



مسئله‌ی ۱. (۲۵ نمره)

یک کد ساده در زبان C را به صورت زیر را در نظر بگیرید:

```
int opr(int n1, int n2) {
    int result = 0;
    result += n1;
    n2 = n2 / 2;

    for (int i = 0; i < n2; i++) {
        result = result * 2;
    }
    return result;
}

int main() {
    int operation = 3;
    int n1 = 4;
    int n2 = 3;
    double result;

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

کد میانی که کامپایلر برای کد داده شده تولید می‌کند را بنویسید. در اینجا می‌توانید از دستورات cmp، je و jne برای مدیریت پرش‌ها استفاده کنید همچنین در جدول زیر نمونه‌ای برای حالت '1' case نشان داده شده است.

i	Program Block	Action Semantic
15	(cmp, 101, #2, 502)	#compare
16	(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 غیرصفر 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.

مسئله‌ی ۲. (۲۵ نمره)

برنامه زیر را در نظر بگیرید.

```
L0: e := 0
    b := 1
    d := 2
L1: a := b + 2
    c := d + 5
    e := e + c
    f := a * a
    if f < c goto L3
L2: e := e + f
    goto L4
L3: e := e + 2
L4: d := d + 4
    b := b - 4
    if b != d goto L1
L5:
```

این برنامه از شش متغیر موقت a-f استفاده می‌کند. فرض کنید تنها متغیری که در هنگام خروج از آن فعال است e باشد.

الف. (۳ نمره)

گراف Control-flow کد داده شده را رسم کنید.

ب. (۱۰ نمره)

پس از انجام liveness analysis وضعیت تمامی متغیرها را تا خط سوم قطعه کد داده شده تعیین کنید. (منظور خط $d := 2$ است)

ج. (۷ نمره)

با توجه به دو قسمت قبلی، گراف تداخل^۱ 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. (۵ نمره)

فرض کنید که چهار رجیستر r_1, r_2, r_3 و r_4 در اختیار داشته باشید. متغیرها را به گونه‌ای به رجیسترها تخصیص دهید که هیچ دو متغیر مجاور^۱ی رجیستر یکسانی نداشته باشند و در صورت نیاز عمل Spiling رخ دهد.

^۱ Register Inference

مسئله‌ی ۳. (۲۵ نمره)

الف. (۵ نمره)

چهار تکنیک برای بهینه‌سازی سراسری^۲ را به نام ببرید و به صورت مختصر توضیح دهید.

ب. (۲۰ نمره)

به بخش‌های کد سه آدرسی زیر توجه کنید. فرض کنید مقادیر m و n از ابتدا معلوم هستند؛ و در پایان، تنها مقدار b مورد نیاز است. بلاک‌های اصلی را شناسایی کرده و نمودار جریان کنترل کد را رسم کنید. سپس با اعمال تمام بهینه‌سازی‌های محلی و سراسری ممکن، کد را بهینه کنید. تمام تبدیل‌هایی که برای بهینه‌سازی استفاده کرده‌اید را نام ببرید.

```
1. a := 0
2. i := n
3. j := m
4. k := 16 * j
5. v := 8 * k
6. i := i - 1
7. j := j + 1
8. u := i * j
9. t2 := 4 * u
10. if t2 > v goto 6
11. p := t2 - 1
12. t2 := t2 - p
13. if i >= j goto 21
14. t3 := v + 1
15. t4 := a + t3
16. a := t4
17. if j > v goto 21
18. t5 := 8 * j
19. b := t5
20. goto 6
21. t6 := 32 * k
22. t7 := a + t6
23. a := t7
24. t8 := 4 * a
25. t8 := t8 + a
26. b := t8
```

^۲Global Optimiztion

مسئله ۴. (۲۵ نمره)

قطعه کد زیر را در نظر بگیرید:

```
0  program main()
1      var x:int;
2      var z:int;
3      procedure f(a:int)
4          var y:int;
5          procedure g(b:int)
6              procedure h(c:int)
7                  x = c;
8                  procedure q(e:int)
9                      z = e + x;
10                     end q;
11                     q(c * 2);
12                 end h;
13                 h(b + a);
14                 procedure k(l:int)
15                     procedure m(n:int)
16                         y = n + z;
17                     end m;
18                     m(l - 3);
19                 end k;
20                 k(b * 3);
21             end g;
22             y = x + a;
23             g(a - 2);
24             procedure j(d:int)
25                 var m:int;
26                 m = d + 5;
27                 m = m * 2;
28                 procedure p(o:int)
29                     y = o + m;
30                 end p;
31                 p(d * 2);
32             end j;
33             j(a * 4);
34         end f;
35         x = 5;
36         z = 10;
37         f(4);
38     end main;
```

الف. (۳ نمره)

ترتیب رویه‌هایی^۳ که صدا زده می‌شوند به چه صورت است؟

ب. (۸ نمره)

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: Actual، Returned Value، parameters، Optional control link، Optional access link، Save machine status، Local data و Temporaries).

^۳Procedures

ج. (۸ نمره)

حال با استفاده از روش Displays، تا انتهای زمان اجرا کد Runtime Stack را رسم نمایید.
(ترتیب قرار گرفتن اطلاعات در Activation Record ها همانند قسمت قبل می باشد).

د. (۶ نمره)

با توجه قسمت ب و ج، آدرس هایی که برای متغیرهای خطوط ۹، ۲۶ و ۲۹ در زمان کامپایل محاسبه می شوند را به هر دو روش بنویسد و مقایسه کنید.

مسئله ۵. (۱۰ نمره امتیازی)

با توجه به گرامر داده شده زیر:

$$\begin{aligned} S &\rightarrow A \mid S - A \\ A &\rightarrow B \mid A * B \\ B &\rightarrow C \mid B + C \\ C &\rightarrow (C) \mid id \end{aligned}$$

الف. (۳ نمره)

Last Term و First Term غیر پایانه ها^۴ را مشخص کنید.

ب. (۷ نمره)

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

Stack	Input	Relatinon
\$	$id - (id * (id + id))$ \$	$\$ < id$
:	:	:

موفق باشید.