

צ'ודנובסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 15

שאלה 1:

א. העמודה השמאלית מראה את הסדר של רשומות ההפעלה במחסנית הקריאות (*main* בתחתית המחסנית).

<i>main</i>	<i>No access link</i>
<i>g</i>	<i>Access link points to (frame for)main</i>
<i>f</i>	<i>Access link points to main</i>
<i>g</i>	<i>Access link points to main</i>
<i>h</i>	<i>Access link points to g</i>
<i>f</i>	<i>Access link points to main</i>

ב.

```
t = *arp // t points to frame for g
t = *t // t points to frame for main
t[12] = 17 // foo = 17
```

שאלה 2:

```
stmt -> STRANGE_IF M {
    gen ( "JMP" M.cond);
    label (M.st1);
}

stmt1 {
    gen ( "JMP" M.exit);
    label (M.cond);
}

(' expression ' ) {
    gen ( "IASN" M.tmp boolexp.result);
    gen ( "JMPZ" M.st2 M.tmp);
    gen ( "JMP" M.st1);
    label (M.st2);
}

stmt2 { label (M.exit);}

M -> epsilon {
    M.cond = new label();
    M.st1 = new label();
    M.st2 = new label();
    M.exit = new label();
    M.tmp = new temp();
}
```

צ'ודנבסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 15

שאלה 3:

```
void stmt() {
    switch(lookahead){
        case WHILE : ...
        case ID : ...
        case IF : ...
        case STRANGE_IF :

            cond = new label();

            st1 = new label();

            st2 = new label();

            exit = new label();

            tmp = new temp();

            match(STRANGE_IF);

            gen ( "JMP" cond);

            label (st1);

            stmt();

            gen ( "JMP" exit);

            label (cond);

            match('(');

            int result = boolexp ();

            match(')');

            gen ( "IASN" tmp result);

            gen ( "JMPZ" st2 tmp);

            gen ( "JMP" st1);

            label (st2);

            stmt();

            label(exit);

            /* end case STRANGE_IF */
            .....
    } /* end stmt */
}
```

צ'ודנובסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 15

שאלה 4:

.א

```
expr -> expr ADDOP term { gen (ADDOP.op == PLUS ? "add" : "sub" ); }
```

```
expr -> term { }
```

```
term -> term MULOP factor { gen ( MULOP.op == MUL ? "mul" : "div" ); }
```

```
term -> factor { }
```

```
factor -> '(' expr ')' { }
```

```
factor -> ID { gen ( "load" ID.name ); }
```

```
factor -> NUM { gen ( "load" NUM.val ); }
```

ב. מניח שפקודה האחרונה ברגע שמבצעים expr היא store.expr.result ואז מכניסים עוד פעם תוצאה של הביטוי למחסנית ומשווים עם 0. אם קטן או שווה יוצאים מהתוכנה ואם לא עוד פעם מכניסים למחסנית תוצאת הביטוי ומחסירים 1. וחוזרים על זה עד שתוצאת ביטוי קטנה או שווה ל-0.

```
stmt -> REPEAT ( expr ) labels
```

```
{
    label (labels.cond);
    gen ("load" expr.result);
    gen ("load" 0);
    gen ("le");
    gen ("jump_if_zero" labels.exit);
    gen ("load" expr.result);
    gen ("load" 1);
    gen ("sub");
    gen ("store" expr.result); }
stmt1
{ gen ("jump" labels.cond);
  label (labels.exit); }
```

```
labels -> epsilon
```

```
{ labels.cond = newlabel();
  labels.exit = newlabel(); }
```

צ'ודנובסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 15

שאלה 5:

א. **הסבר כללי:** משתנה top מייצגת טבלת סמלים כך שמצביע על ראש הטבלה בשרשרת של טבלאות שניצור בזמן קריאה של התוכנית קלט. כל פעם שנכנסים לבלוק חדש { ... } אנו שומרים מצביע על טבלה הנוכחית בעזרת משתנה לוקלי saved. כלומר ברגע שיוצאים מהבלוק top מקבל את ערך שהיה לפני הכניסה לבלוק.

פעולות על טבלת סמלים:

make_table – פונקציה יוצרת ומחזירה טבלת סמלים ריקה.
insert(list, type) – פונקציה מוסיפה רשימה של משתנים או משתנה אחת אם טיפוס יחיד ומחזירה טבלת סמלים מעדכנת.
lookup(symbol table, variable name, variable type) – פונקציה בוליאנית מחזירה true אם משתנה קיים טבלת סמלים עם טיפוס הדרוש ו- false אם לא.

