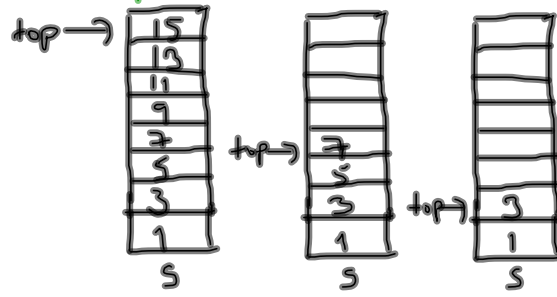


Vize Soruları ve Çözümleri

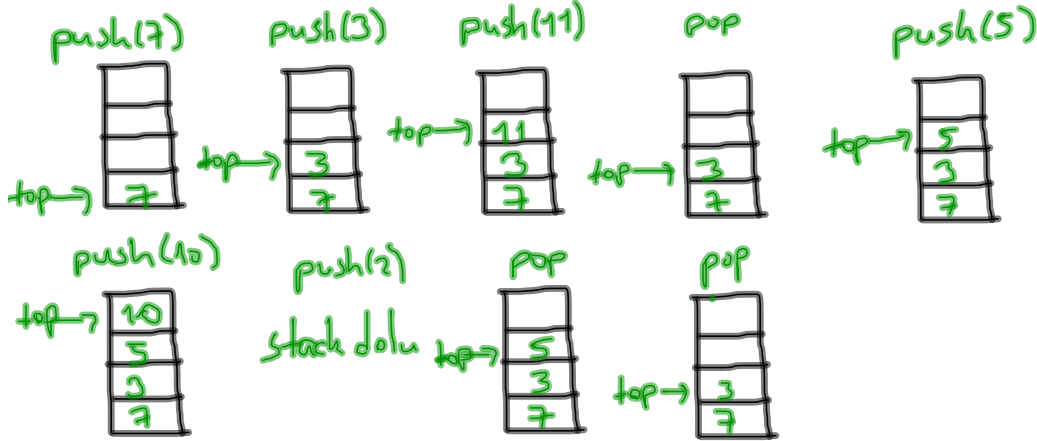
1-) (20 p.) Aşağıdaki kodun çıktısı nedir? Her for döngüsünden sonra stack i çiziniz.

```
int values[8] = {1, 3, 5, 7, 9, 11, 13, 15};
stack s;
initialize(&s);
for (int i = 0; i < 8; i++)
    push (&s, values[ i ]);
int n = 25;
for (int i = 0; i < 4; i++)
    n += pop(&s);
for (int i = 0; i < 2; i++)
    n -= pop(&s);
printf("%d", n);
```



$$n = 25 + 15 + 13 + 11 + 9 - 7 - 5 = 61$$

2-) (20 p.) Aşağıdaki her işlem için dizi kullanılarak gerçekleştirilen stack veri yapısını çiziniz: push(7), push(3), push(11), pop, push(5), push(10), push(2), pop, pop. Stack in boyutunu (Stack_Size) 4 olarak kabul edin. Her işlemde Top (stak in üstü) göstermeyi unutmayın.



3-) (20 p.) Bir dairesel bağlı listenin başındaki ve sonunda düğümlerinin değerlerini toplayan ve sonucu listenin son elemanı olarak ekleyen bir fonksiyon yazınız. Listede en az iki eleman olduğunu kabul edebilirsiniz. Daha önce yazılan fonksiyonları kullanmayın.



```
struct node * circularadd(struct node * head){
    struct node * temp = head;
    while (temp->next != head)
        temp = temp->next;
    struct node * temp2 = new node();
    temp2->data = head->data + temp->data;
    temp->next = temp2;
    temp2->next = head;
    return head;
}
```

4-) (20 p.) Verilen bir bağlı listedeki en küçük değeri tüm düğümlerin (kendisi dahil) değerlerinden çıkaran **subtractMin** adında bir fonksiyon yazınız. Listenin aşağıdaki gibi olduğunu varsayalım:

^{head}
[8, 4, 7, 2, 9, 4, 5, 3] ^{temp}

subtractMin(head) fonksiyonu çağırıldığında, liste aşağıdaki gibi olacaktır.

[6, 2, 5, 0, 7, 2, 3, 1]

```

struct node * subtractMin (struct node *head) {
    struct node * temp = head;
    int min = 99999;
    while (temp != NULL) {
        if (temp->data < min)
            min = temp->data;
        temp = temp->next;
    }
    temp = head;
    while (temp != NULL) {
        temp->data -= min;
        temp = temp->next;
    }
    return head;
}

```

5-) (20 p.) Verilen bir q1 kuyrugundaki tek sayı içeren elemanları yeni bir q2 kuyruğuna, çift sayı içeren kuyruk elemanlarını ise yeni bir q3 kuyruğuna ekleyen bir program yazınız. enqueue, dequeue gibi kuyruk fonksiyonlarını kullanabilirsiniz.

```

int x;
queue q2, q3;
initialize (&q2);
initialize (&q3);
while (!isEmpty (&q1)) {
    x = dequeue (&q1);
    if (x % 2 == 0)
        enqueue (&q3, x);
    else
        enqueue (&q2, x);
}

```