک از اعداد دیگر بخش پذیر نباشد. در این ماکرو اگر عدد ورودی اول باشد پیغامی مبتنی بر اول بودن عدد بر روی صفحه نمایش نمایش داده می شود و همچنین اگر عدد اول نباشد پیغامی مبتنی بر اول نبودن عدد نمایش داده می شود.

ورودی:



خروجی:



۱۹. یک Macro بنویسید که دو مقدار N,M از نوع word و مثبت را گرفته کوچکترین مضرب مشترک آنها را محاسبه نماید.

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'```
maxlen db 20
actlen db ?
num db 20 dup(0)
N DW 0
M DW 0
P DW 0
KM db ',KMM:$'
BMM DB 'BMM:$'
datasg ends
codesg segment para 'code'
PRINT MACRO P ;word
LOCAL A
LOCAL K
LOCAL M1
LOCAL M2
PUSH AX
PUSH CX
PUSH BX
MOV CX,0
A:
MOV DX,0
MOV AX,P
MOV BX,10
DIV BX
INC CX
MOV P,AX
MOV AX,DX
MOV DX,0
PUSH AX
MOV AX,P
CMP AX,0
JNZ A
M1:

```

www\*.PauEB\*com



```

POP AX
    MOV DL, AL
    MOV AH, 02H
    ADD DL, 48
    INT 21H

```

```
LOOP M1
```

```

POP BX
POP CX
POP AX
ENDM

```

```
KMM MACRO N, M
```

ماکرو برای محاسبه ک.م.م ;

```
LOCAL L1
```

```
LOCAL L2
```

```
LOCAL L3
```

```
LOCAL E
```

```
MOV AX, N
```

```
CMP AX, M
```

```
JAE L1
```

```
MOV BX, M
```

```
MOV M, AX
```

```
MOV N, BX
```

```
L1:
```

```
    MOV DX, 0
```

```
    MOV AX, N
```

```
    MOV BX, M
```

```
L2:
```

```
    MOV DX, 0
```

```
    DIV BX
```

```
    CMP DX, 0
```

```
    JE L3
```

```
    MOV AX, BX
```

```
    MOV BX, DX
```

```
    JMP L2
```

```
L3:
```

```
    MOV AH, 09H
```

```
    LEA DX, BMM
```

```
    INT 21H
```

```
    MOV P, BX
```

```
    PRINT P
```

```
    MOV DX, 0
```

```

MOV AX,N
DIV BX
MUL M

PUSH AX
MOV AH,09H
LEA DX,KM
INT 21H
POP AX

MOV P,AX
PRINT P
E:
ENDM

```

```

main proc far           ; برنامه اصلی
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV CX,2
L:
MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

MOV AX,0
LEA SI,num
PUSH CX
MOV CL,actlen
MOV CH,0
L1:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP L1
POP CX
CMP CX,1
JE E
MOV N,AX

```

عدد گرفته شده در متغیر مربوطه ذخیره می شود :



DEC CX

JMP L

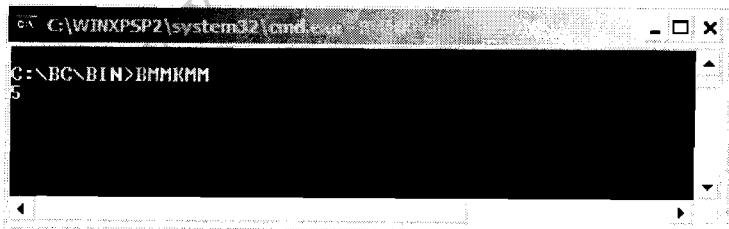
E:

**MOV M, AX** عدد گرفته شده در متغیر مربوطه ذخیره می شود ;

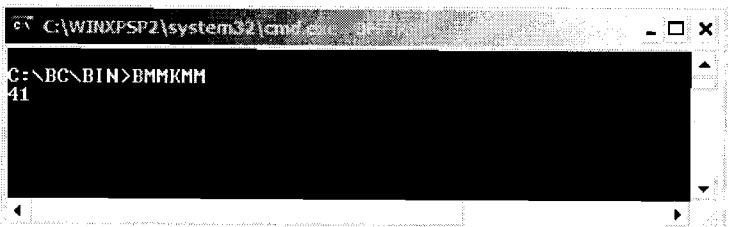
**KMM N, M** استفاده از ماکرو مربوطه;

```
mov AX,4C00H  
INT 21H  
main endp  
codesg ends  
end main
```

ورودی اول:



ورودی دوم:



خروجی:

```
C:\>BC\BIN>BMMKMM
BMM:1, KMM:205
C:\>BC\BIN>
```

برای بدست آوردن کوچکترین مضرب مشترک بین دو عدد  $M$ ,  $N$  را به عنوان پارامتر گرفته، اول بزرگترین مقسوم عليه مشترک این دو عدد را بدست آورده و با داشتن بزرگترین مقسوم علیه مشترک، کوچکترین مضرب مشترک را محاسبه می کند.

۱.۲۰. اگر  $N$  یک عدد صحیح و مثبت و مجدور کامل باشد یک Macro بدهید که جذر آن را محاسبه نماید.

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
datasg segment para 'data'
maxlen db 20
actlen db ?
num db 20 dup(0)
N DW 0
SQUARE DB 0
datasg ends
codesg segment para 'code'
SQR MACRO N, SQUARE      ;ماکرو برای محاسبه جذر عدد
PUSH    BX
PUSH    AX
MOV     BX,1
M:
      MOV     AL,BL
      MUL    BL
      CMP    AX,N
```

```

        JE      EXIT
        INC    BL
        JMP    M

EXIT:
MOV     SQUARE, BL
POP    AX
POP    BX
ENDM

main proc far
assume ss:stacksg,ds:datasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

MOV AX,0
LEA SI,num
MOV CL,actlen
MOV CH,0
L:
    MOV BX,10
    MUL BX
    MOV DL,[SI]
    SUB DL,48
    MOV DH,0
    ADD AX,DX
    INC SI
LOOP    L

MOV N,AX      ; عدد گرفته شده در متغیر مربوطه ذخیره می شود
SQR N,SQUARE

MOV AH,00
MOV AL,03
INT 10H
; چاپ مقدار جذر عدد از اینجا تا انتها برنامه صورت می گیرد
MOV AX,0
MOV AL,SQUARE
MOV BX,0

L1:
    MOV DX,0

```



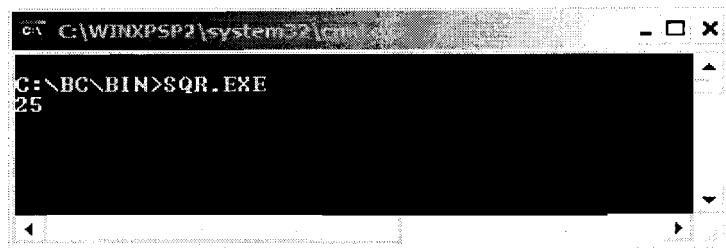
```

MOV CX, 10
DIV CX
PUSH AX
PUSH DX
MOV AX, 10
MUL BX
MOV BX, AX
POP DX
ADD BX, DX
POP AX
CMP AX, 0
JNE L1

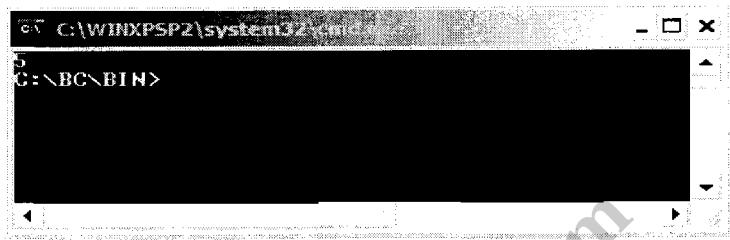
MOV AX, BX
L2 :
    MOV DX, 0
    MOV CX, 10
    DIV CX
    PUSH AX
    MOV AH, 02H
    ADD DL, 48
    INT 21H
    POP AX
    CMP AX, 0
JNE L2
mov AX, 4C00H
INT 21H
main endp
codesg ends
end main

