

1000h adresinden itibaren yerleştirilmiş olan 32 adet veriyi kontrol ederek;

- Tek olanların 30h adresinden ve çift olanları 50h adresinden itibaren kopyalayan
- Tek olanların sayısını 71h ve tek olanların toplamını 72h adresine yazan,
- Çift olanların sayısını 73h ve çift olanların toplamını da 74h adresine yazan,
- Tek olanların sayısı, çift olanların sayısından büyükse 75h adresine, F0h, küçükse 0Fh ya da eşitse FFh yazan,

Programı 8051 komutlarıyla tasarlayınız. (Toplama işleminde taşma olmadığı kabul edilecektir.)

```
MOV R1, #30h      //Tek sayıların kopyalanacağı adres
MOV R0, #50h      //Çift sayıların kopyalanacağı adres
MOV DPTR, #1000h  //Verilen başladığı bellek ve 32 adet var
MOV R3, 32h       //32 tane veri var döngü için değişken
MOV 71h, #00h     //Tek sayıların toplam adedi 71'e yazılacak
MOV 72h, #00h     //Tek sayıların toplamı 72'ye yazılacak
MOV 73h, #00h     //Çift sayıların toplam adedi 73'e yazılacak
MOV 74h, #00h     //Çift sayıların toplamı 74'e yazılacak
MOV 75h, #00h     //Eşit sayıdalarsa #FF, tek fazlaysa #F0 çift #0F
```

TEKRAR:

```
MOVX A,@DPTR      //Bellekteki veri A'ya atanır
JB A,0,TEK        //A'da ki sayının 0. biti 0 değilse Tek sayıdır TEK'e dallanır.
INC 73h           //Çift sayıların adedi 1 arttırılır
MOV @R0,A         //Çift sayı kopyalanacağı ilk adrese yazılır
INC R0            //Diğer gelen sayı sonraki adrese kopyalansın diye arttırılır
ADD A,74h         //Önceki çift sayı toplamı elimizdeki sayıya eklenir
MOV 74h, A        //Elimizdeki toplamı çiftlerin yeni toplamı olarak yazarız.
SJMP KONTROL      //Döngü için KONTROL'e atlar.
```

TEK: //Sayı Tek ise çalışacak alt program
INC 71h //Tek sayıların adedi 1 arttırılır
MOV @R1,A //Tek sayı kopyalanacağı ilk adrese yazılır
INC R1 //Diğer gelen sayı sonraki adrese kopyalansın diye arttırılır
ADD A,72h //Önceki tek sayı toplamı elimizdeki sayıya eklenir
MOV 72h, A //Elimizdeki toplamı teklerin yeni toplamı olarak yazarız.
SJMP KONTROL //Döngü için KONTROL'e atlar.

KONTROL: //Döngünün bitip bitmediğini kontrol eder.
INC DPTR //Bir sonraki veri için adresi bir arttırır
DJNZ R3,TEKRAR //R3 Sıfır olana dek TEKRAR'a dallanır.

MOV A,71h //Tek sayıların adedini A'ya atar
CJNE A,73h, ESITDEGIL //Çift sayılarla tek sayıları karşılaştırır. Çıkartma yaparak kontrol eder. 0' sa devam edeer.
MOV 75h, #FFh //Eşit ise #FF yazar.
SJMP SON //Biter.

ESITDEGIL: //Eşit değilse çalışır.
JC CIFTBUYUK //CJNE'de yapılan çıkarma sonrası elde bayrağı varsa çift sayı daha çok demektir.
MOV 75h, #F0h //Tekler çok olunca #F0h yazılır.
SJMP SON //Biter

CIFTBUYUK: //Çift sayılar çok çalışır.
MOV 75h, #0Fh //Çiftler çok olunca #0Fh yazılır.
SJMP SON //Biter

SON:
END //Program Sonu