



مسئله‌ی ۱.

کد زیر، یک کد ساده در زبان C بوده که کارایی switch را نشان می‌دهد:

```
int main() {
    int operation = 2;
    int n1 = 5;
    int n2 = 7;
    double result;

    switch(operation)
    {
        case '1':
            result = n1 + n2;
            break;
        case '2':
            result = n1 - n2;
            break;
        case '3':
            result = n1 * n2;
            break;
        default:
            result = n1 / n2;
    }
    return 0;
}
```

الف

با چه روش‌هایی می‌توان پرش‌های قسمت switch را مدیریت کرد؟ هرکدام را مختصراً توضیح دهید.

ب

کد میانی که کامپایلر برای کد داده شده تولید می‌کند را بنویسید. در اینجا می‌توانید از دستورات cmp، je و jne برای بخش‌های مختلف switch و مدیریت پرش‌ها استفاده کنید همچنین در جدول زیر نمونه‌ای برای حالت '2' case نشان داده شده است.

i	Program Block	Action Semantic
۷	(cmp, 101, #2, 502)	#compare
۸	(jpf, 502, 15,)	#jump_case

راهنمایی: دستور (cmp, 101, #2, 502) مقدار رجیستر ۱۰۱ و عدد ۲ را مقایسه کرده و خروجی را در ۵۰۲ میریزد، در صورتی که برابر بود مقدار صفر در ۵۰۲ قرار داده، در غیر این صورت مقدار غیر صفر را قرار می‌دهد. همچنین دستور (jne, 101, 102, 14) در صورت برابر نبودن دو رجیستر ۱۰۱ و ۱۰۲ به خط ۱۴ پرش کرده و دستور je در صورتی که برابر باشند پرش را انجام می‌دهد.

مسئله ۲.

قطعه کد زیر را در نظر بگیرید:

```
0  program main()
1      var x:int;
2      procedure f(a:int)
3          var y:int;
4          procedure g(b:int)
5              procedure h(c:int)
6                  > x = c;
7              end h;
8              h(b + a);
9          end g;
10     > y = x + a;
11     g(a - 2);
12     procedure j(d:int)
13         m = d + 5;
14         m = m * 2;
15     end j;
16 end f;
17 x = 5;
18 f(4);
19 end main;
```

فرض کنید رویه‌هایی که صدا زده می‌شوند به این ترتیب باشند:

$f(3) \rightarrow j(4) \rightarrow g(2) \rightarrow h(1) \rightarrow f(5)$

الف

با استفاده از روش Access Link، runtime stack را تا انتها پس از صدا شدن تمام رویه‌ها رسم کنید. (ترتیب قرار گرفتن اطلاعات در Activation Record ها از بالا به پایین به این صورت باشد: Returned Value، Actual parameters، Optional control link، Optional access link، Save machine status، Local data و Temporaries).

ب

حال با استفاده از روش Displays، تا انتهای زمانی که رویه $f(5)$ صدا زده می‌شود runtime stack را رسم نمایید. (ترتیب قرار گرفتن اطلاعات در Activation Record ها همانند قسمت قبل می‌باشد).

ج

با توجه قسمت الف و ب، آدرس‌هایی که برای متغیرهای خطوط ۶ و ۱۰ در زمان کامپایل محاسبه می‌شوند را به هر دو روش بنویسد و مقایسه کنید.

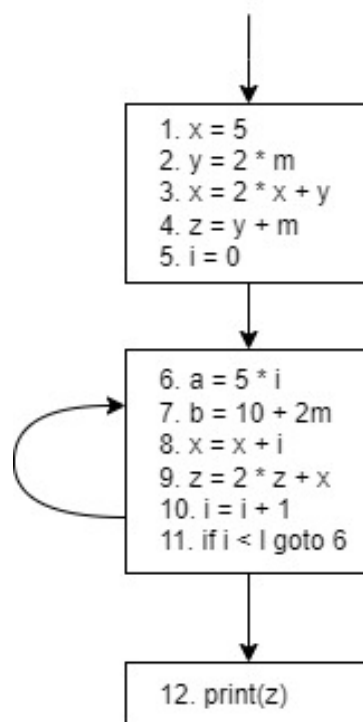
مسئله‌ی ۳.