```

وروجی:



خروجی:



www\*.PnuEB\*com

تمرین:

۱. پیشوند REP در مورد کدام دستور العمل های رشته ای کاربرد دارد؟

STOS, MOVS, LODS

۲. پیشوند REPZ در مورد کدام دستور العمل های رشته ای کاربرد دارد؟

CMPS, SCAS

۳. دستور العمل های SCAS و STOS روی کدام فلگها اثر دارند؟

	OF	DF	IF	TF	SF	ZF	AF	PF	CF
STOS	-	-	-	-	-	-	-	-	-
SCAS	*	-	-	-	*	*	*	*	*

۴. برنامه ای بنویسید که یک رشته ۱۰۰ کاراکتری با استفاده از الگوی -----  
----- زیر ایجاد نماید (راهنمایی ۲۰ مرتبه الگو را کپی نمایید)

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends
```

```
datasg segment para 'data'
PATTERN DB '-----'
DISPAREA DB 100 DUP(0)
datasg ends
```

```
etrasg segment para 'extra'
etrasg ends
```

```
codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,es:etrasg,cs:codesg
```

```

MOV ax,datasg
MOV ds,ax
PUSH DS
POP ES

CLD
MOV CX,50
LEA SI,PATTERN
LEA DI,DISPAREA

REP MOVSW
MOV [DI],'$'

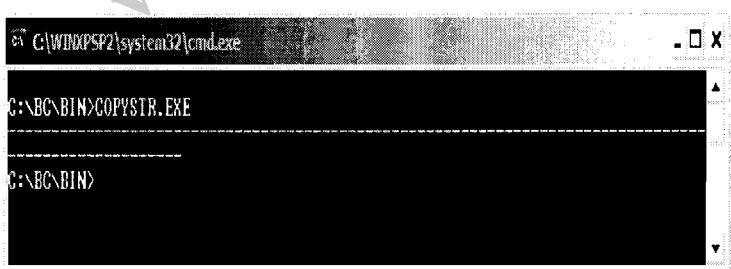
MOV AH,09H
LEA DX,DISPAREA
INT 21H

mov AX,4C00H
INT 21H
main endp
codesg ends

end main

```

خروجی:



رشته ۱۰۰ کاراکتری STRG را در نظر گرفته کلیه کاراکترهای blank آن را به \* تبدیل نمایید.

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends

datasg segment para 'data'
maxlen db 20
actlen db ?
STRG DB 100 DUP(0)
datasg ends

etrasg segment para 'extra'
etrasg ends

codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,es:etrasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

LEA SI,STRG
MOV AL,actlen
MOV AH,0
ADD SI,AX
MOV [SI],'$'

PUSH DS
POP ES

CLD
MOV CX,AX
MOV AL,20H
LEA DI,STRG
L:
REPNE SCASB
JNZ NOTFOUND

```

```

DEC    DI
MOV    BYTE PTR[DI], '*'
INC    DI
JMP    L
NOTFOUND:

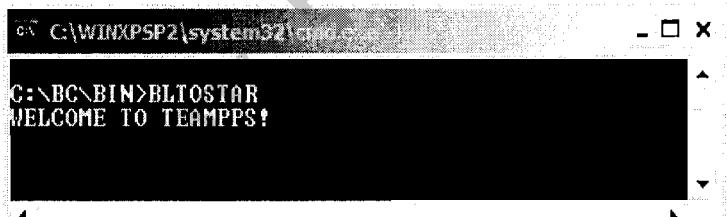
MOV    AH, 09H
LEA    DX, STRG
INT    21H

mov  AX, 4C00H
INT  21H
main endp
codesg ends

end main

```

ورودی:



خروجی:



۶. رشته ۱۰۰ کاراکتری STRG را در نظر گرفته کلیه کاراکترهای آن را به \* تبدیل نمایید