פונקציות אחרות:

make_list(variable name) – פונקציה יוצרת רשימה עם משתנה אחת מחזירה רשימה
add_list(variable name) – פונקציה מוסיפה לרשימה משתנה חדש ומחזירה רשימה מעודכנת
duplicate(list, variable name) – פונקציה בוליאנית מחזירה true אם יש ברשימת משתנים, משתנים עם אותו שם אחרת מחזירה false

נשתמש גם בתכונות הנבנות הבאות:

S.count, C.count, E.count – ספירת המרות בתוכנה

T.type, E.type – טיפוס של המשתנה

ID.name – שם של המשתנה

N.list – רשימת משתנים

צ'ודנבסקי דמיטרי 324793900 קומפילציה 20364 א 2021

ממ"ן 15

```
1. P → {
    top = null;
    saved = top;
    top = make_table();
    } L S {
        top = saved;
        print("Number of castings %d",S.count);
    }

2. L → L1 D

3. L → epsilon

4. D → N ':' T ';' {top = insert(N.list,T.type)}

5. N → N1 ' ' ID { if (duplicate(N1.list,ID.name)) print("Duplicate variables !!! ");
    else{
        N.list = N1.list;
        N.list = addlist(ID.name);
    }
}

6. N → ID { N.list = makelist(ID.name) }

7. T → FLOAT { T.type = float; }

8. T → INT { T.type = integer; }

9. S → S1 C { S.count = S1.count + C.count; }

10. S → epsilon { S.count = 0; }

11. C → ID '!=' E ';' {
    if(!lookup(top,ID.name,ID.type)) print("Not exist this variable or wrong type in symbol table !!! ");
    else{
        if (ID.type != E.type) { C.count = E.count + 1; }
        else { C.count = E.count; }
    }
}
```

צ'ודנבסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 15

```

12. C → BEGIN {
    saved = top;
    top = make_table();
    } L S END {
    top = saved;
    C.count = S.count;
    }

13. E → E1 '+' ID {
    if(!lookup(top, ID.name, ID.type)) print("Not exist this variable or wrong type in symbol table !!! ");
    else{
        if(E1.type == float && ID.type == float ) {
            E.type = float;
            E.count = 0;
        }

        else if(E1.type == float || ID.type == float ) {
            E.type = float;
            E.count = 1;
        }

        else {
            E.type = integer;
            E.count = 0;
        }
    }
}

14. E → ID {
    if(lookup(top, ID.name, ID.type)) E.type = ID.type; E.count = 0;
    else print("Not exist this variable or wrong type in symbol table !!! ");
}

```

ב. טבלאות הסמלים כפי שיראו בזמן שקומפיילר נמצא בשורה 10:

```

1. x , y: int;
2. x@1 := y@1 ;
3. begin x , w: float ;
4.     x@3 := x@3 + w@3 ;
5. end
6. y@1 := x@1 + y@1 ;
7. begin y , w: int ;
8.     x@1 := w@7 + y@7 ;
9.     begin a: int ;
10.         a@9 := x@1 ;
11.     end
12. end

```

Symbol Table	
x	1
y	1
y	7
w	7
a	9

צ'ודנובסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 15

שאלה 6:

- א. לפי שורות: $100 + 4 * (4 * 30 + 10) = 620$
 לפי עמודות: $100 + 4 * (7 * 10 + 4) = 396$
- ב. $\times \left(\text{array} \left(100, \text{pointer} \left(\text{pointer} \left(\text{record} \left((p \times \text{int}) \times (\text{next} \times \text{pointer}(\text{record})) \right) \right) \right) \right) \right) \times \text{double}) \rightarrow \text{int}$

שאלה 7:

- א. האובייקט B יועבר לרשימה free.

unreached	unscanned	scanned
A B C D E F		
B C D E F	A	
B D E F	A C	
B E F	A C D	
B E F	C D	A
B F	C D E	A
B F	D E	A C
B	D E F	A C
B	E F	A C D
B	F	A C D E
B		A C D E F

- ב.

כתובת	אובייקט
10000	A
לא מועבר ל- To Space כי הוא garbage	B
10100	C
10200	D
10300	E
10400	F