

تمرین اول زبان اسمبلی

ساخت جمع کننده ۱۰ بایتی

نیکی نزاکتی

98522094

ابتدا طبق دستور العمل سوال دو آرایه از نوع unsigned_int 16 به نام های ARR1 و ARR2 و به اندازه ۱۰ بایت

تعریف میکنیم و به آن ها مقدار میدهیم:

```
void TForm2::Sum(void)
{
    BYTE ARR1[] = {0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01},
    ARR2[] = {0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01}; //unsigned_bit 16
```

برای نمایش دادن این دو مقدار در خروجی استرینگ comment را تعریف کرده و با استفاده از Memo مقادیر را در کنسول نمایش میدهیم:

```
UnicodeString comment=L"" ;
Memo1->Lines[0].Text = L"Numbers:" ;
comment.cat_sprintf(L"NUM1: %xH, %xH, %xH, %xH, %xH, %xH, %xH, %xH, %xH, %xH",
    ARR1[0], ARR1[1], ARR1[2], ARR1[3], ARR1[4], ARR1[5], ARR1[6], ARR1[7], ARR1[8], ARR1[9]) ;
Memo1->Lines->Append(comment) ;
comment = UnicodeString().sprintf(L"NUM2: %xH, %xH, %xH, %xH, %xH, %xH, %xH, %xH, %xH, %xH",
    ARR2[0], ARR2[1], ARR2[2], ARR2[3], ARR2[4], ARR2[5], ARR2[6], ARR2[7], ARR2[8], ARR2[9]) ;
Memo1->Lines->Append(comment) ;
```

در قسمت asm به کد زدن با زبان اسمبلی میپردازیم.

در ابتدا از esi و edi به عنوان pointer برای آرایه ها استفاده میکنیم و esi را بر ARR1 لود کرده و edi را برای ARR2. سپس مقدار esi و edi را 9+ میکنیم تا به قسمت MSB آرایه ها اشاره کنند. در مرحله بعد رجیستر ax را صفر کرده و همچنین از ecx برای شمارنده حلقه استفاده میکنیم که چون اندازه آرایه ها ۱۰ است، مقدار ecx را برابر مقدار ۹ هگز (000000009H) قرار میدهیم. برای شروع جمع کردن اول MSB ARR1 را داخل ah میریزیم و سپس MSB ARR2 را به آن add میکنیم و این مقدار ah را در MSB ARR2 قرار میدهیم.(طبق خواسته سوال پاسخ را باید در آرایه دوم بریزیم). در ادامه وارد حلقه L1 میشویم. در حلقه ابتدا از esi و edi یک واحد کم میکنیم که به خانه بعدی آرایه ها اشاره کنند و سپس خانه اشاره شده از ARR1 را در al قرار میدهیم، خانه اشاره شده از ARR2 را به مقدار al اضافه میکنیم و این مقدار al را در خانه اشاره شده ARR2 قرار میدهیم.

```

asm{
    lea esi, ARR1      ; load esi as the pointer to ARR1
    add esi, 9          ; increase esi to point to MSB of ARR1
    lea edi, ARR2      ; load edi as the pointer to ARR2
    add edi, 9          ; increase edi to point to MSB of ARR2
    xor ax, ax          ; set ax as zero
    mov ecx, 00000009H ; set ecx as 9H as loop counter
    mov ah, [esi]       ; move MSB of ARR1 to ah
    add ah, [edi]       ; add MSB of ARR2 to ah
    mov [edi], ah       ; move ah to MSB of ARR2
L1:
    dec esi            ; lower esi to point to next element of ARR1
    dec edi            ; lower edi to point to next element of ARR2
    mov al, [esi]       ; move ARR1 element into ah
    adc al, [edi]       ; add ARR2 element to ah
    mov [edi], al       ; move ah to ARR2 element
    ;dec cx
    ;jnz L1
    loopnz L1
} ;

```

و در آخر نتیجه جمع را روی کنسول نمایش میدهیم:

```

Memo1->Lines->Append(L"Sum:");
comment = UnicodeString().sprintf(L"%xH, %xH, %xH, %xH, %xH, %xH, %xH, %xH, %xH, %xH",
    ARR2[0], ARR2[1], ARR2[2], ARR2[3], ARR2[4], ARR2[5], ARR2[6], ARR2[7], ARR2[8], ARR2[9]);
Memo1->Lines->Append(comment);

```

نمایش خروجی:

