



## مسئله ۱.

برای برنامه زیر، activation tree را رسم کنید.  
- فرض کنید record activation برای هر تابع شامل موارد زیر می شود:

- ورودی های تابع
- متغیر های محلی

```
void main(){
    int arr[] = {3, 34, 1, 5, 6, 8};
    mergeSort(arr, 0, arr.length - 1);
}

void mergeSort(int arr[], int l, int r) {
    if (l < r) {
        int m = (l + r) / 2;
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);
        merge(arr, l, m, r);
    }
}

void merge(int arr[], int l, int m, int r)
{
    // Merges arr[l, m] and arr[m + 1, r]
}
```

## مسئله ۲.

برای قطعه کد زیر ساختار Stack Frame را برای هر یک از موارد زیر نشان دهید. (در پشته نشان داده شده تعیین کنید هر مقدار را کدام تابع در پشته push می کند).

الف) زمانی که اولین بار خط ۱۱ اجرا می شود. (بعد از اجرا شدن)

ب) زمانی که خط ۸ اجرا می شود. (بعد از اجرا شدن)

( مقدار fp را نسبت به fp اولیه تعیین کنید و مقدار ra را با استفاده از شماره خطوط زیر تعیین کنید.)

```
1. void main() {
2.     int a = 24;
3.     int b = 36;
4.     int c = gcd(a, b);
5. }
6. int gcd(x, y) {
7.     if (x == 0){
8.         return y;
9.     }
10.    int r = b % a;
11.    return gcd(r, a);
12. }
```

### مسئله 3.

خروجی برنامه زیر را در هر یک از شرایط زیر با ذکر دلیل تعیین کنید.

- Call by value
- Call by reference
- Call by name
- Call by value-result

```
int n = 1;
int array[] = {0, 1, 2, 3, 4};
```

```
void printAll(int a, int b){
    int t;
    int i = 0;
    t = a;
    a = b;
    b = t;
    print(a, b, i, array);
}
```

```
Void main() {
    printAll(i, array[i]);
    print(i, array);
}
```

#### مسئله‌ی 4.

با توجه به کد زیر vtable را برای اشیا زیر رسم کنید.

SUV s = SUV() •

Drivable b = BMW() •

```
Class Vehicle {
    Public void getName();
    Public int getSpeed();
}
Class SUV extends Vehicle {
    Int wheelDrive;
    Public void getCapacity();
}
Interface Drivable {
    Public void drive();
    Public void park();
}
Class BMW extends SUV implements Drivable {
    Int modelNum;
    Public void maintain();
}
```

#### مسئله‌ی 5.

برای کد زیر، کد سه آدرس تولید کنید. (پارامترها به شکل pass by value به توابع داده می شوند)

```
class A {
    int fieldA;
    int methodA(int a){
        return (a + this.fieldA) * 10;
    }
}
class B extends A {
    int fieldB;
    int methodB(int param){
        return this.methodA(6) * (param + this.fieldB);
    }
}
```

```

}
Void main(){
    A a = B();
    b.fieldA = 5;
    b.fieldB = 10;
    int x;
    x = b.methodB(2);
}

```

## مسئله‌ی 6.

کد زیر را در نظر بگیرید.

```

a = 1 + 2;
b = a + b;
z = a * 2;
c = b + e;
d = c + b;
x = b + 3;
z = a * 8;
t = c - 2;
f = x + f;
y = x - 2;
d = d - y;

```

فرض کنید بعد از اجرای این کد متغیرهای d و z و x زنده اند.

الف) برای هر خط متغیرهای زنده را بنویسید.

ب) با کمک بهینه سازی های محلی که در درس یادگرفته اید سعی کنید این کد را به بهینه ترین حالت ممکن تبدیل کنید.

## مسئله‌ی 7.

کد زیر را در نظر بگیرید.

```

l = a + 1;
m = b + 2;
n = c + x;
do {

```

```

c = c - 1;
k = c + m;
i = m;
k = 2;
m = m * i - k;
if( c < a )
    n = y;
    b = k;
else
    c = z;
    l = n + j;
k = m - a;
x = n + j;
}
while (k > 0);

```

الف) نمودار جریان داده ای را برای آن رسم کنید.

