

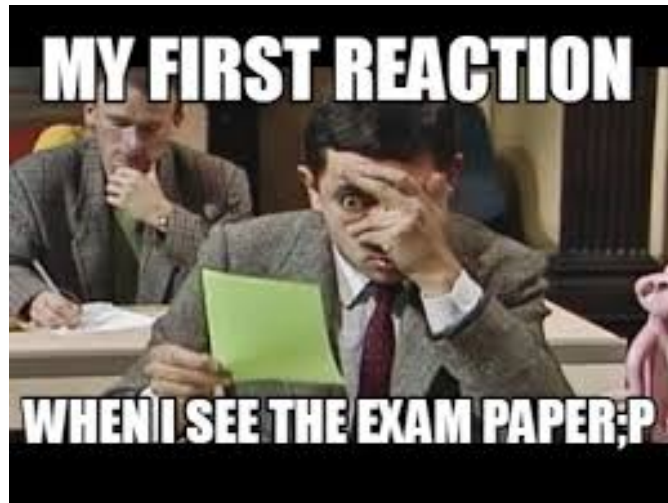
بسم الله الرحمن الرحيم

امتحان میان ترم برنامه نویسی پیشرفته

دکتر جهانشاهی

سه شنبه ۹ اردیبهشت ۱۳۹۹

مدت زمان: ۱۵۰ دقیقه



۱ مقدمه

بررسی های اخیر نشان داده اند که نمرات پایانی دانشجویان AP از الگو های خاصی پیروی می کنند. پس از مطالعات آماردان ها مشخص شد نمره پایانی هر دانشجو یک متغیر تصادفی با توزیع احتمال (pdf^1) مشخص می باشد. این توزیع احتمال یکی از دو توزیع معروف نرمال یا نمایی می باشد که پارامتر آن توسط عملکرد دانشجو در طول ترم تعیین می شود. امروز ابتدا به پیاده سازی یک صف 2 برای نگهداری داده ها خواهید پرداخت و سپس از آن صف برای نگهداری دانشجو ها استفاده خواهید کرد. جهت جذاب تر کردن امتحان امروز، برخی از نوع متغیر ها، خروجی های توابع و تعریف توابع بطور صریح گفته نشده و انتظار می رود که شما بتوانید با کمک توضیحات داده شده به همراه Google Test های داده شده آن ها را تعیین کنید.

لطفا تمامی فایل های خود را در قالب یک فایل zip (نه rar) در سامانه courses بارگذاری کنید. نام فایل شما باید در فرمت زیر باشد.

Name+StudentNumber.zip

مثلا

MohammadHosseinAmini9423013.zip

موفق باشید.

¹Probability Density Function

²Queue Data Structure

۲ APQueue

ابتدا به پیاده سازی یک صف برای نگهداری هر چیزی! خواهید پرداخت. همانگونه که می دانید، صف نوعی از ساختمان داده است که در آن قانون FIFO برقرار است. بدین معنی که هر عضوی که اول وارد شود، اول خارج خواهد شد. درست مانند یک صف در دنیای واقعی. برای پیاده سازی صف از کلاس APQueue استفاده خواهید کرد. همانگونه که در کلاس درس و تمرینات دیده اید، در این کلاس از اشیائی با نوع Item جهت نگهداری داده ها استفاده خواهید کرد. هر شیء از کلاس Item دارای آدرس آیتم بعدی می باشد. دقت کنید که به هیچ وجه نباید آدرس آیتم قبلی را در هر آیتم ذخیره کنید (صف باید بصورت یک طرفه باشد). متغیرها و توابع کلاس Item بصورت زیر می باشند.

۱. pnext

پوینتر برای نگهداری آدرس آیتم بعدی.

۲. pvalue

پوینتر برای نگهداری شیء یا داده ای که قصد نگهداری آن توسط آیتم را داریم.

کلاس APQueue دارای متغیرها و توابع زیر می باشد.

۱. phead

پوینتر جهت نگهداری آدرس اولین آیتم صف.

۲. ptail

پوینتر جهت نگهداری آدرس آخرین آیتم صف.

۳. enqueue

این تابع یک داده جدید می گیرد و آن را به آخر صف اضافه می کند.

۴. dequeue

این تابع اولین عضو صف را از صف خارج کرده و آن را return می کند.

توجه تعریف تابع مخرب برای این کلاس را فراموش نکنید!

صف شما باید به گونه ای پیاده سازی شود که کدهای زیر بدون هیچ خطایی اجرا شوند و نتیجه مطلوب را داشته باشند.

```
۱
۲ std::vector<int> vec{28, 13, 49};
۳ APQueue<int> apq{vec};
۴ apq.show();
۵ apq.enqueue(52);
۶ apq.show();
۷ apq.dequeue();
۸ apq.show();
۹ apq.dequeue();
۱۰ apq.dequeue();
۱۱ apq.dequeue();
۱۲ apq.dequeue();
۱۳ apq.show();
۱۴ (&apq)->enqueue(12)->enqueue(13)->enqueue(28);
۱۵ apq.show();
```

خروجی کدهای بالا در شکل ۱ نمایش داده شده است.

