

۱- تمرین کتاب Mitchell

۵-۱- می توان تابع زیر را در نظر گرفت که در حالت ایستا درست است زیرا تایپ `procedure` در `Algol` همان `proc` است و به پارامترها ربطی ندارد. و چون طبق فرض سوال جمع بولین و عدد صحیح خطای زمان اجرا تولید می کند، این تابع نیز در زمان اجرا به خطا بر می خورد.

```
procedure Q(t); integer x; begin Q = x + t; end;
```

(a-۳-۵)

```
fun f(p) = if #2(p) = 0 then #1(p) else if #1(p) = 0 then #2(p) else #1(p) + #2(p)
```

(b)

خیر، مگر آنکه در شرط اول نیز از $x=y$ استفاده شود.

(c) به صورت کلی باید نام متغیرهای ورودی را تغییر دهیم و تساوی را در هر قسمت چک کنیم.

```
fun eq(x,y) if x = y then true else false
```

(d) چون اگر ورودی تابع تکرار داشت، باید تابعی وجود می داشت تا تساوی آن دو را چک کند. اما طبق صورت سوال تابع تساوی برای توابع تعریف نشده است.

۵-۴-۱) بر اساس ساختار این تابع، اگر ورودی به صورت برگ باشد، f آن محاسبه می شود و اگر نود میانی باشد، تابع روی تر دو فرزندش پیاده می شود.

```
fun maptree(f, leaf(y)) = leaf(f(y)) | maptree(f, node(x, y)) =  
maptree(f, f(x), f(y))
```

(b)

```
('a →' b) →' atree →' btree
```

چون از جنس ورودی و خروجی f خبر ندارد و اعلام نشده است و می تواند هر چیزی باشد.

(a-۶-۵)

```
fun f1(a, b) = c; fun f2(a) = b; fun f3(b) = c
```

```
fun curry(f1) = f3(f2(a))
```

```
fun uncurry(f3(f2(a))) = f1(a, b)
```

(b)

$$unCurry(curry(f1)) = uncurry(f3(f2(a))) = f1(a, b)$$

-۷-۵

(a) در زمان محاسبه ی y ، اگر x رشته باشد، $x.i$ مقدار مشخصی ندارد و جواب خروجی بدون هیچ خطایی مقداری غیر معلوم خواهد بود. (بر اساس آنکه چه عددی از قبل در خانه ای که $x.i$ به آن اشاره می کند وجود داشته است).

(b) این عبارت در ml خطا خواهد بود زیرا ml با تگ کردن نوع داده ی تعریف شده در `union`، کار را پیش می برد و اگر در اینجا m رشته باشد، `TAG_INT` نمی تواند بر آن اپلی شود. پس ادامه ی محاسبه متوقف می شود.