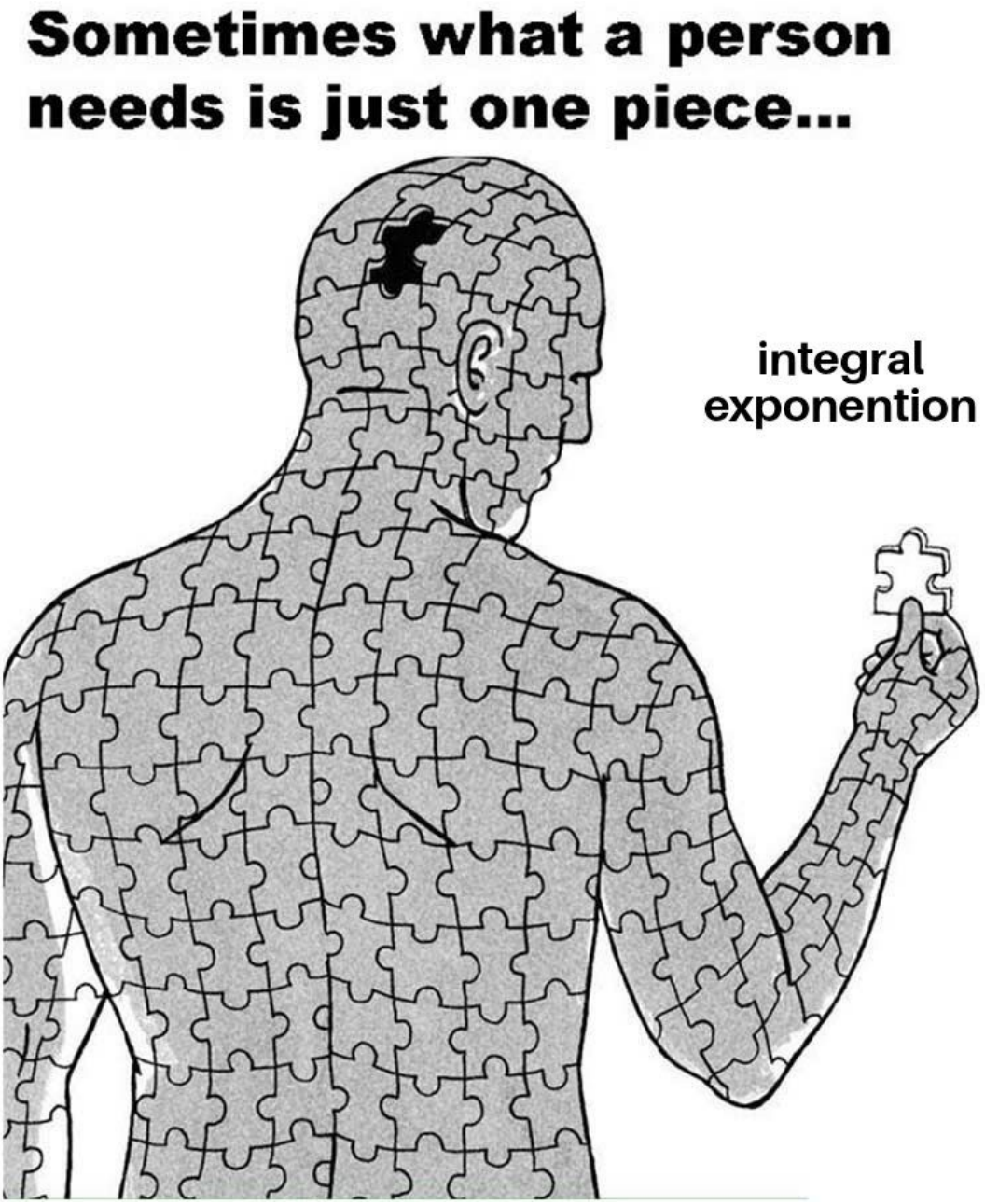
תרגיל בית 4

מגישים : אסף לובטון 209844414 ועדן דמבינסקי 212227888



שאלה 1:

התבוננו בקטע הקוד הבא של שפת תכנות כלשהי וענו על הסעיפים הבאים, בתוספת הסבר קצר.

שימו לב:

- **תשובה ללא הסבר לא תיבדק!**
- הניחו שהקוד אינו מכיל שגיאות סינטקטיות, ושהוא רץ ומסתיים ללא שגיאות על כל קלט אפשרי.
- כמו כן, הניחו שהפונקציה input_string_read מקבלת קלט מהמשתמש ומחזירה ערך מטיפוס string.

```
x = 3;

s = read_string_input();

if (isnumber(s) and strToInt(s) > 0)

     x = s + 1.3;

else

     x = "negative input";

 print(x);
```

<u>א. האם השפה היא typed?</u>

השפה המתוארת היא typed:
ראשית, לפי ההנחה מקבלים מכל שורת קוד input_string_read מחזירה ערך מטיפוס string, יתרה מכך קיימת המרה ממחרוזת למספר (`strToInt`).
בנוסף יש לציין כי בודקים אם הקלט הוא מספר(int) והופכים אותו לכזה, לכן השפה מחלקת לסוגים שונים של ערכים מחלקת אותם לקבוצות וכן גם לטיפוסים ומאפשרת לבצע פעולות על משתנה מסוים לפי הטיפוס אליו הוא שייך זהו מאפיין של מערכת טיפוסים.