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends

datasg segment para 'data'
maxlen db 100
actlen db ?
STRG DB 100 DUP(0)
datasg ends
codeseg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,es:etrasg,cs:codeseg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

LEA SI,STRG
MOV AL,actlen
MOV AH,0
ADD SI,AX
MOV [SI],'$'

PUSH DS
POP ES

MOV CX,100
MOV AL,'*'
MOV DI,OFFSET STRG
CLD
REP STOSB
MOV [DI],'$'

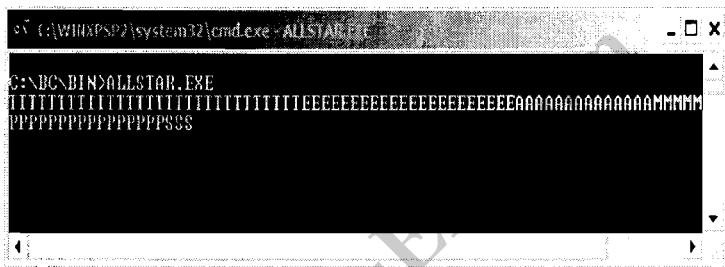
MOV AH,00
MOV AL,03
INT 10H

MOV AH,09H
LEA DX,STRG
INT 21H

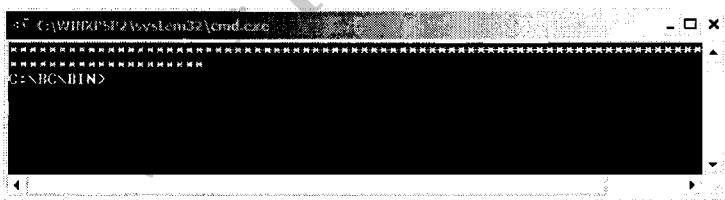
```

```
mov AX, 4C00H  
INT 21H  
main endp  
codesg ends  
end main
```

ورودی:



## خروجی:



۷. رشته ۱۰۰ کاراکتری STRG را در نظر گرفته اولین کاراکتر \* در رشته را به & تبدیل نمایید.

```
stacksg segment para stack 'stack'  
dw 20 dup(0)  
stacksg ends
```

```
datasg segment para 'data'  
 maxlen db 100  
 actlen db ?  
 STRG DB 100 DUP(0)
```

```
datasg ends

etrasg segment para 'extra'

etrasg ends

codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,es:etrasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

LEA SI,STRG
MOV AL,actlen
MOV AH,0
ADD SI,AX
MOV [SI],'$'

PUSH DS
POP ES

CLD
MOV AL,'*'
MOV CX,100
LEA DI,STRG
REPNE SCASB
JNZ NOTFOUND
DEC DI
MOV BYTE PTR[DI], '&'

NOTFOUND:
MOV AH,09H
LEA DX,STRG
INT 21H

mov AX,4C00H
INT 21H
main endp
codesg ends

end main
```

ورودی:

```
C:\BCN\BIN>S6.EXE
MILAD * MOHAMMAD
```

خروجی:

```
C:\BCN\BIN>S6.EXE
MILAD & MOHAMMAD
C:\BCN\BIN>
```

۱۰۰ کاراکتری STRG1 را در نظر گرفته ۲۰ کاراکتر وسط رشته را به رشته STRG2 منتقل نمایید.

```
stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends

datasg segment para 'data'
maxlen db 100
actlen db ?
STRG1 db 100 dup(0)
STRG2 DB 20 DUP(0)
datasg ends

codesg segment para 'code'
main proc far
assume ss:stacksg,ds:datasg,es:etrasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH
```

```

LEA    DX, MAXLEN
INT    21H

LEA    SI, STRG1
MOV    AL, actlen
MOV    AH, 0
ADD    SI, AX
MOV    [SI], '$'

PUSH DS
POP ES

LEA    SI, STRG1
LEA    DI, STRG2
MOV    CX, 40
LOOP1:
    INC    SI
    LOOP   LOOP1
MOV    CX, 20
REP    MOVSB
MOV    [DI], '$'

