


```

int main ()
{
    int myArray1[2][3] = {{1, 2, 3}, {0, -6, 7}};
    int myArray2[3][2];

    int index1, index2;

    myTranspose(myArray1, 2, 3, myArray2);

    printf("MATRIS TRANSPOZU:\n");

    for(index1 = 0; index1 < 3; index1++){
        for(index2 = 0; index2 < 2; index2++){
            printf("%d\t", myArray2[index1][index2]);
        }

        printf("\n");
    }

    return 0;
}

```

-----DOĞRU ÇIKTI:-----

```

MATRIS TRANSPOZU:
1      0
2      -6
3      7

```


SORU 4

________________________________\\

myCircularLinkedList() fonksiyonuna ait prototip aşağıdaki gibidir:

```

void myCircularLinkedList(struct listEntry *param1, struct
listEntry *param2, struct listEntry *param3);

```

myLinkedList() fonksiyonu, "param1" 'in işaret ettiği m1 üyesinde 1 değerini, "param2" 'in işaret ettiği m1 üyesinde 2 değerini ve "param3" 'ün işaret ettiği m1 üyesinde 3 değerini sakladıktan sonra bu yapılar arasında bir dairesel bağlı liste (circular linked list) oluşturur ("param1" 'in işaret ettiği yapı, "param2" 'nin işaret ettiği yapıyı; "param2" 'in işaret ettiği yapı, "param3" 'ün işaret ettiği yapıyı; "param3" 'ün işaret ettiği yapı ise "param1" 'in işaret ettiği yapıyı işaret eder).

_____________________________\\

::::::::::: ÖRNEK :::::::::::::
-----main4.c DOSYASININ İÇERİĞİ:-----
#include <stdio.h>

```

struct listEntry {
    int m1;
    struct listEntry *m2;
};

```

```

void myCircularLinkedList(struct listEntry *param1, struct listEntry
*param2, struct listEntry *param3);

```

```

int main ()
{
    struct listEntry var1, var2, var3;

```

```
myCircularLinkedList(&var1, &var2, &var3);

printf("%d\t%d\t%d\t%d", var1.m1, (var1.m2)->m1, ((var1.m2)->m2)->m1,
((var1.m2)->m2)->m2)->m1);

return 0;
}
-----DOĞRU ÇIKTI:-----
1      2 3      1
```