APQueue of size: 3		
No :	1	28
No :	2	13
No :	3	49
APQueue of size: 4		
No :	1	28
No :	2	13
No :	3	49
No :	4	52
APQueue of size: 3		
No :	1	13
No :	2	49
No :	3	52
APQueue of size: 0		
APQueue of size: 3		
No :	1	12
No :	2	13
No :	3	28

شکل ۱: خروجی شماره ۱

۳ Student

در AP دو نوع دانشجو داریم: دانشجویان نرمال و دانشجویان نمایی! دانشجویان نرمال، دانشجویانی هستند که نمره نهایی آن‌ها دارای توزیع احتمال نرمال با میانگین و انحراف معیار مشخص می‌باشد. همانگونه که حدس زده‌اید، دانشجویان نمایی کسانی هستند که نمره پایانی آنان دارای توزیع احتمال نمایی با پارامتر مشخص می‌باشد. پس ابتدا، کلاس Student را پیاده‌سازی کنید. این کلاس دارای توابع و متغیرهای زیر می‌باشد.

۱. name

یک متغیر از نوع string برای نگهداری نام دانشجو. مقدار دیفالت این متغیر را برابر نام خود بگذارید (:

۲. pdf

این تابع یک عدد اعشاری به عنوان نمره گرفته و احتمال این که دانشجو آن نمره را کسب کند باز می‌گرداند.

۳. integrator

این تابع به ترتیب ۳ ورودی a، b و step_size می‌گیرد و سپس انتگرال تابع pdf را از نقطه a تا b به روش دوزنقه‌ای محاسبه می‌کند. ۳. step_size دارای مقدار دیفالت 0.001 می‌باشد.

۴. between

این تابع به ترتیب ۲ ورودی a و b می‌گیرد و احتمال اینکه نمره دانشجو بین a تا b شود را توسط انتگرال‌گیری عددی به روش دوزنقه‌ای از تابع pdf در بازه a تا b محاسبه می‌کند.

توجه پیاده‌سازی شما باید به گونه‌ای باشد که به هیچ وجه نتوان هیچ شیئی از کلاس Student ایجاد کرد!

۴ NormalStudent

توجه در این کلاس فقط و فقط باید از متغیرها و توابع توضیح داده شده زیر استفاده کنید و به هیچ وجه متغیر یا تابع دیگری تعریف نکنید!

این کلاس از کلاس Student ارث‌بری می‌کند و دارای متغیرها و توابع زیر می‌باشد.

۱. mean

این متغیر از نوع double برای نگهداری میانگین توزیع نرمال استفاده می‌شود. مقدار دیفالت این متغیر را برابر ۱۲ قرار دهید.

^۳ در صورتی که روش انتگرال‌گیری عددی دوزنقه‌ای را فراموش کرده‌اید می‌توانید از این لینک استفاده کنید.

۲. std

این متغیر از نوع double برای نگهداری انحراف معیار توزیع نرمال استفاده می شود. مقدار دیفالت این متغیر را برابر ۲ قرار دهید.

۳. pdf

مانند pdf در کلاس Student این تابع یک عدد اعشاری به عنوان نمره گرفته و احتمال اینکه دانشجو آن نمره را بگیرد طبق رابطه زیر محاسبه می کند و آن را باز می گرداند.

$$pdf(x) = \frac{1}{\sqrt{2\pi}(\text{std})} e^{-\frac{1}{2(\text{std})^2} (x - \text{mean})^2} \quad (۱)$$

۴. Constructors

یک دانشجو با نام "Moradi" و میانگین ۱۴ و انحراف معیار ۲ بصورت زیر ایجاد می گردد.

```
۱ NormalStudent s{"Moradi", 14, 2};
```

کلاس شما باید از دیفالت کانستراکتور نیز پشتیبانی کند.

کد زیر احتمال اینکه یک دانشجو نمره ۱۹ بگیرد را نمایش می دهد.

```
۱ NormalStudent mha{"MHA", 18, 2};  
۲ std::cout<<mha(19)<<std::endl; // prints 0.176033
```

کد زیر احتمال اینکه دانشجوی مزبور نمره ای بین ۱۵ تا ۲۰ بگیرد را نمایش می دهد.

```
۱ std::cout<< mha(15, 20)<<std::endl; // prints 0.774509
```

۵ ExponentialStudent

توجه در این کلاس فقط و فقط باید از متغیر ها و توابع توضیح داده شده زیر استفاده کنید و به هیچ وجه متغیر یا تابع دیگری تعریف نکنید!

