תכנון קוד – קומפי 5:

1. קוד להקצאת רגיסטרים (בדומה ל-freshVar מההרצאה(.

הסבר: פונקציה שמגרילה שמות עם אינדקס רץ. לפני שם של משתנה צריך להופיע %. שם של משתנה מהצורה %bla3.

1. חישובים לביטויים אריתמטיים. התחילו מחישובים פשוטים והתקדמו לחישובים מורכבים יותר. בדקו אותם בעזרת הדפסות.

הסבר:

1. נוסיף ל-EXP תכונה var
2. בגזירה של חישוב אריתמטי :

* נקצה משתנה לביטוי משמאל (שלב 1)
* נדפיס את השורה המתאימה לפי כלל הגזירה

1. נעשה את שלב ב גם לביטויים מהצורה NUM, NUM B, נעשה חיבור עם 0.
2. חלוקה ב-0 – להוסיף תנאי if עם בדיקת הארגומנט השני, כדי שההדפסה תקרא בזמן ריצה. כמובן שיש פה בקפצ'ון קטן כי למה לא.
   * לעדכן קריאות לפונקציות + שימוש במחרוזת
3. INT הוא 32 ביט להשלים את byte עם zext.
4. גלישה מטווח של BYTE – לשמור עליו בטווח 0-255.
5. פעולה עם מספר קבוע – צריך להדפיס את המספר!
6. לוודא udiv, sdiv – אם אחד מהם int אז signed ואם שניהם byte אז unsigned?
7. טיפול ב-set:

* להפוך את TYPE לstruct שהשדות שלו הם הenum types ושני שדות מספריים שיחזיקו את ההתחלה והסוף של הסט (במידה וזה סט..). לעדכן בהתאמה את setcheck.
* לשנות שימוש בenum TYPES לשימוש בstruct.
* לסדר חיבור וחיסור של משהו מספרי ושל set.
* לבדוק האם יש משמעות לסדר בין המספר לסט.
* טיפול ב-in
* איך לייצג את bool? בינתיים התייחסנו לזה בתור i32 של 0 או 1.
* המרה מ-set ל- int – לעבור על כל הסט?

1. לבדוק קבצים ב-MOBA
2. חישובים לביטויים בוליאניים מורכבים . בדקו אותם בעזרת הדפסות.

הסבר:

1. לא ניתן לשמור ביטויים בוליאנים ברגיסטרים! צריך לממש את זה עם קפיצות וbackpatching . צריך לעשות את זה ל-OR, NOT, AND, לוודא שגם שרשור שלהם מביא לנו את התוצאה הדרושה.
2. להבין איך משתמשים בפונקציית phi.
3. שמירת וקריאת משתנים במחסנית.

הסבר:

1. שינוי הoffset כך שיכיל את הגודל האמיתי של המשתנה, נתייחס לbyte ולbool כi32 ונשתמש בפקודה zext כדי להשלים אפסים.
2. שינוי offsets של set ?
3. בכניסה לפונקציה צריך להקצות מקום לכל המשתנים הלוקליים על המחסנית באמצעות הפקודה alloca - מספר משתנים לוקליים בכל פונקציה – קטן מ-50 –
   1. מחסנית משתנים לוקליים - %locals
4. רצף של statements.

הסבר:

1. אתחול משתנים
   1. אתחול של סט –
   2. השמה של ערך –
2. גישה למשתנים
3. וידוא הדפסה לכל statement
4. מבני בקרה.

הסבר:

1. If
2. While
3. Break
4. continue
5. קריאה לפונקציות הפלט.

הסבר:

1. Print, printi – הפונקציות שהם הגדירו לנו להוסיף לקוד
2. קריאה לפונקציות .

הסבר:

1. קריאה לפונקציה
2. Return
   1. ווידוא שתמיד יש ret מתאים בסוף
   2. החזרה של set באופן דיפולטי- פיאצה
   3. Ret Exp
   4. החזרה של set בעזרת return Exp
3. כתיבת מסגרת – הצהרה + סוגר בסוף
4. הקצאת מקום לארגומנטים והעתקה שלהם
5. טיפול בשגיאות
6. קלט ופלט המנתח
7. מחרוזות??
8. מעבר על כל התרגיל ולוודא את כל הדרישות שלהם

לטפל בהמרות בין i256 ל-[256 x i1]:

1. כניסה לפונקציה
2. חזרה מפונקציה
   1. Ret exp
   2. Ret דיפולטי שאנחנו מחזירים
3. העברת ארגומנטים לפונקציה
4. קריאה מפונקציה.