

پاک‌شدن پروژه

- محدودیت زمان: ۰.۵ ثانیه
- محدودیت حافظه: ۶۴ مگابایت

کیخسرو که دنبال افزایش نمره‌ی خود در درس ریاضی ۱ بود تصمیم گرفت پروژه‌ای انجام دهد و برای استادش بفرستد. اما متأسفانه قبل از اینکه پروژه را برای استاد ایمیل کند، دستش خورد و هر چیزی که در برنامه‌اش قبل از تابع main نوشته بود پاک شد 🤖

Copy ++C

```

1
2 everything was here destroyed :(
3 help Keykhosro and insert something here!
4
5 int main() {
6     cout << fixed << setprecision(2);
7     double x1, x2, y1, y2;
8     cin >> x1 >> y1 >> x2 >> y2;
9     Complex c = constructComplex(x1, y1);
10    Complex c2 = constructComplex(x2, y2);
11    Complex c3 = conjugate(c);
12    cout << c3.real << ' ' << c3.img << '\n';
13    Complex c4 = add(c, c2);
14    cout << c4.real << ' ' << c4.img << '\n';
15    Complex c5 = sub(c, c2);
16    cout << c5.real << ' ' << c5.img << '\n';
17    Complex c6 = mul(c, c2);
18    cout << c6.real << ' ' << c6.img << '\n';
19    printPolarForm(c);
20    return 0;
21 }
```

کیخسرو که دیگر حال و حوصله‌ی دوباره انجام دادن پروژه را ندارد از شما خواسته تا بجای او پروژه‌اش را کامل کنید.

در این پروژه قرار بود کیخسرو ساختاری بنویسد که عددی مختلط به فرم $x + iy$ در آن ذخیره شود و به کمک توابع نوشته‌شده در برنامه عملیات زیر روی آن قابل انجام باشد:

۱. ساخت و ذخیره‌کردن عدد مختلط $x + iy$ در صورتی که x و y را داشته باشیم.

```
1 | Complex c;
2 | c = constructComplex(1, 1.732); //1 + 1.732i
```

Copy ++C

۲. بدست‌آوردن مزدوج عدد مختلط c و ذخیره‌ی آن در متغیری دیگر.

```
1 | Complex c;
2 | c = constructComplex(1, 1.732); //1 + 1.732i
3 | Complex c2;
4 | c2 = conjugate(c); //1 - 1.732i
```

Copy ++C

۳. جمع و تفریق دو عدد مختلط c و $c2$ و ذخیره‌ی آن در متغیری دیگر.

```
1 | Complex c = constructComplex(2, 3); //2 + 3i
2 | Complex c2 = constructComplex(5, -2); //5 - 2i
3 | Complex res = add(c, c2); //(2 + 3i) + (5 - 2i) = 7 + i
4 | Complex res2 = sub(c, c2); //(2 + 3i) - (5 - 2i) = -3 + 5i
```

Copy ++C

۴. ضرب‌کردن دو عدد مختلط c و $c2$ و ذخیره‌ی آن در متغیری دیگر.

```
1 | Complex c = constructComplex(1, 1); //1 + i
2 | Complex c2 = constructComplex(2, -1); //2 - i
3 | Complex res = mul(c, c2); //(1+i)(2-i) = 3 + i
```

Copy ++C

۵. تبدیل به فرم قطبی و چاپ کردن فرم تبدیل‌شده.

```
1 | Complex c = constructComplex(1, 1); //1 + i
2 | printPolarForm(c); //1.41e^(i45.02)
```

Copy ++C

در مثال بالا خروجی چاپ‌شده $1.41e^{(i45.02)}$ خواهد بود که با تقریب خوبی برابر $\sqrt{2}e^{i45}$ است.

راهنمایی:

- برای \tan^{-1} گرفتن در C++ می‌توانید کتابخانه‌ی `cmath` را `include` کنید و با استفاده از `atan2(y,x)` مقدار θ را در $\tan\theta = \frac{y}{x}$ برحسب رادیان بدست آورید و سپس به واحد درجه تبدیل کنید. (برای سادگی کار تضمین می‌شود x یا همان بخش حقیقی عدد هرگز صفر نیست)
- برای رادیکال گرفتن (بعد از `include` کردن `cmath`) از تابع `sqrt` استفاده کنید.
- در تبدیل از رادیان به درجه عدد π را 3.14 در نظر بگیرید.
- در تابع `main` هیچ تغییری ایجاد نکنید و هر چیزی اضافه می‌کنید به قبل از `main` اضافه کنید.

مثال

ورودی نمونه ۱

```
1 1.732
1 -1.732
```

Copy

خروجی نمونه ۱

```
1.00 -1.73
2.00 0.00
0.00 3.46
4.00 0.00
2.00e^(i60.03)
```

Copy

ورودی نمونه ۲

```
1 0.57735
0 10.36
```

Copy

خروجی نمونه ۲

Copy

1.00 -0.58
1.00 10.94
1.00 -9.78
-5.98 10.36
 $1.15e^{(i30.02)}$

ورودی نمونه ۳

Copy

37.43 33.061
3.7 7.4

خروجی نمونه ۳

Copy

37.43 -33.06
41.13 40.46
33.73 25.66
-106.16 399.31
 $49.94e^{(i41.47)}$

ورودی نمونه ۴

Copy

11 7.9
93 10

خروجی نمونه ۴

Copy

11.00 -7.90
104.00 17.90
-82.00 -2.10
944.00 844.70
 $13.54e^{(i35.70)}$

