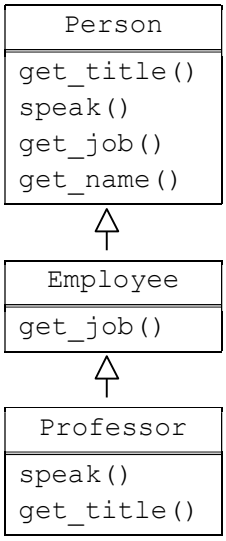
پرسش ۱ (۱ نمره) – با در نظر گرفتن سلسله مراتب کلاس‌های مقابل، مشخص کنید در قطعه برنامه‌ی زیر، هر یک از جملات کدام‌یک از متدها را فراخوانی می‌کند.




```
Professor p;
p.get_name();
p.get_title();
p.get_job();
p.speak();
```

پرسش ۲ (۲ نمره) – فرض کنید سه کلاس Ball، MovingObject و Point به ترتیب متناظر با مفاهیم *توپ*، *متحرک* و *نقطه* تعریف شده‌اند. بین این سه کلاس کدام‌یک از رابطه‌های Has-A و Is-A برقرار است؟

پرسش ۳ (۲ نمره) – چهار خطای زمان کامپایل برنامه‌ی زیر را مشخص کنید، مختصراً توضیح دهید و بگویید چگونه می‌توان آن را رفع کرد.

```
#include <string>
using namespace std;

class Date {
public:
    Date(int d, int m, int y) : day(d), month(m), year(y) {}
private:
    int day, month, year;
};

class Robot {
public:
  void Robot(string mod, string man, int d, int m, int y)
    : model(mod), manufacturer(man) {
      production_date = Date(d, m, y);
    }
  bool same(Robot r) { return (r.model == model); }
private:
  string model;
  string manufacturer;
  Date production_date;
};

int main() {
  Robot r("Asimo", "Honda", 10, 3, 2001);
  if (same(Robot("Jane", "Fonda", 21, 12, 1937)))
    return r.model.length();
}
```

پرسش ٤ (۴ نمره) – تابعی بازگشتی بنویسید که با دریافت دو بردار از اعداد صحیح، اگر زیردنباله‌هایی که از حذف اعداد فرد در آنها به دست می‌آید دقیقاً یکسان بودند مقدار درست وگرنه مقدار نادرست برگرداند. به عنوان مثال، نتیجه‌ی این تابع برای دو بردار `{1,0,2,3,5,4,0}` و `{0,1,2,4,7,0,3,5}` مقدار درست خواهد بود، چون زیردنباله‌ای که از حذف اعداد فرد از هر دوی آنها به دست می‌آید `{0,2,4,0}` است. در صورت نیاز برای این تابع پارامتر(های) اضافه تعریف کنید. در حل این سؤال از حلقه یا تابع کمکی دیگری استفاده نکنید و به جز دو پارامتر ورودی از تایپ‌های نگهداری دنباله‌هایی از اعداد (بردار، لیست، آرایه، ...) استفاده نکنید.

پرسش ٥ (۳ نمره) – قطعه برنامه زیر استفاده از کلاسی به نام Rectangle را نشان می‌دهد. با توجه به این قطعه برنامه، بخش عمومی کلاس Rectangle را به طور کامل تعریف کنید. واسط این کلاس را از قطعه کد زیر استنتاج کنید. نیازی به نوشتن پیاده‌سازی کلاس، شامل فیلدها و بدنه‌ی متدها نیست و فقط امضای متدها و سازنده(ها) را بنویسید.

```
Rectangle r(0, 0, 10, 20);
cout << r.x() << r.y() << endl;
Rectangle s = r.scale_by(2);
cout << s.width() << ' ' << s.height() << endl;
cout << r.contains(Rectangle(5, 5, 3, 2));
vector<Rectangle> v(10);
v.push_back(r);
```

پرسش ٦ (۴ نمره) – کلاس‌های زیر را در نظر بگیرید و نتیجه‌ی کامپایل یا اجرای هر یک از قطعه‌کدهای زیر را مشخص کنید.

```
#include <iostream>
using namespace std;

class A {
public:
  virtual void f() { cout << 1; }
  virtual void g() { f(); }
  void h() { cout << 3; }
  virtual void i() = 0;
};

class B : public A {
public:
  virtual void f() { cout << 2; }
  void h() { f(); }
};

class C : public B {
public:
  virtual void i() { cout << 4; }
  virtual void f() { cout << 5; }
};

class D : public C {
public:
  virtual void i() { cout << 6; }
  virtual void g() { cout << 7; }
};
```

الف)

```
C c;
A* ap = &c;
ap->f();
ap->g();
ap->h();
ap->i();
```

ب)

```
B b;
B* bp = &b;
bp->f();
bp->g();
bp->h();
bp->i();
```

پ)

```
C c;
B* bp = &c;
bp->f();
bp->g();
bp->h();
bp->i();
```

ت)

```
C c;
D* dp = &c;
dp->f();
dp->g();
dp->h();
dp->i();
```

پرسش ۷ در صفحه بعد

پرسش ۷ (۴ نمره) – فرض کنید مجموعه‌ای از لغات داده شده‌اند. هدف این است که با دریافت یک رشته از حروف الفبا بدون کاراکتر فاصله تعیین کنیم آیا می‌توان با اضافه کردن تعدادی فاصله‌ی خالی بین حروف جمله‌ای ایجاد کرد که از لغات داده شده تشکیل شده باشد؟ مثلاً اگر مجموعه لغات زیر را داشته باشیم:

ark, go, mark, the, them, there, they

رشته‌ی themarkgothere را می‌توان به صورت them ark go there فاصله گذاری کرد. البته لزومی ندارد جمله‌ی حاصل معنی‌دار یا حتی از نظر دستور زبان صحیح باشد. در نظر داشته باشید از یک لغت می‌توان چند بار در رشته استفاده کرد. مثلاً نتیجه برای thegothe درست است.

تابعی با امضای زیر بنویسید که در صورتی که رشته‌ی ورودی را در s، و مجموعه لغات را در dict بگیرد و مسئله را به روش backtracking حل کند. در صورتی که نتیجه‌ی فاصله‌گذاری درست باشد مقدار درست وگرنه نادرست برمی‌گرداند. اگر نتیجه درست باشد، کلمات جدا شده با فاصله در words قرار می‌گیرند.

```
bool solve(const string& s, const list<string>& dict, list<string>& words);
```

این تابع را طوری بنویسید که فقط یک حلقه در آن باشد و برای نوشتن آن تابع کمکی دیگری ننویسد، هر چند از متدهای string و list می‌توانید استفاده کنید.