MOV    AH, 00
MOV    AL, 03
INT    10H

MOV    AH, 09H
LEA    DX, STRG2
INT    21H

mov    AX, 4C00H
INT    21H
main endp
codesg ends
end main

```

ورودی:



خروجی:



۹. رشته ۵۰ کاراکتری STRG1 و رشته ۱۰۰ کاراکتری STRG2 را در نظر بگیرد مشخص نمایید که آیا ۵۰ کاراکتر آخر رشته STRG2 معادل STRG1 میباشد یا خیر؟

```
datasg segment para 'data'
STRG2 DB 100 DUP(0)
```

```
datasg ends
```

```
etrasg segment para 'extra'
STRG1 db 50 dup(0)
etrasg ends
```

```
LEA      SI,STRG2
LEA      DI,STRG1
MOV      CX,50
STD
REPE    CMPSB
JZ      SAME
.
.
.
SAME:   ...
```

۱۰. رشته ۵۰ کاراکتری STRG را در نظر بگیرید کلیه عناصر\* در آن را حذف نمایید. (کاراکتر بعدی را جایگزین \* نمایید)

```

stacksg segment para stack 'stack'
dw 20 dup(0)
stacksg ends

datasg segment para 'data'
 maxlen db 50
 actlen db ?
 STRG DB 50 DUP(0)
datasg ends

codesg segment para 'code'

main proc far
assume ss:stacksg,ds:datasg,es:etrasg,cs:codesg
MOV ax,datasg
MOV ds,ax

MOV AH ,0AH
LEA DX,MAXLEN
INT 21H

LEA SI,STRG
MOV AL,actlen
MOV AH,0
ADD SI,AX
MOV [SI],'$'

MOV AH,00
MOV AL,03
INT 10H

PUSH DS
POP ES

CLD
MOV CX,50
MOV AL,'*'
LEA DI,STRG

L:

```



```

        REPNE      SCASB
JNZ       NOTFOUND
MOV       DL, [DI]
        DEC       DI
MOV       BYTE PTR[DI], DL
INC       DI
JMP       L

NOTFOUND:

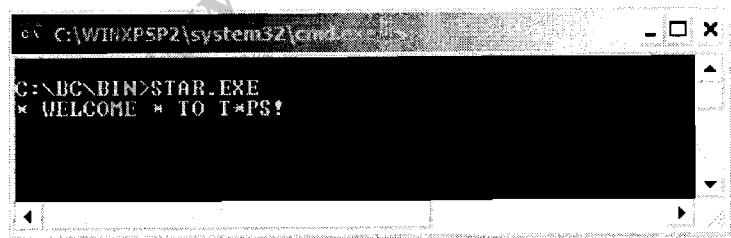
MOV       AH, 09H
LEA       DX, STRG
INT       21H

mov     AX, 4C00H
INT     21H
main    endp
codesg ends

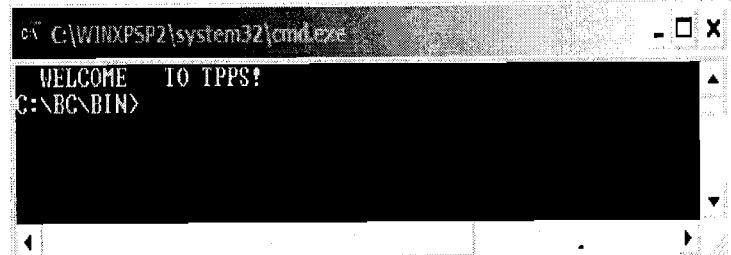
end main

```

ورودی:



خروجی:



# سولانچ چهل گزینه ای

www.RuEB\*com



A vertical column of 25 checkmarks (✓) on a grid background. The checkmarks are arranged in a staggered pattern, starting at the top left and ending at the bottom right. The grid consists of horizontal and vertical lines that intersect to form a series of small squares across the page.