



# طراحی زبان‌های برنامه‌سازی

نیم‌سال دوم ۱۴۰۰-۱۴۰۱

استاد: محمد ایزدی

گردآورندگان: امیررضا میرزایی، آرین احدی‌نیا

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی کامپیوتر

مهلت ارسال: ۲۹ اسفند

کار با RACKET

تمرین اول

به موارد زیر توجه کنید:

- (۱) برنامه‌های خود را به زبان Racket بنویسید.
- (۲) مهلت ارسال تمرین ساعت ۵۹ : ۲۳ روز ۲۹ اسفند ۱۴۰۰ است.
- (۳) در مسائلی که نیاز به خروجی دادن است، پاسخ را چاپ نکنید بلکه آن را به عنوان حاصل تابع برگردانید.
- (۴) حتما نام تابع اصلی خود را در مانند نمونه‌های ورودی بگذارید تا هنگام داوری به مشکل نخورید. منظور از تابع اصلی تابعی است که اجرای آن روی ورودی، خروجی مطلوب را نشان می‌دهد.
- (۵) جواب هر سوال برنامه‌نویسی را در یک فایل Racket با نام و فرمت زیر قرار دهید.

[StudentID]\_[ProblemNumber].rkt

- (۶) در نهایت تمام فایل‌های Racket را در یک فایل زیپ قرار داده و با نام `HW1_StudentID` در سامانه کوثرآپلود کنید.
- (۷) هرگونه سوالی راجع به تمرین را در زیر پست مربوطه در کوثرای درس مطرح کنید.
- (۸) در مجموع تمامی تمرین ۷ روز مهلت تاخیر مجاز دارید و پس از تمام شدن این تاخیرهای مجاز به ازای هر روز ۱۰ درصد از کل نمره تمرین شما کم می‌شود.
- (۹) لطفا تمرین‌ها را از یکدیگر کپی نکنید. در صورت وقوع چنین مواردی مطابق با سیاست درس رفتار می‌شود.

سوالات (۱۰۰ نمره)

۱. (۱۰ نمره) برنامه‌ای بنویسید که یک لیست از اعداد طبیعی غیر تکراری و یک عدد طبیعی ورودی بگیرد و به عنوان خروجی تمام اعضای لیست که مضربی از عدد ورودی هستند را به صورت یک لیست مرتب خروجی بدهد. (می‌توانید از تابع مرتب‌سازی خود رکت استفاده کنید).

نمونه ورودی:

```
(main '(25 2 3 45 43 (23 5 30 (10 35 1))) 5)
```

نمونه خروجی:

```
'(5 10 25 30 35 45)
```

۲. (۱۵ نمره) برنامه‌ای بنویسید که مسیر ورودی و خروجی دریافت کند و کد رکت موجود در فایل ورودی را اجرا کند و خروجی این کد را در فایل خروجی بنویسید.

(راهنمایی: در مورد توابع `file` - `output` - `open` و `read` موجود در Racket تحقیق کنید).

نمونه ورودی:

```
(main "input.txt" "output.txt")
```

۳. (۲۰ نمره) برنامه‌ای بنویسید که یک ماتریس مربعی  $A$  ورودی بگیرد و حاصل  $A^T A$  را حساب کند.

نمونه ورودی:

```
(main '((1 3) (2 2)))
```

نمونه خروجی:

```
'((5 7) (7 13))
```

۴. (۴۰ نمره)

(الف) (۱۵ نمره) تابعی به نام `binary-tree` بنویسید که یک لیست را دریافت و عناصر آن را به ترتیب در یک درخت دودویی جستجو درج کند. رتوس درخت دودویی جستجو را یک لیست در نظر بگیرید که به ترتیب شامل مقدار راس، زیردرخت سمت چپ و زیر درخت سمت راست است. میتوانید لیست خالی را به عنوان درخت خالی در نظر بگیرید. فرض کنید که عناصر درخت تماما عدد هستند.

نمونه ورودی:

```
(binary-tree '(5 1 3 8 7 6 4))
```

نمونه خروجی:

```
'(5 (1 () (3 () (4 () ()))) (8 (7 (6 () ())) ()))
```

(ب) (۵ نمره) تابعی به نام `flatten` بنویسید که یک درخت دودویی جستجو را دریافت و آن را به یک لیست مرتب تبدیل کند.

نمونه ورودی:

```
(flatten '(5 (1 () (3 () (4 () ()))) (8 (7 (6 () ())) ())))
```

نمونه خروجی:

```
'(1 3 4 5 6 7 8)
```

(ج) (۱۰ نمره) تابعی به نام `depth` بنویسید که یک درخت دودویی جستجو و یک عدد را در ورودی دریافت کند و بگوید اولین بار آن عدد در چه عمقی از درخت ظاهر شده است. عمق ریشه را صفر در نظر بگیرید. اگر مقدار مورد نظر در درخت موجود نبود، `ERROR۴۰۴` را نمایش دهید.

نمونه ورودی ۱:

```
(depth 10 '(5 (1 () (3 () (4 () ()))) (8 (7 (6 () ())) ())))
```

نمونه خروجی ۱:

```
'Error404
```

نمونه ورودی ۲:

```
(depth 3 '(5 (1 () (3 () (4 () ()))) (8 (7 (6 () ())) ())))
```

نمونه خروجی ۲:

```
2
```

(د) (۱۰ نمره) در یک درخت جستجوی دودویی، اولین جد مشترک مجموعه `A` را راسی در نظر میگیریم که درختی که ریشه‌اش آن باشد، شامل تمام مقادیر در مجموعه `A` باشد.

تابعی به نام `minimal-tree` بنویسید که ورودی اول آن یک درخت جستجوی دودویی و ورودی دوم آن یک لیست از عناصر تماما متمایز باشد که نمایانگر مجموعه `A` باشد و در جواب مقداری که در اولین جد مشترک قرار گرفته است را بازگرداند.

در این بخش تضمین می‌شود که مقادیر درج شده در درخت تماما متمایز هستند. ورودی نمونه:

```
(minimal-tree '(6 (4 (2 (1 () ())) (3 () ())) (5 () ())) (7 () ())) '(1 3 5))
```

خروجی نمونه:

```
4
```

۵. (۱۵ نمره) برنامه‌ای بنویسید که تمام زیرمجموعه‌های یک مجموعه در قالب لیست را خروجی بدهد. (ترتیب خروجی دادن مهم نیست). نمونه ورودی:

```
1 powerset '(1 2 3 4))
```

نمونه خروجی:

```
1 '((1 2 3 4) (1 2 3) (1 2 4) (1 2) (1 3 4) (1 3) (1 4) (1) (2 3 4) (2 3)
   (2 4) (2) (3 4) (3) (4) ())
```