



مبانی کامپیوتر و برنامه‌نویسی

استاد : دکتر مرادی

تمرین سری ۶ - لیست پیوندی و

نیمسال دوم ۹۸-۹۹

گیت‌های پایه

موعد تحویل: ۳۰ اردیبهشت

نوشتاری:

۱. درستی یا نادرستی عبارات زیر را با ذکر دلیل بیان کنید.

(آ) عملیات insert در جای مشخصی از آرایه ساده‌تر از همان عملیات در لیست پیوندی است.

(ب) عملیات search در آرایه دلخواه سخت‌تر از یک لیست پیوندی است.

(پ) عملیات search در آرایه مرتب شده آسان‌تر از یک لیست پیوندی است.

(ت) عملیات merge دو آرایه ساده‌تر از دو لیست پیوندی است.

۲. اگر به تابع زیر لیست پیوندی $\{1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 3 \rightarrow 9\}$ را که در آن ۱ سرلیست

است بدهیم، با آن چه می‌کند؟ توضیح دهید و لیست را پس از فرآیندی که در تابع برای آن رخ داده‌است، چاپ کنید.

توضیح دهید آیا کد زیر موجب memory leak می‌شود یا خیر؟

```
struct List {
    int data;
    struct List* next;
};

void f(struct List* list) {
    struct List* curr = list;
    struct List* temp;
    struct List* l;
    curr = curr->next;
    curr->data = 3;
    list->next = curr->next;
    temp = curr->next->next;
    curr->next->next = curr;
    curr->next = temp;
    curr = curr->next;
    curr->next->next = curr->next->next->next;
    curr = list->next;
    curr->data = 9;
    curr = curr->next->next->next->next;
    if (curr->data == 2)
        curr->next = curr->next->next;
    l = (struct List*)malloc(sizeof(struct List));
    l->data = 0;
```

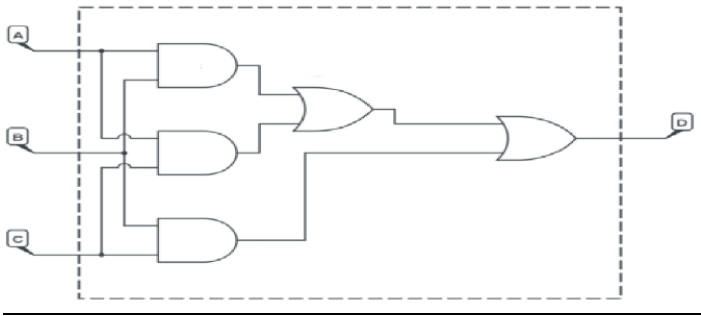
```
l->next = curr->next->next;
curr->next->next = l;
}
```

۳. جدول درستی زیر را در نظر بگیرید. پیاده‌سازی مناسبی از گیت‌های پایه (and و not و or) برای آن بیابید.

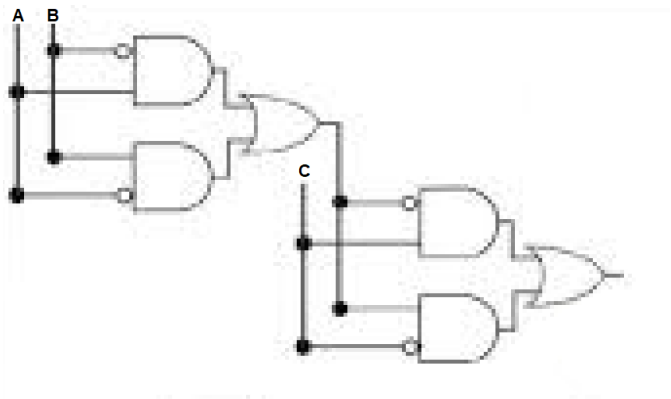
جدول شماره 1

<i>X</i>	<i>Y</i>	<i>Output</i>
<i>F</i>	<i>F</i>	<i>F</i>
<i>T</i>	<i>F</i>	<i>T</i>
<i>F</i>	<i>T</i>	<i>T</i>
<i>T</i>	<i>T</i>	<i>F</i>

۴. جدول درستی مدارات زیر را رسم کنید.



مدار شماره 1



مدار شماره 2

۵. حال می خواهیم به کمک مدار شماره ۱ و شماره ۲ یک *Full Adder* طراحی کنیم. به این صورت که A و B و C_{in} ورودی و S و C_{out} خروجی‌های سیستم هستند و میدانیم جدول درستی زیر برای این متغیرها برقرار است. برای اطلاعات بیشتر در مورد *Full Adder* به [https://en.wikipedia.org/wiki/Adder_\(electronics\)](https://en.wikipedia.org/wiki/Adder_(electronics)) مراجعه کنید.

جدول درستی *Full Adder*

A	B	C_{in}	S	C_{out}
F	F	F	F	F
F	F	T	T	F
F	T	F	T	F
F	T	T	F	T
T	F	F	T	F
T	F	T	F	T
T	T	F	F	T
T	T	T	T	T

کامپیوتری:

۶. تابعی بنویسید که آرایه‌ای از اعداد به همراه طول آن را بگیرد و با آنها لیست پیوندی یک طرفه‌ای ایجاد کرده و آن را بازگرداند.