الف

چهار تکنیک برای بهینه‌سازی محلی^۱ را به نام ببرید و به صورت مختصر توضیح دهید.

ب

به کمک تکنیک‌های بهینه‌سازی محلی CFG روبرو را تا حد امکان ساده کنید. (مرحله به مرحله بهینه‌سازی را مشخص کنید)



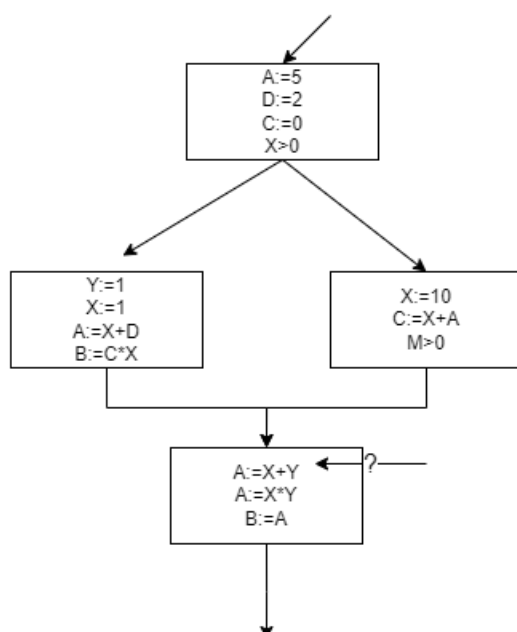
ج

آیا در قسمت بالا امکان آن را داریم که بهینه‌سازی سراسری^۲ داشته باشیم؟ برای جواب خود دلیل آورید.

^۱ Local Optimizatino
^۲ Global Optimiztion

مسئله ۴.

گراف control-flow زیر را در نظر بگیرید:



الف

پس از اجرای الگوریتم constant propagation تا تکمیل، data-flow information صحیح را برای تمامی متغیرها را به دست آورید.

ب

پس از انجام liveness analysis وضعیت تمامی متغیرها را تا خط مشخص شده تعیین کنید. با فرض آنکه تمام متغیرها در خروج در حالت dead قرار گرفته باشند.

ج

با توجه به دو قسمت قبلی، گراف تداخل^۳ را رسم کنید و براساس آن بگویید که حداقل تعداد رجیستر لازم برای اجرای این قطعه کد چقدر باید باشد؟ توضیح دهید که چرا نمی‌توان با کمتر از این تعداد، این قطعه کد را اجرا کرد.

د

فرض کنید که سه رجیستر r_1 ، r_2 و r_3 در اختیار داشته باشید. متغیرها را به گونه‌ای به رجیسترها تخصیص دهید که هیچ دو متغیر مجاور رجیستر یکسانی نداشته باشند و در صورت نیاز عمل Spilling رخ دهد.

^۳Register Inference

مسئله ۵. (امتیازی)

با توجه به گرامر داده شده زیر:

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid a$$

الف

Last Term و First Term غیر پایانه‌ها^۴ را مشخص کنید.

ب

جدول پارس عمل تقدم عملگرا این گرامر را مطابق جدول زیر تکمیل کنید:

	+	*)	(a	\$
+	<				>	
*		<				
)	>		>	=		
(<		
a				<		
\$		>				=

ج

مطابق جدول زیر، رشته وروی $a * (a + a)$ طبق جدول پارس آن با روش تقدم عملگرا تجزیه کنید.

Stack	Input	Relation
\$	$a * (a + a) \$$	$\$ < a$
$\$ < a$	$* (a + a) \$$	$a > +$
$\$ E$	$* (a + a) \$$	$\$ < *$
\vdots	\vdots	\vdots

موفق باشید.