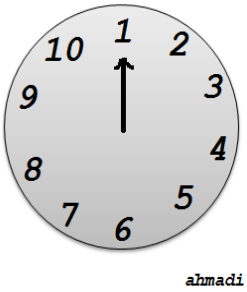
مسئله ژوزفوس:

این یک مسئله قدیمی و گونه ای از مسئله باستانی منتسب به فلاویوس ژوزفوس، تاریخدان قرن اول میلادی است. او در زمان جنگ رومیان با یهودیان،یکی از 41 یهودی‌ای بود که به وسیله رومیان در یک غار محاصره شده بودند.به جای تسلیم،این گروه از جنگجویان تصمیم گرفتند که همگی به این صورت خود کشی کنند:آنها قرار گذاشتند تا با شروع از نفر اول،و به صورت حلقوی،هر بار نفر دوم زنده ها خود را بکشند و نوبت به نفر زنده بعدی برسد، تا اینکه هیچ کس باقی نماند. ولی ژوزفوس زرنگ‌تر از آنها بود و مکان نشستن آخرین فردی را که قاعدتا باید خود را به تنهایی می کشت محاسبه کرد و از ابتدا در آن مکان نشست و جان سالم به در برد.

مسئله در حالت کلی بدین صورت است: اگر n نفر با شماره های 1 تا n دور دایره ای قرار بگیرند و با شروع از شماره 1 و در جهت ساعتگرد هر بار دومین نفر زنده (یا kامین نفر در حالت کلی) خودش را بکشد،آخرین نفر چه شماره ای دارد؟مثلا مطابق شکل،برای n=10 به ترتیب افراد

9 , 1 , 7 , 3 , 10 , 8 , 6 , 4 , 2

خودکشی می کنند و 5 زنده می ماند.



جواب این مسئله یا J(n) ، از رابطه بازگشتی زیر قابل محاسبه است:

$$J(1) = 1$$

$$J(2n) = 2J(n)-1 \quad , \text{for } n>= 1 ,$$

$$J(2n+1) = 2J(n)+1 \quad \text{for } n>=1;$$

اگر n به صورت یک عدد دودویی بنویسیم و آن را به چپ یک بیت شیفت دورانی دهیم J(n) به دست می آید. مثلا برای

$$n = 100 = (110010)_2$$,

جواب

$$J(n) = (1001001)_2 = 73$$

است.

ما در اینجا مسئله را با استفاده از یک لیست پیوندی حلقه ای پیاده سازی می کنیم و جواب را با شبیه سازی خودکشی ها به دست می آوریم. روشن است که این راه حل کند، برای حالات دیگر این مسئله نیز قابل استفاده است.

و حالا راه حل مسئله ژوزفوس(شبه کد):

```
p ← First(L)
while next[p] ǂ p
        do Delete-After(L,p)
                p← next[p]
Print element[p]
```

کدنویسی برنامه به زبان C++:

```
int Josephous(Node *head ) {
Node *p=head;
while(p->next != p){
        remove(p->next);
        p = p->next;
}
cout<< p->data;
}
```