Input: $n = 5$, $arr = [3, 10, 1, 5, 4]$

Output: $List = \{3 \rightarrow 10 \rightarrow 1 \rightarrow 5 \rightarrow 4\}$

```
List* create_linked_list(int n, int* arr)
```

۷. حال به گونه‌ای تابع سوال قبل را تغییر دهید که اعداد به صورت مرتب شده در لیست پیوندی قرار گیرند.

Input: $n = 5$, $arr = [3, 10, 1, 5, 4]$

Output: $List = \{1 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 10\}$

```
List* create_sorted_linked_list(int n, int* arr)
```

۸. تابعی بنویسید که دو لیست پیوندی مرتب شده را دریافت کرده و بدون ساختن لیست جدید، لیست پیوندی مرتب حاصل از ادغام این دو لیست را برگرداند. (یعنی تنها اشاره گرها را باید تغییر دهید.)

Input: list1 = {1 → 3 → 4 → 5 → 10}, list2 = {2 → 4 → 8}

Output: {1 → 2 → 3 → 4 → 5 → 8 → 10} or {1 → 2 → 3 → 4 → 4 → 5 → 8 → 10}

```
List* merge(List* list1, List* list2)
```

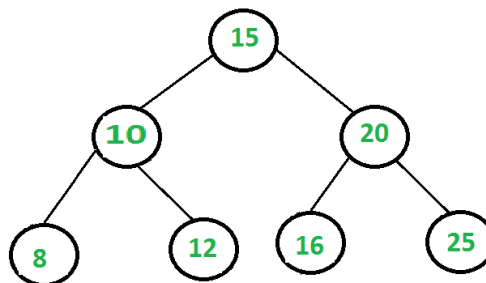
۹. تابعی بنویسید که یک لیست پیوندی مرتب شده را بگیرد و عناصر تکراری آن را حذف کند.

Input: list = {1 → 3 → 3 → 10 → 10}

Output: {1 → 3 → 10}

```
List* remove_repetition(List* list)
```

۱۰. یکی از ساختارهای داده درخت دودویی^۱ است. درخت دودویی به این صورت است که یک گره به صورت ریشه وجود دارد و این گره دارای ۲ فرزند می باشد؛ یکی فرزند راست و دیگری فرزند چپ و برای فرزندان نیز هر یک به همین صورت دو فرزند چپ و راست وجود دارد و این فرزندان تا زمانی که به یک برگ برسند، ادامه می یابد. داده ی موجود در فرزند راست از داده ی پدر بیشتر و داده ی فرزند چپ از داده ی پدر کمتر است. به شکل زیر توجه کنید.



برای اطلاعات بیشتر در مورد درخت های دودویی، می توانید به لینک زیر مراجعه کنید:

https://en.wikipedia.org/wiki/Binary_tree

¹ Binary tree

در این سوال قرار است که شما از روی یک آرایه‌ی مرتب‌شده، یک درخت دودویی بسازید و گره‌ی ریشه را بازگردانید. به این صورت که عنصر وسط آرایه که میانه‌ی آرایه نام دارد را به عنوان ریشه در نظر می‌گیریم. سپس آرایه به دو آرایه با طول نصف آرایه‌ی اصلی به وجود می‌آید. حال باید عنصر وسط آرایه‌ی سمت چپ را به عنوان فرزند چپ ریشه و عنصر وسط آرایه‌ی سمت راست را به عنوان فرزند سمت راست ریشه در نظر بگیرد و این کار را برای فرزندان نیز انجام دهد. برای مثال، آرایه‌ی مرتب متناظر با درخت دودویی بالا به صورت [۲۵, ۲۰, ۱۶, ۱۵, ۱۲, ۱۰, ۸] است. (سوال امتیازی)

```
struct TNode
{
    int data;
    struct TNode* left;
    struct TNode* right;
};
```

توضیحات:

۱. تمرینها باید تا پایان مهلت تعیین شده در سایت درس آپلود شوند و امکان تحویل تمرین تا یک روز بعد با تاخیر وجود دارد. ددلاین هر دو بخش نوشتاری و کامپیوتری یکسان است.
۲. هدف این تمرین یادگیری شماسست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.
۳. برای این تمرین دو مکان آپلود در نظر گرفته شده است. فایل آپلودی بخش نوشتاری باید فرمت PDF داشته باشد و فایل آپلودی بخش کامپیوتری باید فرمت zip باشد. نام هر دو بخش باید به صورت HW6_StudentID باشد.
۴. تمیز بودن کد شما اهمیت ویژه‌ای دارد. عدم رعایت فاصله از سر خط در کدنویسی و نامگذاری نامناسب متغیرها و توابع نمره‌ی منفی دارد.
۵. به فرمت خروجی (عبارت چاپ شده) خواسته شده در هر سوال توجه کنید. در صورتی که از فرمت گفته شده استفاده نکرده باشید نمره‌ی آن را نخواهید گرفت. دقت کنید که به غیر از مواردی که در صورت سوال یا شکل به آن‌ها اشاره شده مورد دیگری را چاپ نکنید.
۶. مجاز به استفاده از دستور goto و متغیرهای global نیستید. در صورت استفاده در هر سوال، نمره‌ی آن سوال صفر در نظر گرفته می‌شود.
۷. هر سوالی درباره این تمرین داشتید، به ایمیل hesam.77s@gmail.com پیام دهید.
۸. در صورت وجود هرگونه مشکل در تمارین و یا آپلود آنها با ovaheb@gmail.com در ارتباط باشید.

موفق باشید