

[דף סיכום בחינה](#)

מזהה סטודנט: N102279113

מזהה קורס: 20905 שם קורס: שפות תכנות

מספר שאלה	ניקוד מירבי	ציון
1	35.00	28.00
2	35.00	29.00
3	30.00	30.00

ציון בחינה סופי : 87.00

הבחינה הבדוקה בעמודים הבאים

N102279113



מספר סידורי: 4 ת.ז.: 208252908

84 20905 מספר תעודת זהות (9 ספרות) מספר הקורס מועד

לשימוש הבודק

קצרה שלה 1:

W

lang.scm =

Parser J toy

(expression {"{" (separated-list identifier ")") "}" } enum-exp)

חסרה המילה enum

(expression {"<" expression "." identifier ">" } enum-elmt-exp)

(expression {"match" "[" identifier ":@" expression "]"

-2

arbno

(separated-list "?" identifier "=>" expression ";")

match-exp)

data-structures.scm =

expval J טו

(identifier-val (val symbol?))

(enum-val (vals (list-of identifier-val)))

(define (expval->enum v)

(cases expval v

(enum-val (vals) vals)

(else (eopl:error "v is not an enum")))



enum J טו
ident J טו
identifier-val J טו

interp.scm =

(enum-exp (ids)

(if (null? ids)

(eopl:error "empty identifier list in enum definition")

(enum-val (map (lambda (e) (identifier-val e)) ids))))

אין בדיקה בהגדרת enum האם השמות שונים זה מזה -3

(enum-elmt-exp (enum id)

(let ((enum (expval->enum (value-of enum env))))

(if (not (is-member? enum (identifier-val id)))

(eopl:error "~s not found" id)

(list-ref enum (index-of-mem enum id))))

חיותר, ניתן פשוט להחזיר את (identifier-val id)

480074

```

(match-exp (enum exp1 enmids exps)
  (let ((enum (expval → enum) (apply-env env enmid)))
    (exp1-val (value-of exp1 env))
    (if (not (is-member? enum exp1-val))
        (eopl:error "ms not found" (expval → id exp1-val))
        (let ((ids (map (λ (i) (identifier-val i)) enmids)))
            (if (not (lists-equal? enum ids))
                (eopl:error "match cases is not correspond to enum ms cases" enmid)
                (value-of (list-ref exps (index-of-mem enum exp1-val) env)))))))

```



"ms"

exp1-val)

110 158

```

(define (is-member? lst elem)
  (if (null? lst)
      #f
      (and (equal? (car lst) elem)
           (is-member? (cdr lst) elem))))

```

or

```

(define (index-of-mem enum id)
  (if (null? enum)
      -1
      (if (equal? (car enum) id)
          0
          (+ 1 (index-of-mem (cdr enum) id)))))

```

0 קניי זמני, קניי זמני, קניי זמני, קניי זמני

```

(define (lists-equal? lst1 lst2)
  (if (null? lst1)
      (null? lst2)
      (and (equal? (car lst1) (car lst2))
           (lists-equal? (cdr lst1) (cdr lst2)))))

```

-2

סדר הופעת השמות
 ב-match לא חייב להיות כמו הסדר שלהם
 בהגדרת ה-enum
 לכן צריך להשוות את הרשימות בצורה שתחשב בכך

data-structures.scm:

```

(define (expval → id v)
  (cases expval v
    (identifier-val (i) i)
    (eopl:error "ms is not an identifier-val" v)))

```

lang.scm:

:2 הנה

```
(expression ("for(proc" "(" identifier "i"
            "[" (separated-list identifier "," ) "]" ";"
            "[" (separated-list expression "," ) "]" ")")
            (arbno "<" Guard ">")
            "{ expression }") forproc-exp)
```



```
(Guard ("skip" ":" expression) skip-guard)
```

```
(Guard ("break" ":" expression) break-guard)
```

~~interp.scm:~~ data-structures.scm:

```
(skip-val (val bool?)) - expval ׀ גויף
```

```
(break-val (val bool?))
```

```
(define (expval → skip v)
```

מחזיר

```
(exp cases expval v
```

```
(skip-val (b) by (expval → bool b))
```

```
(else (eopl:error "is not a skip-val" v)))
```



```
(define (expval → break v)
```

```
(cases expval v
```

```
(break-val (b) by (expval → bool b))
```

```
(else (eopl:error "is not a break-val" v)))
```

interp.scm:

```
(skip-guard (exp)
```

```
(skip-val (expval → bool (value-of exp env))))
```



```
(break-guard (exp)
```

```
(break-val (expval → bool (value-of exp env))))
```