ب) تحلیل متغیرهای زنده را برای این کد انجام دهید.

ج) انتشار سراسری ثابت انجام دهید.

د) افزونگی های جزئی را حذف کنید.

مسئله 8.

کد زیر را در نظر بگیرید.

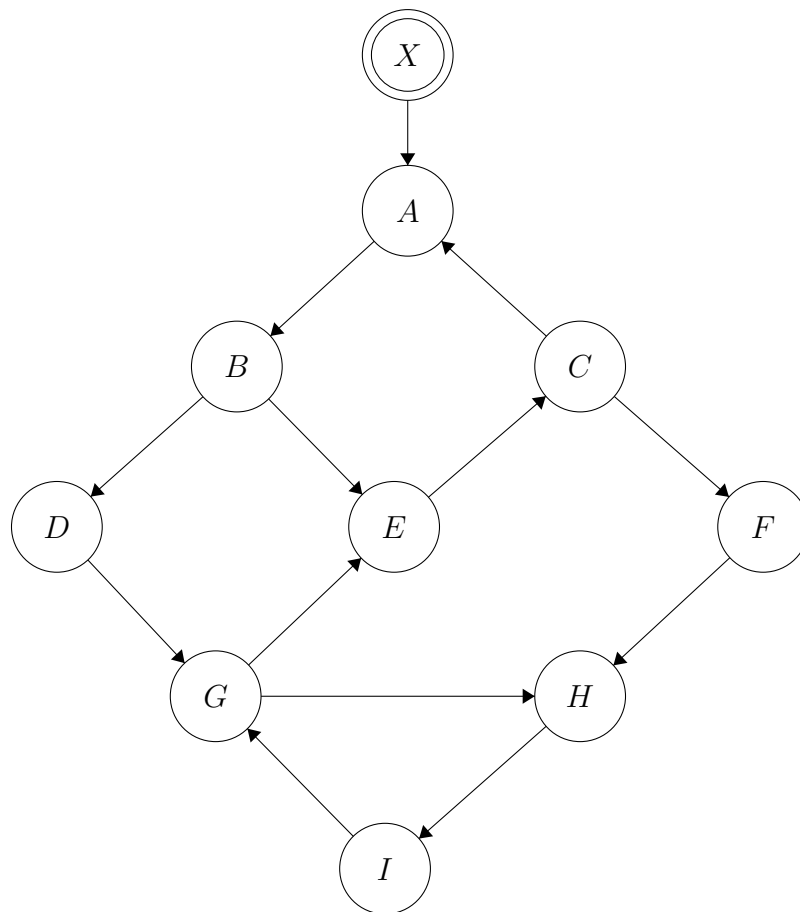
```

a = b
c = 7 + 7 * e
d = a
a = d * d
d = 5 * a
f = c * 5 + 10
f = d - f
c = f + 1
e = c * b
print(c, b)

```

الف) گراف تداخل رجیسترها را رسم کنید.

ب) روال اجرای الگوریتم chaitin را بر روی آن توضیح دهید و کمترین تعداد رجیستر را بدست آورید.



با توجه به شکل فوق به سوالات زیر پاسخ دهید:

- تعداد ارجاعات به هر شیء پس از حذف اشاره‌گر A به B چگونه خواهد بود؟
- پس از حذف اشاره‌گر A به D چه اتفاقی برای ارجاعات رخ خواهد داد؟
- فرض کنید که اشیاء به الفبایی در حافظه چیده شده‌اند و سائز هر یک نیز ۵۰ بایت می‌باشد. با فرض اینکه آدرس این اشیاء از خانه ۰ام حافظه شروع شده باشد پس از حذف رأس B و انجام عمل زباله‌روبی، آدرس هر شیء چه خواهد بود؟

گفته بودی که چرا محو تماشای منی؟  
 آن‌چنان مات که یک‌دم مژه بر هم نزنمی  
 مژه بر هم نزنم تا که ز دستم نرود  
 ناز چشم تو به قدر مژه بر هم زدنمی