



سوال ۱ (۲)

کد میانی که کامپایلر برای کد زیر تولید میکند را بنویسید. از جدول زیر برای نوشتن کد میانی کمک بگیرید.

```
int f(int a, double d)
{
    int i = a / d;
    return i;
}

int main()
{
    int a = 5;
    return f(a, 2.5);
}
```

i	PB[i]	Actions Semantic
.		
\		

سوال ۲ (۲)

کد زیر بر اساس یک زبان فرضی می باشد، (ترکیبی از C++ و java). جدول نمادهای (Symbol Table) و Scope Stack آن را در زمان کامپایل در خطهای ۸ و ۱۴ و ۱۷ و ۱۹ را رسم نمایید.
جدول نمادها شامل موارد زیر است:

scope, type, No. Arg/cell, proc/func/var, Lexeme

```
1: Program lexem()
2:   Var j , i , digit[1..10] real
3:   Procedure getNum(k: integer)
4:       Var blo[1..10] integer
5:       Procedure getDigit()
6:           Var i, j[1..15] integer
7:           digit (8) := j (5) + blo (7)
8:           i := digit (5) + blo (7)
9:       End getDigit
10:      Function getLine(d: integer): integer
11:          Var digit integer
12:          Procedure sum()
13:              Var blo real
14:              blo := digit + j
15:          End sum
16:      End getLine
17:      blo (8) := digit(4) + j
18:  End getNum
19:  digit(3) := digit(9) + j
20: End lexem
```

سوال ۳ (۳)

برنامه زیر را در نظر بگیرید که متغیرهای live در آن مشخص شده اند.

```
// live: {v, x}
u = v + 1
// live: {u, v, x}
w = u - v
// live: {u, w, x}
x = x + w
// live: {u, w, x}
y = u - w
// live: {x, y}
z = x + y
// live {z}
```

الف) گراف تداخل (Interference Graph) را برای آن رسم کنید.
ب) کمترین تعداد رنگ برای رنگ کردن گراف به گونه ای که ریختن (Spilling) صورت نگیرد را بیابید. توضیح دهید که چگونه عددی که یافتید کمترین تعداد رنگ است؟
پ) فرض کنید که سه رجیستر r_1 , r_2 , r_3 دارید. متغیرها را به گونه ای به رجیسترها تخصیص دهید (Allocate) که هیچ دو متغیر مجاوری رجیستر یکسان نداشته باشند و در صورت نیاز Spilling رخ دهد.

سوال ۴ (۳)

کد زیر را در نظر بگیرید و به سوالات پاسخ دهید

الف- نمودار بلوک های پایه را رسم کنید (در هر بلاک پایه شماره دستورالعمل ها را بنویسید)

ب- حلقه ها را مشخص کنید

پ- دستورات مستقل از حلقه را در جای مناسب قرار دهید

ت- بهینه سازی محلی در سطح بلوک پایه را برای کد مقابل انجام دهید.

ث- با فرض اینکه کل برنامه تولید شده کد مقابل است، آیا دستوری وجود دارد که بتوان آن را حذف کرد؟

```

1:      :=, I, #0
2:      L0:
3:      >=, I, N, EXIT
4:      L1:
5:      %, I, X, t0
6:      ==, t0, #2, L2
7:      :=, J, #0
8:      L3:
9:      <, J, N, L2
10:     *, #4, n, t1
11:     *, #2, n, t2
12:     *, #4, #11, t3
13:     +, t3, t2, t3
14:     +, t3, t3, t2
15:     :=, t4, #0
16:     ^, #3, I, t5
17:     +, K, I, t6
18:     +, t6, J, t7
19:     *, J, t1, t8
20:     +, t8, t5, t9
21:     :=, @t7, t9
22:     Inc J
23:     JP L3
24:     L2:
25:     Inc I
26:     Jp L0
27:     EXIT:

```