-- הנה פירוט

guard מורכב
מביטוי אך
הוא עצמו
אינו ביטוי
(הוא לא
וריאנט של
expression
לכן זו שגיאה
לעדכן את
expval
בהקשרו

```

(forproc-exp (Id Ids Bodies Guards Body)
  (if (not (equal? (length Ids) (length Bodies)))
    (opl:error "Length of Ids and Bodies
      must be equal")
    (let ((new-env (extend-env Id
      (newref (proc-val (procedure
        (car Ids) (car Bodies)
        env))) env)))


```

```

(forproc-exp (Id Ids Bodies Guards Body)
  (if (not (equal? (length Ids) (length Bodies)))
    (opl:error "Length of Ids and Bodies must be
      equal")
    (begin
      (do-eval Id Ids Bodies Guards Body env)
      (num-val 27))))

```

~~define~~

:value-of Γ פירמ 27/27

```

(define (do-eval Id Ids Bodies Guards Body env)

```

```

  (let* ((new-env (extend-env Id (newref (proc-val
    (procedure (car Ids) (car Bodies) env))) env))
    (guard-res (eval-guards Id Guards new-env)))
    (opl:error
    (if for (null? guard-res) ; #t  $\Gamma$  env  $\Gamma$ 
      (do-eval Id (cdr Ids) (cdr Bodies) Guards Body env)
      (begin
        (value-of Body new-env)
        (do-eval Id (cdr Ids) (cdr Bodies) Guards
          Body env))
      (cases exprval guard-res
        (skip-val (v) (do-eval Id (cdr Ids) (cdr Bodies)
          Guards Body env))
        (break-val (v) (num-val 27))
        (else (num-val 27) ; unreachable))))

```

```

(define (eval-guards Guards env)


```

```

(if (null? Guards)
  (1)
  (let (|yes (value-of (car Guards) env)))
    (cases exprval res
      (skip-val (v) ((if (bool


```


480074

7

480074

480074

גליון תשובות לשאלות רב-בררתיות

הקף במעגל את התשובה שבחרת (לכל שאלה יש רק תשובה אחת נכונה).
 אם תרצה לבטל תשובה שבחרת, סמן עליה X.
 דוגמה לתשובה שבחרת: א ב ג ד ה ז ח ט
 דוגמה לתשובה שבטלת: א ב ג ד ה ז ח ט

שאלה	תשובה
1	א ב ג ד ה ו ז ח ט
2	א ב ג ד ה ו ז ח ט
3	א ב ג ד ה ו ז ח ט
4	א ב ג ד ה ו ז ח ט
5	א ב ג ד ה ו ז ח ט
6	א ב ג ד ה ו ז ח ט
7	א ב ג ד ה ו ז ח ט
8	א ב ג ד ה ו ז ח ט
9	א ב ג ד ה ו ז ח ט
10	א ב ג ד ה ו ז ח ט
11	א ב ג ד ה ו ז ח ט
12	א ב ג ד ה ו ז ח ט
13	א ב ג ד ה ו ז ח ט
14	א ב ג ד ה ו ז ח ט
15	א ב ג ד ה ו ז ח ט
16	א ב ג ד ה ו ז ח ט
17	א ב ג ד ה ו ז ח ט
18	א ב ג ד ה ו ז ח ט
19	א ב ג ד ה ו ז ח ט
20	א ב ג ד ה ו ז ח ט
21	א ב ג ד ה ו ז ח ט
22	א ב ג ד ה ו ז ח ט
23	א ב ג ד ה ו ז ח ט
24	א ב ג ד ה ו ז ח ט
25	א ב ג ד ה ו ז ח ט
26	א ב ג ד ה ו ז ח ט
27	א ב ג ד ה ו ז ח ט
28	א ב ג ד ה ו ז ח ט
29	א ב ג ד ה ו ז ח ט
30	א ב ג ד ה ו ז ח ט
31	א ב ג ד ה ו ז ח ט
32	א ב ג ד ה ו ז ח ט
33	א ב ג ד ה ו ז ח ט
34	א ב ג ד ה ו ז ח ט
35	א ב ג ד ה ו ז ח ט
36	א ב ג ד ה ו ז ח ט
37	א ב ג ד ה ו ז ח ט
38	א ב ג ד ה ו ז ח ט
39	א ב ג ד ה ו ז ח ט
40	א ב ג ד ה ו ז ח ט

לשימוש פנימי

מספר התשובות הנכונות: _____ ציון: _____

שם הבודק: _____

480074