این کلاس از کلاس Student ارث بری می کند و دارای متغیر ها و توابع زیر می باشد.

۱. lambda

این متغیر double برای نگهداری پارامتر توزیع احتمال نمایی به کار می رود. مقدار دیفالت آن را برابر 0.07 قرار دهید.

۲. pdf

مانند pdf در کلاس Student این تابع یک عدد اعشاری به عنوان نمره گرفته و احتمال اینکه دانشجو آن نمره را بگیرد طبق رابطه زیر محاسبه می کند و آن را باز می گرداند.

$$pdf(x) = (\text{lambda})e^{-(\text{lambda})x} \quad (۲)$$

۳. Constructors

یک دانشجو با نام "Hassanzade" با پارامتر 0.05 بصورت زیر ایجاد می گردد.

```
۱ ExponentialStudent s{"Hassanzade", 0.05};
```

کلاس شما باید از دیفالت کانستراکتور نیز پشتیبانی کند.

کد زیر باید به ترتیب احتمال این که نمره دانشجو برابر 10 و همچنین بین 0 تا 10 باشد را نمایش دهد.

```
۱ ExponentialStudent vh("VH", 0.01);
۲ std::cout<<vh(10)<<std::endl; // prints 0.00904837
۳ std::cout<<vh(0, 10)<<std::endl; // prints 0.0951721
```

۶ apmidterm

در فایل های `apmidterm.h` و `apmidterm.cpp` می توانید هر تابعی که شما را در رسیدن به تست های موفق کمک کند تعریف کنید. همچنین باید تابع زیر نیز تعریف گردد.

۱. `getData`

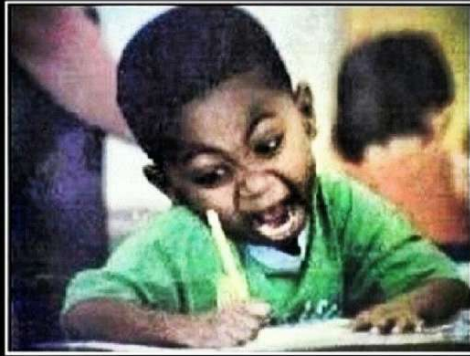
این تابع نام یک فایل `csv` را دریافت کرده و وکتوری از جنس پوینترهای `Student` یا `Student*` بازگرداند. هر سطر این فایل به یک دانشجو مربوط می شود. چنانچه دانشجو از نوع نرمال (!) باشد پس از نام، دارای ۲ پارامتر میانگین و انحراف معیار خواهد بود و چنانچه دانشجو نمایی (!) باشد پس از نام تنها دارای یک پارامتر خواهد بود.

کد زیر باید خروجی مطابق شکل ۲ داشته باشد. لازم به ذکر است که نمره A بین ۱۷ تا ۲۰، نمره B بین ۱۴ تا ۱۷، نمره C بین ۱۰ تا ۱۴ و نمره D بین ۰ تا ۱۰ می باشد.

```
۱ APQueue<Student*> apqueue1{getData("midterm.csv")};
۲ apqueue1.show();
```

```
APQueue of size: 8
No : 1 , Name: Moradi
Pr{A Grade} =0.10691 Pr{B Grade} =0.38497 Pr{C Grade} =0.44530 Pr{D Grade} =0.05482
No : 2 , Name: Karimi
Pr{A Grade} =0.02278 Pr{B Grade} =0.81869 Pr{C Grade} =0.15878 Pr{D Grade} =0.00000
No : 3 , Name: Mahikhar
Pr{A Grade} =0.00000 Pr{B Grade} =0.00000 Pr{C Grade} =0.00000 Pr{D Grade} =0.85324
No : 4 , Name: Majedinia
Pr{A Grade} =0.00000 Pr{B Grade} =0.00000 Pr{C Grade} =0.00000 Pr{D Grade} =0.84499
No : 5 , Name: Ghahramani
Pr{A Grade} =0.11791 Pr{B Grade} =0.11791 Pr{C Grade} =0.14016 Pr{D Grade} =0.19742
No : 6 , Name: Keihani
Pr{A Grade} =0.81850 Pr{B Grade} =0.02272 Pr{C Grade} =0.00000 Pr{D Grade} =0.00000
No : 7 , Name: Hassanzade
Pr{A Grade} =0.00000 Pr{B Grade} =0.00000 Pr{C Grade} =0.00000 Pr{D Grade} =0.84376
No : 8 , Name: Amini
Pr{A Grade} =0.53283 Pr{B Grade} =0.28571 Pr{C Grade} =0.02273 Pr{D Grade} =0.00003
```

شکل ۲: خروجی شماره ۲



Last five minutes
Of exam