שאלה 1

(1+2+3)

int m = 0;

int n = 0;

int f(int x){

m++;

return 1;

}

int g(int x){

return x + x;

}

int h(int x){

n++;

return x;

}

void main(){

int x = g(f(h(1)));

if(m==2){ //normal, f will be evaluated 2 times.

printf("normal");

}else if(m==1){

if(n==1){//eager, f will be evaluated one time and

h is evaluated once.

printf("eager");

}else if(n==0){//lazy, f will be evaluated one

time and there is no need to evaluate h.

printf("lazy");

}

}

שאלה 3)

א )

: השיטה הזו כאשר נרצה לשערך ביטוי שנכנסNormal order evaluation

לפונקציה או איזה סביבה אחרת, נשערך הביטוי בכל פעם שמשתמשים בו.

: זה מצב שבו פונקציה שנקראת מתוצאות בסביבה המקוריLexical scoping

שלהן כלומר הסביבה בה הוגדרה הפונקציה בניגוד לסביבה בה נקראת הפונקציה.

ב)

: כל פעם שנכניס ביטוי מסוים לאיזה מכונה Normal order evaluation

לא נשערך אותו ונעביר אותו רקורסיבית פנימה, רק כאשר השערוך עומד לבד נשערך אותו (כלומר זה התנאי העצירה לביצוע הרקורסיה) .

: נצרך לשמור הפונקציה כ זוג, בחלק הראשון תהיה Lexical scoping

הפונקציה ובחלק השני הוא מצביע לסביבה המקורית, בכול פעם שקוראים לפונקציה שומרים מצביע לסביבה הנוכחית (על מנת שנחזור אליה) ואז לעבור לסביבת הפונקציה לבצע אותה כרגיל ובסיום הפונקציה נחזור למיקום שבו הינה בשימוש במצביע שכבר שמרנו .

ג)

If((eval Boolean\_exp env)=(ATOM(SYMBOL("true"),env))

Then (eval true\_expression env)

Else (eval false\_expression env);