

Parrray

حال در این سوال می خواهیم کلاسی را طراحی کنیم که بیانگر آرایه از نقاط (Points) هستند. در این سوال قصد داریم که آرایه به صورت داینامیک و پویا سایز خودش را تغییر دهد و سایز خود را ردیابی می کند تا اگر بخواهیم آن را به تابعی بفرستیم نیازی به فراهم کردن اندازه ی آرایه نباشد.

کلاسی به نام `PointArray` بسازید که دارای دو عضو `private` می باشد که عبارتند از یک اشاره گر که به ابتدای آرایه ای از نقاط اشاره می کند و همچنین یک عدد صحیح که نشان دهنده ی سایز (اندازه) آرایه می باشد.

کلاستون دارای سازنده (Constructors) های زیر می باشد.

- یک سازنده پیشفرض (default constructor) که آرایه ای با سایز 0 درست می کند.
- یک سازنده که آرایه ای از `point` ها به نام `points` و همچنین یک `int` به نام `size` را به عنوان آرگومان های ورودی می گیرد. و این سازنده باید یک `PointArray` با سایز گفته شده بازند و مقادیر داخل `points` را کپی کند. شما باید به آرایه ی داخل کلاستون به صورت پویا به اندازه ی گفته شده (`size`) حافظه تخصیص بدین. (حواستان باشد در جاهایی مثال اینجا که قرار نیست مقداری که به عنوان ورودی به تابع داده می شود را تغییر دهیم حتما آن را به صورت `const` تعریف کنید.)

کلاستون دارای مخرب (Destructor) زیر می باشد

- یک `Destructor` که کارش `Delete` کردن آرایه ی داخلی کلاسمون می باشد.
- از آنجایی که می توانیم بر روی آرایه ی خود اصلاحات انجام دهیم باید راهی پیدا کنیم که آرایه داخلی کلاسمون رو بتونیم بزرگ یا کوچک کنیم.
- یکی از کار هایی که می توانیم انجام دهیم نوشتن یک `member function` به نام `PointArray::resize(int n)` می باشد که کارش این است که به آرایه ی جدیدی حافظه ای به سایز `n` تخصیص دهد. مقادیر آرایه ی قبلی را وارد این آرایه ی جدید کند و آرایه ی قبلی را `deallocate` کند.
- در صورت که سایز آرایه قرار باشه که اضافه بشه مشکلی نیست که تابع `resize` مون فضاهای جدید را مقدار دهی اولیه نکند و پر کردن فضاهای اضافه وظیفه ی کسی هست که این تابع را صدا زده.

سپس هر جایی که قرار باشد سائز آرایه را تغییر دهیم (حتی در هنگام clear) می توانیم از این تابع استفاده کنیم. در بعضی موارد لازم هستش که پس از resize کردن آرایه مون رو به چپ یا راست شیفت بدیم تا جا برای مقادیر جدید باز بشه یا از شر مقادیر قبلی خلاص بشیم :) که البته خیلی کارآمد نمی باشد ولی در این تمرین خیلی نگران کارآمد بودن از این لحاظ نیستیم ولی اگر می خواستین از راه "درست" پیش برین باید حواستان باشه که طول آرایه تون چقدر هستش و چقدرش پر شده و تنها در صورتی reallocate کنید که به حد فعلی رسیدین و یا زیر یک آستانه ای که در نظر گرفتن قرار گرفتید.

پس تابع `PointArray::resize(int n)` را به عنوان توابع کلاستون اضافه کنید و به آن access modifier بدهید که تنها توابع داخلی کلاستون بتونن ازش استفاده کنند.

و در نهایت توابع زیر را به عنوان Member Functions به کلاس خود اضافه کنید.

```
void PointArray::PushBack(const Point &p)
```

یک Point در انتهای آرایه اضافه می کند.

```
void PointArray::insert(const int position, const Point &p)
```

با شیفت دادن آرایه به راست یک point را در جای دلخواه قرار می دهد.

```
void PointArray::remove(const int pos)
```

از جای دلخواه آرایه یک point را حذف می کند.

```
const int PointArray::getSize() const
```

سائز آرایه را باز می گرداند.

```
void PointArray::clear()
```

آرایه را پاک می کند و سائزش را برابر با صفر قرار می دهد.

```
Point *PointArray::get(const int position)
const Point *PointArray::get(const int position) const
```

پوینتری به عنصر موجود در مکان دلخواه مشخص شده توسط position باز می گرداند. در صورتی که همچنین position ای وجود نداشته باشد null بر می گرداند.

تکات

- از چند فایل به نوشتن استفاده کنید
- حواستان به چک کردن محدوده آرایه تون در توابع پابلیکتون باشید و در صورت رعایت نکردنش پیام مناسب چاپ کنید.