<u>ב. האם בשפה יש strong typing או weak typing?</u>

לשפה יש טיפוסיות חלשה מאחר והיא מאפשרת(לפי ההנחה התוכנית מתקבלת ללא שגיאות סינטקטיות) לעשות פעולה חיבור בין טיפוס string לטיפוס int כאשר רושמים x=s+1.3, יתרה מכך בתחילת התוכנית איקס אותחל להכיל את הערך שלוש (כלומר הוא היה מטיפוס int) ולאחר מכן בהמשך התוכנית הייתה השמה ל"negative input" כלומר הוא קיבל ערך מטיפוס string.

<u>ג. האם בשפה יש explicit typing או implicit typing?</u>

בשפה המתוארת יש implicit typing ניתן לראות שבאף שלב בתוכנית המתוארת לא צוין שם של טיפוס, לכן יש להניח כי השפה מסיקה לבד את הטיפוס של כל משתנה.

שאלה 2:

למדו את שפת GO מתוך התיעוד הרשמי שלה:
א. מהם בנאי הטיפוסים בשפה? האם יש שם בנאים שלא מופיעים בקורס?האם יש בנאים תיאורטיים שנלמדו בקורס ולא מופיעים בשפה? מיהם?

בנאי הטיפוסים בשפה:

Method sets-integral exponentiation
Boolean types-Mapping
Numeric types-integral exponentiation
String types- integral exponentiation
Array types-integral exponentiation
Slice types-integral exponentiation
Struct types-Record
Pointer types-integral exponentiation(קבוצה של כל המצביעים לאותו)
Function types-Mapping (קבוצה של כל הפעולות בעלות אותן פרמטרים וערך החזרה)
Interface types-הוא בנאי שלא ראינו מסוגו בקורס
Map types-Mapping
Channel types-הוא בנאי שלא ראינו מסוגו בקורס
type definition -branding

על מנת לממש טיפוס המוגדר רקורסיבית נעשה זאת באופן הבא:

simple recursive-

```
type Category struct {
   id int
   Parent *Category
}
```

בנאים תיאורטיים שלמדנו ולא מופיעים:
power set, product, disjoint union, subrange, multiple recursive.
נוסיף כי בשפת גו לא קיימות מחלקות class.

ב. מהם הטיפוסים האטומיים?

bool byte complex64 complex128 error float32 float64
int int8 int16 int32 int64 rune string
uint uint8 uint16 uint32 uint64 uintptr nil struct map interface

ג. נתח את שפת GO לפי הקריטריונים שנלמדו בקורס לסיווג מערכת טיפוסים.
מערכת טיפוסים: לשפה יש מערכת טיפוסים כפי שראינו בסעיף הראשון.

טיפוסיות חזקה/חלשה: לשפה טיפוסיות חזקה לא ניתן להפר את הקשר בין משתנה לטיפוס.

אורתונולוגיות: השפה היא אורתונולוגית מאחר והבנאים שלה לא מפלים טיפוסים.

חלוקת אחריות: המפרש מסיק בעצמו את הטיפוסים של המשתנים. ההסקה היא כמו בinferref typing- ml.

זמן אכיפת הטיפוסים: לשפה יש טיפוסיות סטטית חוקיות הטיפוסים נבדקת בזמן קומפילציה.

שקילות טפוסים: לשפה שקילות מבנית. לפי התיעוד שני טיפוסים הם שקולים אם"ם יש להם את אותו המבנה.

הסבר: שתי פונקציות הן זהות אם הן מחזירות ומקבלות את אותו מספר פרמטרים ו(שטיפוסייהם זהה), שני מערכים שקולים רק אם יש להם את אותו האורך והאיברים מאותו הטיפוס. שני סלייסים זהים אם יש להם את אותם הטיפוסים. שני מצביעים הם זהים אם מסוג אותו טיפוס פרימיטבי, שני ממשקים זהים אם הם מכילים את אותה קבוצת מתודות בעלות שמות זהים וסוגי טיפוסים זהים, שני מילונים זהים אם יש להם את אותם המפתחות ואלמנטים מאותו הסוג, שני ערוצים זהים אם הם מאותם הטיפוסים ובעלי אותו כיוון.

רמת תחכום: השפה היא מרמת תחכום 4 מאחר ויש לה מערכת טיפוסים רקורסיבית (רמה 4), על כן היא עונה על על רמת התחכום הרביעית שהיא: -recursive type system לדוגמה ניתן להגדיר בשפה:

```
type Category struct {
    id int
    Parent *Category
}
```

יתרה מכך פונקציות הן first class values.

גמישות: השפה גמישה שכן קיימים סוגים רבים של טיפוסים והמשתמש מקבל חופש רב בכתיבת הביטויים.