

۳. [۱۵] آرایه ای به طول ۵۱۱ داریم . میخواهیم با استفاده از binary search عدد ۱۱۰ را در این آرایه جستجو کنیم . اگر این عدد در آرایه موجود نباشید چه تعداد مقایسه در اجرا الگوریتم رخ داده است ؟ کدام ایندکس های آرایه مورد مقایسه قرار گرفتند ؟ در صورتی که الگوریتم اجرایی linear search باشد تعداد مقایسه ها چه تفاوتی می کند ؟

.....

۴. [۲۰] درستی یا نادرستی عبارات زیر را با علامت (✓) یا (X) مشخص کنید. دلیل خود را در نقطه چین زیر هر عبارت توضیح دهید.

(a) ___ The following array is a max heap: [10, 3, 5, 1, 4, 2]

.....

(b) ___ In max-heaps, the operations insert, find-max, and find min all take $O(\log n)$ time.

.....

(c) ___ stack is the most suitable data structures if you only need to implement recursion in a programming language

.....

(d) ___ tree is the most suitable data structure if you need to store the directory structure on your hard drive

.....

۵. [۱۰] چگونه با استفاده از یک *priopityQueue* یک *stack* پیاده سازی کنیم؟

.....

۶. [۱۵] توضیح دهید قطعه کد روبرو چه کاری انجام میدهد .

```
void function()
Node * current ← head
Node * pre ← Null
Node * next ← Null
while current ≠ NULL do
    next ← current->next
    current->next ← pre
    pre ← current
    current ← next
end
head ← pre
```

سپس الگوریتم را بر روی *LinkedList* مقابل به صورت مرحله به مرحله اجرا کنید و خروجی را نمایش دهید .

1- > 2- > 3- > null

.....

.....

.....

.....

.....

.....

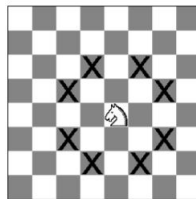
.....

.....

.....

.....

۷. [۱۵] یک اسب را در صفحه شطرنج $n * n$ در نظر بگیرید . خانه هایی که اسب میتواند به آن ها برود در تصویر زیر قابل مشاهده است . یک الگوریتم برای محاسبه تعداد راه هایی که در آن می توانید با $k \geq 0$ حرکت از خانه (i_s, j_s) به (i_t, j_t) برود طراحی کنید . همچنین پیچیدگی محاسباتی و پیچیدگی حافظه ای آنرا به دست بیاورید .



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....